

UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ

FACULTAD DE INGENIERÍA

POSGRADO EN INGENIERÍA MECÁNICA

ANÁLISIS Y EVALUACIÓN DEL DESEMPEÑO DE TAREAS DE ENSAMBLE EN  
AMBIENTES VIRTUALES ASISTIDOS POR SISTEMAS HÁPTICOS.

TESIS

PARA OBTENER EL GRADO DE:

DOCTOR EN INGENIERÍA MECÁNICA

PRESENTA:

M.I. GERMÁNICO GONZÁLEZ BADILLO

DIRECTOR DE TESIS:

DR. HUGO IVÁN MEDELLÍN CASTILLO

San Luis Potosí, S.L.P., Febrero de 2014



UNIVERSIDAD AUTÓNOMA  
DE SAN LUIS POTOSÍ

21 de noviembre de 2013

**M.I. GERMÁNICO GONZÁLEZ BADILLO  
P R E S E N T E.**

En atención a su solicitud de Temario, presentada por el **Dr. Hugo Iván Medellín Castillo**, Asesor de la Tesis que desarrollará Usted, con el objeto de obtener el Grado de **Doctor en Ingeniería Mecánica**. Me es grato comunicarle que en la Sesión de Consejo Técnico Consultivo celebrada el día 21 de noviembre del presente año, fue aprobado el Temario propuesto:

**TEMARIO:**

**"ANÁLISIS Y EVALUACIÓN DEL DESEMPEÑO DE TAREAS DE ENSAMBLE EN AMBIENTES VIRTUALES ASISTIDOS POR SISTEMAS HÁPTICOS"**

Introducción.

- I. Marco teórico.
- II. Desarrollo del sistema de ensamblajes virtuales.
- III. Metodología de ensamble.
- IV. Experimentación.
- V. Resultados.

Conclusiones.

Bibliografía.

Apéndice.

**"MODOS ET CUNCTARUM RERUM MENSURAS AUDEBO"**

**ATENTAMENTE**



**M. I. JORGE ALBERTO PÉREZ GONZÁLEZ  
DIRECTOR.**

UNIVERSIDAD AUTÓNOMA  
DE SAN LUIS POTOSÍ  
FACULTAD DE INGENIERÍA  
DIRECCION



AÑOS DE  
**AUTONOMÍA**  
**UASLP**  
Primera Universidad  
Autónoma en México



**FACULTAD  
DE INGENIERÍA**  
Av. Manuel Nava 8

Examen final de grado presentado por: M.I. Germánico González Badillo.

Comité evaluador:

---

Dr. Hugo Iván Medellín Castillo  
Asesor

---

Dr. Emilio Jorge González Galván  
Sinodal

---

Dr. Mauro Eduardo Maya Méndez  
Sinodal

---

Dr. Dirk Frederik De Lange  
Sinodal

---

Dr. Gilberto Mejía Rodríguez  
Sinodal

---

Dr. Ulises Zaldívar Colado  
Sinodal externo

Veredicto:

---

SAN LUIS POTOSI, S.L.P. FEBRERO DE 2014

**A mis antepasados:**

Mi abuelito Salvador González, un hombre con una mente brillante y que sin un título profesional, tenía los conocimientos y habilidades del mejor de los Ingenieros.

Mi abuelo Lorenzo Badillo, porque aun sin conocerlo, es para mí ejemplo de sabiduría y entereza.

**A mis hijos:**

Germánico, Rodrigo y Yaretzi, que me dan la energía y la motivación para enfrentar los retos diarios de la vida y en quienes se verá reflejado por siempre el esfuerzo puesto en este trabajo.

## AGRADECIMIENTOS

---

---

Para llevar a buen término este trabajo de investigación se requirió del apoyo de muchas personas importantes para mí y sería difícil listar las contribuciones de todas y cada una de ellas. Sin embargo deseo agradecer muy especialmente a:

- ❖ Mi Esposa Mary, pues este trabajo también es suyo por siempre estar a junto a mí, animarme en los momentos difíciles e impulsarme en cada momento.
  
- ❖ Mi Papa, porque siempre he contado con su amor y su apoyo incondicional y guiarme por el camino correcto, el camino del éxito verdadero.
  
- ❖ Mi Mama, por estar siempre cerca y al pendiente de mí, por ser mi inspiración para tomar el camino de la investigación y la docencia.
  
- ❖ Mi Hermano, porque siempre está disponible para ayudarme, porque es un ejemplo de integridad y valores.
  
- ❖ Al Dr. Hugo Iván Medellín Castillo, por creer en mí y darme la oportunidad de trabajar junto a él en este proyecto y por siempre contar con su apoyo en este camino.
  
- ❖ A Dios, a la vida, por haberme dado esta tarea cuando no veía claro cuál era el camino a seguir.

**Título:**

Análisis y evaluación del desempeño de tareas de ensamble en ambientes virtuales asistidos por sistemas hápticos.

**Tipo de trabajo:**

Investigación aplicada, área de ingeniería, tecnología, innovación y desarrollo tecnológico.

**Áreas académicas:**

Diseño, Manufactura, Planeación de ensambles, Realidad virtual, Sistemas hápticos.

**Responsable:**

M.I. Germánico González Badillo

Estudiante del programa de Doctorado en Ingeniería Mecánica, Facultad de Ingeniería, Universidad Autónoma de San Luis Potosí.

**Director:**

Dr. Hugo Iván Medellín Castillo

Profesor Investigador de la Facultad de Ingeniería, Universidad Autónoma de San Luis Potosí

**Palabras Clave:**

Realidad virtual, ensambles virtuales, evaluación de ensambles, modelado basado en física, modelado basado en restricciones, detección de colisiones, simulador físico, realidad virtual, sistemas hápticos, retroalimentación de fuerza.

El objetivo del presente proyecto es desarrollar una plataforma de ensamblajes virtuales que permita analizar y evaluar el efecto que pueden tener los diferentes elementos de la simulación sobre el desempeño de tareas de ensamblaje llevadas a cabo en ambientes virtuales asistidos por dispositivos hápticos. El trabajo de investigación se enfoca en cinco áreas principales:

1. Desarrollo de una plataforma háptica para ensamblajes virtuales.
2. Análisis y evaluación de diferentes simuladores físicos.
3. Análisis y evaluación de métodos de representación de objetos 3D.
4. Análisis y evaluación de diversos parámetros de diseño y simulación.
5. Creación, análisis y evaluación de restricciones de ensamblaje.

En este proyecto se presentan diversos métodos de representación de modelos 3D y el uso de restricciones de ensamblaje dinámicas con el fin de agilizar el proceso de ensamblaje virtual. Los resultados de las evaluaciones se expresan mediante parámetros cuantitativos como el tiempo de realización de la tarea de ensamblaje, tiempo de actualización de la simulación y fuerza de retroalimentación. Se considera además la percepción de usuario sobre el sistema y el proceso de ensamblaje virtual. Mediante el presente proyecto se busca aprovechar las propiedades interactivas e intuitivas que brindan los sistemas hápticos para lograr un sistema virtual de diseño que permita el desarrollo de ensamblajes virtuales de manera rápida e intuitiva tomando en cuenta el factor humano.

#### ABSTRACT

The aim of this project is to develop a virtual assembly platform which allows the assessment of the effect of various simulation components on the performance of haptic virtual assembly. This research work is focused on five key aspects:

1. The development of a haptic virtual assembly platform.
2. The analysis and evaluation of different physics simulation engines.
3. The analysis and evaluation of 3D collision shape representation methods.
4. The analysis and evaluation of various design and simulation parameters.
5. The creation, analysis and evaluation of assembly constraints.

Several techniques of 3D model representation and the use of dynamic assembly constraints are proposed in this work in order to speed up the virtual assembly process. Task completion time, simulation update rates and force feedback are measured as quantitative evaluation parameters. User perception of the system and the virtual assembly process is also considered. The project takes advantage of the interactive and intuitive properties of haptic devices to create a virtual design system which allows performing virtual assemblies in a fast and intuitive way considering human knowledge.

|   |               |
|---|---------------|
| <b>Introducción .....</b>                                 | <b>- 1 -</b>  |
| Conceptos generales.....                                  | - 1 -         |
| 1. <i>Realidad virtual</i> .....                          | - 1 -         |
| 2. <i>Sistemas hápticos</i> .....                         | - 2 -         |
| 3. <i>Comportamiento basado en física</i> .....           | - 3 -         |
| 4. <i>Sistemas hápticos y ensambles virtuales</i> .....   | - 3 -         |
| Justificación del tema.....                               | - 4 -         |
| Objetivo general.....                                     | - 6 -         |
| Objetivos específicos.....                                | - 6 -         |
| Metodología.....  | - 7 -         |
| Objetivo final.....                                       | - 8 -         |
| Aportaciones y alcances.....                              | - 8 -         |
| <br>  |               |
| <b>1. Capítulo I: Marco teórico.....</b>                  | <b>- 10 -</b> |
| 1.1 Introducción .....                                    | - 10 -        |
| 1.2 Realidad virtual en ingeniería .....                  | - 10 -        |
| 1.2.1 Simulación de procesos de diseño y manufactura..... | - 10 -        |
| 1.2.2 Aplicaciones hápticas .....                         | - 12 -        |
| 1.3 Ensamblajes.....                                      | - 17 -        |
| 1.3.1 Sistemas de ensamble virtual.....                   | - 17 -        |
| 1.3.2 Sistemas de ensamble virtual háptico.....           | - 23 -        |
| 1.3.3 Ensamble basado en física (PBM).....                | - 29 -        |
| 1.3.4 Ensamble basado en restricciones (CBM).....         | - 40 -        |
| 1.4 Fuerzas de contacto .....                             | - 42 -        |
| 1.5 Importancia de la retroalimentación de fuerza .....   | - 44 -        |
| 1.6 Percepción del peso .....                             | - 47 -        |
| 1.7 Discusión.....  | - 48 -        |



|  |               |
|--|---------------|
| <b>2. Capítulo II: Desarrollo del sistema de ensamblajes virtuales .....</b>           | <b>- 50 -</b> |
| 2.1    Introducción .....  | - 50 -        |
| 2.2    Descripción del sistema.....  | - 50 -        |
| 2.3    Integración.....  | - 53 -        |
| 2.4    Creación de modelos.....  | - 55 -        |
| 2.4.1    Representación gráfica .....  | - 57 -        |
| 2.4.2    Representación física.....  | - 57 -        |
| 2.4.3    Representación háptica.....   | - 58 -        |
| 2.5    Interacción háptica .....   | - 60 -        |
| 2.5.1    Etapas de la interacción háptica .....  | - 60 -        |
| 2.5.2    Simulación háptica.....   | - 63 -        |
| 2.5.3    Transformación de coordenadas .....   | - 64 -        |
| 2.6    Funcionalidad del sistema HAMS .....  | - 65 -        |
| 2.6.1    Cálculo de la masa del objeto virtual .....                                   | - 65 -        |
| 2.6.2    Rotaciones del objeto respecto al punto de selección .....                    | - 67 -        |
| 2.6.3    Modo háptico dual.....  | - 70 -        |
| 2.6.4    Manipulación de cámara.....   | - 70 -        |
| 2.6.5    Registro de la información durante el ensamble .....                          | - 72 -        |
| 2.6.6    Manipulación de variables de simulación .....                                 | - 75 -        |
| 2.6.7    Widgets para manipulación de propiedades .....                                | - 77 -        |
| 2.6.8    Mouse háptico.....  | - 78 -        |
| 2.6.9    Visualización inmersiva .....   | - 78 -        |
| <br>   |               |
| <b>3. Capítulo III: Metodología.....</b>   | <b>- 80 -</b> |
| 3.1    Introducción .....  | - 80 -        |
| 3.2    Modelado basado en física.....  | - 80 -        |
| 3.2.1    Escena virtual física .....   | - 81 -        |
| 3.2.2    Ciclo de simulación.....  | - 81 -        |
| 3.2.3    Representación física.....  | - 82 -        |
| 3.2.4    Representación por elementos convexos creados desde la malla triangular ..... | - 85 -        |
| 3.3    Modelado basado en restricciones.....   | - 88 -        |
| 3.3.1    Restricciones de ensamble dinámicas (DACs).....                               | - 89 -        |

|           |  |                |
|-----------|--|----------------|
| 3.3.2     | Definición de restricciones de ensamble.....                               | - 91 -         |
| 3.3.3     | Aplicación de restricciones de ensamble.....                               | - 97 -         |
| 3.3.4     | Creación de sub-ensambles.....   | - 100 -        |
| <b>4.</b> | <b>Capítulo IV: Experimentación.....</b>                                   | <b>- 102 -</b> |
| 4.1       | Introducción .....   | - 102 -        |
| 4.2       | Evaluación de simuladores físicos.....                                     | - 102 -        |
| 4.2.1     | Prueba de caída libre.....   | - 103 -        |
| 4.2.2     | Prueba de balanza.....   | - 104 -        |
| 4.2.3     | Pila de cajas.....   | - 105 -        |
| 4.2.4     | Prueba de empaquetado.....   | - 106 -        |
| 4.2.5     | Prueba de tamaño de la malla .....   | - 106 -        |
| 4.2.6     | Ensamble de extractor de poleas.....                                       | - 107 -        |
| 4.2.7     | Ensamble de la prensa de banco.....  | - 108 -        |
| 4.3       | Análisis y evaluación de modos de representación física.....               | - 109 -        |
| 4.3.1     | Evaluación de precisión geométrica .....                                   | - 109 -        |
| 4.3.2     | Evaluación del desempeño .....   | - 110 -        |
| 4.4       | Análisis y evaluación del efecto del peso.....                             | - 115 -        |
| 4.5       | Análisis y evaluación de restricciones de ensamble dinámicas.....          | - 118 -        |
| 4.6       | Evaluación de la percepción del usuario sobre el sistema HAMS .....        | - 119 -        |
| 4.7       | Comparativa del sistema HAMS contra plataformas de ensamble similares..... | - 121 -        |
| <b>5.</b> | <b>Capítulo V: Resultados .....</b>  | <b>- 124 -</b> |
| 5.1       | Introducción .....   | - 124 -        |
| 5.2       | Evaluación de los simuladores físicos.....                                 | - 124 -        |
| 5.2.1     | Prueba de caída libre.....   | - 124 -        |
| 5.2.2     | Prueba de balanza.....   | - 126 -        |
| 5.2.3     | Pila de cajas.....   | - 127 -        |
| 5.2.4     | Prueba de empaquetado.....   | - 127 -        |
| 5.2.5     | Prueba de tamaño de la malla .....   | - 128 -        |
| 5.2.6     | Ensamble de extractor de poleas.....                                       | - 129 -        |

|   |  |                |
|---|--|----------------|
| 5.2.7   | Ensamble de tornillo de banco .....  | - 130 -        |
| 5.2.8   | Resumen de la evaluación de los simuladores físicos .....                            | - 131 -        |
| 5.3   | Evaluación de los métodos de representación física .....                             | - 132 -        |
| 5.3.1   | Evaluación de precisión geométrica .....   | - 132 -        |
| 5.3.2   | Evaluación del desempeño .....   | - 137 -        |
| 5.4   | Resultados del análisis y evaluación del efecto del peso en tareas de ensamble ..... | - 146 -        |
| 5.5   | Evaluación de restricciones de ensamble dinámicas.....                               | - 149 -        |
| 5.5.1   | Ensamble del extractor de poleas.....  | - 149 -        |
| 5.5.2   | Ensamble de la bomba de engranes .....   | - 150 -        |
| 5.5.3   | Ensamble del cilindro neumático .....  | - 152 -        |
| 5.6   | Resultados de la evaluación de la percepción del usuario sobre el sistema HAMS ..... | - 153 -        |
| 5.7   | Comparativa del sistema HAMS contra plataformas de ensamble similares.....           | - 154 -        |
| <br><b>Conclusiones y trabajo a futuro.....</b> |  | <b>- 157 -</b> |
| <br><b>Bibliografía.....</b>                    |  | <b>- 162 -</b> |
| <br><b>Anexo A.....</b>                         |  | <b>- 176 -</b> |

|   |        |
|---|--------|
| Tabla 1.1. Plataformas de ensamble virtual.-----  | - 22 - |
| Tabla 2.1. Densidad de materiales utilizados.-----  | - 66 - |
| Tabla 2.2. Representación gráfica de los movimientos del usuario.-----                              | - 74 - |
| Tabla 3.1. Métodos de representación de modelos en los simuladores físicos.-----                    | - 84 - |
| Tabla 3.2. Resultados del algoritmo de identificación automática de formas cilíndricas.-----        | - 97 - |
| Tabla 4.1. Pruebas para evaluar el desempeño de simuladores físicos en el sistema HAMS.-----        | 103 -  |
| Tabla 4.2. Niveles de escala para la evaluación del tamaño de los objetos.-----                     | 107 -  |
| Tabla 4.3. Métodos de representación física disponibles en cada simulador de física.-----           | 112 -  |
| Tabla 4.4. Niveles de peso para cada elemento de la bomba de engranes.-----                         | 117 -  |
| Tabla 4.5. Diseño de experimentos para evaluación de restricciones de ensamble.-----                | 119 -  |
| Tabla 5.1. Porcentaje de incremento en el tiempo de caída libre.-----                               | 126 -  |
| Tabla 5.2. Resultados de la prueba de balanza respecto a cada simulador de física.-----             | 126 -  |
| Tabla 5.3. Tiempo de finalización de la tarea para la prueba de apilado de cajas.-----              | 127 -  |
| Tabla 5.4. Valores cualitativos para evaluación de la estabilidad y respuesta a colisiones.-----    | 128 -  |
| Tabla 5.5. Resultados de la prueba de tamaño de la malla.-----                                      | 129 -  |
| Tabla 5.6. Resultados del ensamble del extractor de poleas, evaluación de simuladores físicos.----- | 130 -  |
| Tabla 5.7. Resultados del ensamble del tornillo de banco, evaluación de simuladores físicos.-----   | 131 -  |
| Tabla 5.8. Resultados de la evaluación a los simuladores físicos.-----                              | 131 -  |
| Tabla 5.9. Número de grupos mínimos para una buena precisión geométrica.-----                       | 134 -  |
| Tabla 5.10. Resultados de la prueba de caída libre respecto al método de representación.-----       | 138 -  |
| Tabla 5.11. Resultados de la prueba de balanza respecto al método de representación, Bullet.-----   | 139 -  |
| Tabla 5.12. Resultados de la prueba de balanza respecto al método de representación, PhysX.-----    | 140 -  |
| Tabla 5.13. Resultados del ensamble del extractor de poleas.-----                                   | 143 -  |
| Tabla 5.14. Resultados del ensamble de la bomba de engranes.-----                                   | 144 -  |

---

|   |       |
|---|-------|
| Tabla 5.15. Resultados del ensamble del cilindro neumático.-----                                  | 145 - |
| Tabla 5.16. Resultados del ensamble de la prensa de banco.-----                                   | 145 - |
| Tabla 5.17. Resultados del ensamble del extractor de poleas, con restricciones de ensamble. ----- | 150 - |
| Tabla 5.18. Resultados del ensamble de la bomba de engranes, con restricciones de ensamble. ----  | 151 - |
| Tabla 5.19. Resultados del ensamble del cilindro neumático, con restricciones de ensamble. -----  | 152 - |
| Tabla 5.20. Comparativo de la plataforma HAMS contra sistemas similares. -----                    | 155 - |

|   |      |
|---|------|
| Figura 1. Ensamble real vs ensamble virtual.-----   | 5 -  |
| Figura 2. Metodología para el desarrollo del trabajo.-----  | 7 -  |
| Figura 1.1. Fresadora virtual desarrollada por Pérez y Sanz (2007) [26].-----   | 11 - |
| Figura 1.2. Cálculo de fuerza de reacción: (a) modelado por elemento finito, (b) modelado por masa-resorte.-----                            | 13 - |
| Figura 1.3. Acoplamiento virtual para manipulación de objetos y retroalimentación de fuerza.-----   | 13 - |
| Figura 1.4. Sistema desarrollado por Yang y Chen (2005) [36] para actividades de ingeniería inversa.-                                       | 14 - |
| -   |      |
| Figura 1.5. Sistema háptico para planeación de manufactura y prototipado desarrollado por Ren, et al. (2006) [43].-----                     | 15 - |
| Figura 1.6. Máquina de medición por coordenadas (HVCMM) desarrollado por Chen, et al. (2004) [45].-----                                     | 16 - |
| Figura 1.7. Simulador para maquinado dental desarrollado por Arbabtafti, et al. (2011) [46].-----   | 17 - |
| Figura 1.8. Sistemas de planeación de ensambles, (a) COSTAR y (b) HAMMS presentados por Sung y Ritchie (2009) [56].-----                    | 19 - |
| Figura 1.9. Escena virtual del sistema VTS desarrollado por Brough, et al. (2007)[57].-----   | 20 - |
| Figura 1.10. Sistema de ensamble virtual y prototipado rápido VARP, desarrollado por Lim, et al., (2009) [59].-----                         | 21 - |
| Figura 1.11. Ensamble por reconocimiento de geometrías, Sato, et al., (2011) [61].-----   | 21 - |
| Figura 1.12. Sistema de ensamble virtual HIDRA desarrollado por McDermott, <i>et al.</i> , 1999 [81].----                                   | 24 - |
| Figura 1.13. Estudio de movimientos en el proceso de ensamble, presentado por Ritchie, et al., (2004) [76].-----                            | 25 - |
| Figura 1.14. Sistema de ensamble virtual háptico propuesto por Seth, Su y Vance (2005) [84].-----   | 25 - |
| Figura 1.15. Sistema colaborativo de ensambles virtuales, Iglesias, et al., (2007) [88].-----   | 27 - |
| Figura 1.16. Sistema de RV inmersivo para ensambles virtuales, Bhatti, et al., (2009) [94].-----  | 27 - |
| Figura 1.17. Brazo robótico virtual para ensambles virtuales desarrollado por Chen, <i>et al.</i> , 2010 [67].-                             | 28 - |
| -   |      |
| Figura 1.18. Escenario de animación basada en física, Aleotti y Caselli (2011) [65].-----   | 30 - |
| Figura 1.19. Aplicación PhysicsPlayground, Kaufmann, et al., (2007) [105].-----   | 32 - |
| Figura 1.20. Clasificación de modelos: (a) objetos convexos, (b) objetos cóncavos (no convexos).- -   | 36 - |
| Figura 1.21. Esfera teselada: (a) baja resolución, (b) media resolución, (c) alta resolución.-----  | 37 - |
| Figura 1.22. Representación en dos dimensiones de la detección de colisiones para representación por voxel, Lock y Wills (2000) [113].----- | 38 - |

Figura 1.23. Aplicación de la guía por restricciones virtuales, propuesta por Tching, *et al.*, (2010) [123].--  
41 -

Figura 1.24. Representaciones de los objetos virtuales acoplados en el sistema desarrollado por Garbaya  
y Zaldivar (2007)[130].----- 44 -

Figura 1.25. Objeto dinámico manipulado por dispositivo háptico en un escenario complejo, McNeely,  
*et al.*, (1999) [132].----- 45 -

Figura 2.1. Interfaz del sistema de ensamblajes virtuales HAMS.----- 51 -

Figura 2.2. Interfaz gráfica de usuario del sistema HAMS.----- 52 -

Figura 2.3. Interfaz háptica del sistema de ensamblajes virtuales HAMS.----- 52 -

Figura 2.4. Tasas de actualización de cada módulo de simulación. ----- 53 -

Figura 2.5. Ciclo de actualización total del sistema HAMS.----- 54 -

Figura 2.6. Ventana para cargar un nuevo modelo en HAMS. ----- 55 -

Figura 2.7. Representación del objeto virtual en cada módulo de simulación del sistema de ensamblajes  
virtuales. ----- 56 -

Figura 2.8. Proceso para la creación de objetos virtuales en HAMS.----- 56 -

Figura 2.9. Proceso para la creación de la representación gráfica. ----- 57 -

Figura 2.10. Proceso para la creación de la representación física.----- 58 -

Figura 2.11. Proceso para la creación de la representación háptica. ----- 59 -

Figura 2.12. Fase de reconocimiento.----- 61 -

Figura 2.13. Fase de manipulación de objetos. ----- 62 -

Figura 2.14. Proceso de manipulación de objetos virtuales.----- 62 -

Figura 2.15. Ciclo de renderizado háptico. ----- 63 -

Figura 2.16. Proceso de transformación de coordenadas del espacio gráfico a coordenadas de los  
espacios físico y háptico. ----- 64 -

Figura 2.17. Proceso de transformación de coordenadas del espacio físico al espacio gráfico. ----- 65 -

Figura 2.18. Selección de material del objeto virtual.----- 66 -

Figura 2.19. Proceso de cálculo y asignación de la masa. ----- 66 -

Figura 2.20. Rotación del objeto virtual: (a) rotación respecto al centro geométrico, (b) rotación  
respecto al punto de selección.----- 67 -

Figura 2.21. Proceso para realizar la rotación del objeto virtual.----- 68 -

Figura 2.22. Proceso para corregir rotación de la representación háptica. ----- 69 -

Figura 2.23. Rotaciones alrededor del punto de selección.----- 69 -

Figura 2.24. Botones de vistas predefinidas y modos de manipulación de cámara. ----- 70 -

Figura 2.25. Movimiento de cámara gráfica y cámara háptica, (a) posición original, (b) movimiento de cámara gráfica, (c) movimiento de ambas cámaras. ----- - 71 -

Figura 2.26. Mapeo del mundo virtual al espacio real de trabajo.----- - 71 -

Figura 2.27. Rotación del espacio de trabajo háptico.----- - 72 -

Figura 2.28. Archivo de registro de datos.----- - 73 -

Figura 2.29. Representación gráfica de los movimientos del usuario.----- - 74 -

Figura 2.30. Panel de manipulación de propiedades hápticas.----- - 75 -

Figura 2.31. Panel de manipulación de propiedades físicas.----- - 76 -

Figura 2.32. Panel de propiedades de la escena virtual.----- - 76 -

Figura 2.33. Representación de objetos transparente y “wireframe”.----- - 77 -

Figura 2.34. Escena principal del sistema HAMS con paneles activos.----- - 77 -

Figura 2.35. Widgets para controlar variables de las restricciones de ensamble.----- - 78 -

Figura 2.36. Ejecución de HAMS en un muro de proyección tridimensional.----- - 79 -

Figura 3.1. Proceso de asignación del tiempo de paso y avance de la simulación.----- - 82 -

Figura 3.2. Representación de un engrane: (a) representación gráfica, (b) representación física.----- - 82 -

Figura 3.3. Carcasa de bomba de aceite: (a) modelo original, (b) representación convexa, (c) representación por unión de elementos convexos, (d) representación por malla triangular.----- - 83 -

Figura 3.4. Proceso de creación del modelo por descomposición en elementos convexos.----- - 85 -

Figura 3.5. Resultado de la descomposición en elementos convexos: (a) modelo original, (b) algoritmo ACD, (c) algoritmo HACD.----- - 85 -

Figura 3.6. Proceso de conversión de cada elemento de la malla triangular en un elemento convexo.- 86 -

-

Figura 3.7. Proceso de creación de la representación física utilizando el método ConvexFT----- - 86 -

Figura 3.8. Carcasa de bomba de aceite: (a) modelo original, (b) representación por descomposición en elementos convexos (ACD), (c) representación por método ConvexFT.----- - 87 -

Figura 3.9. Elementos convexos adyacentes: (a) vacío en superficies cóncavas, (b) traslape en superficies convexas.----- - 87 -

Figura 3.10. Restricciones de ensamble: (a) visualización de las restricciones, (b) aplicación de las restricciones.----- - 88 -

Figura 3.11. Restricciones de ensamble planares: (a) restricciones en cada plano, (b) aplicación de la restricción.----- - 90 -

Figura 3.12. Aplicación de las restricciones de ensamble.----- - 90 -

Figura 3.13. Menú para la definición manual de las restricciones de ensamble.----- - 91 -

Figura 3.14. Selección de puntos para definición de una restricción planar.----- - 92 -



Figura 3.15. Selección de los primeros tres puntos para crear una restricción cilíndrica. ----- - 92 -

Figura 3.16. Selección del cuarto punto para la definición de la restricción cilíndrica. ----- - 93 -

Figura 3.17. Determinación de la orientación y profundidad para la restricción cilíndrica.----- - 94 -

Figura 3.18. Índices de las líneas de borde de forma circular. ----- - 95 -

Figura 3.19. Modelos utilizados en el algoritmo de reconocimiento de formas cilíndricas: (a) base de un extractor de poleas, (b) tornillo del extractor, (c) perno del extractor, (d) brazo del extractor, (e) engrane de bomba, (f) carcasa de bomba, (g) cojinete, (h)t----- - 96 -

Figura 3.20. Eje de rotación de una forma cilíndrica. ----- - 98 -

Figura 3.21. Ensamble de partes utilizando restricción cilíndrica: a) restricciones sin aplicar, b) movimiento restringido. ----- - 98 -

Figura 3.22. Aplicación de restricción de ensamble cilíndrica.----- - 99 -

Figura 3.23. Aplicación de restricciones planares: a) restricciones sin aplicar, b) movimiento restringido. ----- - 100 -

Figura 3.24. Manipulación de sub-ensamble por medio de juntas cinemáticas: a) partes a ensamblar, b) componente ensamblado, c) manipulación de todo el objeto ensamblado. ----- - 101 -

Figura 4.1. Pruebas de caída libre.----- - 104 -

Figura 4.2. Balanza para la evaluación de la precisión de colisiones. ----- - 104 -

Figura 4.3. Prueba de apilado de cajas. ----- - 105 -

Figura 4.4. Prueba de empaquetado.----- - 106 -

Figura 4.5. Escala de ensamble perno-agujero: 1x, 0.1x, 0.01x y 0.001x. ----- - 107 -

Figura 4.6. Ensamble del extractor de poleas: (a) piezas sin ensamblar; (b) ensamble virtual; (c) ensamble real.----- - 107 -

Figura 4.7. Ensamble de la prensa de banco: (a) piezas sin ensamblar, (b) ensamble virtual, (c) ensamble real. ----- - 109 -

Figura 4.8. Modelos utilizados para realizar la evaluación de precisión geométrica. ----- - 110 -

Figura 4.9. Empaquetado de cajas para la evaluación del método de representación. ----- - 112 -

Figura 4.10. Método de representación mixta: (a) objeto dinámico, (b) objeto estático ensamblado.- 113 -

-

Figura 4.11. Caso de ensamble de bomba de aceite de engranes: (a) partes desensambladas, (b) proceso de ensamble, (c) prototipo real. ----- - 114 -

Figura 4.12. Caso de ensamble de cilindro neumático: (a) proceso de ensamble, (b) componente ensamblado, (c) prototipo rápido real.----- - 115 -

Figura 4.13. Percepción del peso por medio del dispositivo háptico.----- - 116 -

Figura 4.14. Partes reales del caso de ensamble de bomba de engranes. ----- - 117 -

Figura 4.15. Casos de ensamble utilizados para la evaluación de DACs. ----- - 118 -

Figura 4.16. Sistemas de ensamble virtual con características similares a HAMS: (a) SHARP, (b) HVAS, (c) HIVEx, (d) HIDRA.----- 123 -

Figura 5.1. Tiempo de caída libre de objetos con diferente complejidad.----- 125 -

Figura 5.2. Modelos originales utilizados en la evaluación de precisión geométrica.----- 132 -

Figura 5.3. Representación física de los objetos utilizando el algoritmo ACD en PhysX.----- 133 -

Figura 5.4. Representación física de los objetos utilizando el algoritmo HACD en Bullet con parámetros de descomposición recomendados.----- 134 -

Figura 5.5. Representación física de los objetos utilizando el algoritmo HACD en Bullet con un número de grupos alrededor del 90% del número de triángulos en la malla.----- 135 -

Figura 5.6. Representación física de los objetos utilizando el algoritmo HACD en PhysX.----- 135 -

Figura 5.7. Errores en la representación de objetos al incrementar el número de grupos en la descomposición HACD-PhysX.----- 136 -

Figura 5.8. Representación física de los objetos utilizando el algoritmo GIMPACT en Bullet.----- 136 -

Figura 5.9. Representación física de los objetos utilizando el algoritmo ConvexFT en PhysX.----- 137 -

Figura 5.10. TFT prueba de empaquetado, evaluación de métodos de representación.----- 141 -

Figura 5.11. TSF prueba de empaquetado, evaluación de métodos de representación.----- 142 -

Figura 5.12. TSH prueba de empaquetado, evaluación de métodos de representación.----- 142 -

Figura 5.13. Crono-ciclos para la evaluación del efecto del peso: (a) nivel de peso N1, (b) nivel de peso N2.----- 146 -

Figura 5.14. Tiempo de finalización de la tarea para cada nivel de peso utilizando una mano.----- 147 -

Figura 5.15. Tiempo de finalización de la tarea para cada nivel de peso utilizando dos manos.----- 148 -

Figura 5.16. Caso de ensamble del extractor de poleas: (a) ensamble utilizando únicamente PBM, (b) ensamble utilizando DACs, (c) ensamble final, (d) ensamble real.----- 149 -

Figura 5.17. Bomba de engranes: (a) ensamble utilizando únicamente PBM, (b) ensamble utilizando PBM y DACs, (c) ensamble final, (d) ensamble real.----- 150 -

Figura 5.18. Tiempos de simulación en el ensamble de la bomba de engranes: (a) utilizando únicamente PBM, (b) utilizando PBM y DACs.----- 151 -

Figura 5.19. Ensamble de cilindro neumático: (a) ensamble utilizando únicamente PBM, (b) ensamble utilizando DACs, (c) ensamble final, (d) ensamble real.----- 152 -

Figura 5.20. Resultados de las pruebas con varios usuarios en el sistema HAMS.----- 153 -

|              |   |
|--------------|---|
| $\Delta r_i$ | Diferencia de orientaciones inicial ( $^\circ$ )  |
| $\Delta r$   | Diferencia de orientaciones actual ( $^\circ$ )   |
| $\Delta t$   | Paso de la simulación   |
| $\Phi$       | Diámetro  |
| $\rho$       | Densidad ( $\text{Kg}/\text{cm}^3$ )  |
| $a$          | Aceleración ( $\text{m}/\text{s}^2$ )   |
| $B$          | Constante de amortiguamiento  |
| AABB         | <i>Axis Aligned Bounding Boxes</i>  |
| $AB_i, j, k$ | Cosenos directores del vector AB  |
| ABB          | <i>Aligned Bounding Boxes</i>   |
| ACD          | Descomposición aproximada en elementos convexos ( <i>Approximate Convex Decomposition</i> ) |
| C/D          | Relación control/display  |
| CAAP         | Planeación de ensamble asistida por computadora ( <i>Computer Aided Assembly Planning</i> ) |
| CAD          | Diseño asistido por computadora ( <i>Computer Aided Design</i> )                            |
| CAE          | Ingeniería Asistida por Computadora ( <i>Computer Aided Engineering</i> )                   |
| CAM          | Manufactura Asistida por Computadora ( <i>Computer Aided Manufacturing</i> )                |
| CBM          | Modelado Basado en Restricciones ( <i>Constraint Based Modeling</i> )                       |
| $CD_i, j, k$ | Cosenos directores del vector CD  |
| CMM          | Máquina de medición por coordenadas   |
| CNC          | Control Numérico por Computadora  |
| ConvexFT     | Elementos convexos en base a triángulo ( <i>Convex element From Triangle</i> )              |
| CSG          | <i>Constructive Solid Geometry</i>  |
| DACs         | Restricciones de ensamble dinámicas ( <i>Dynamic Assembly Constraints</i> )                 |
| DE           | Desviación estándar   |
| DFA          | Diseño para el ensamble   |
| DOF          | Grados de Libertad ( <i>Degrees of Freedom</i> )  |

|       |   |
|-------|---|
| Dx    | Dimensión en el eje X   |
| Dy    | Dimensión en el eje Y   |
| Dz    | Dimensión en el eje Z   |
| EV    | Ensamble Virtual  |
| Ex    | Coordenada X del centro del círculo proyectado sobre plano XY   |
| Ey    | Coordenada Y del centro del círculo proyectado sobre plano XY sobre el eje Y  |
| F     | Fuerza (Newton)   |
| g     | Aceleración debida a la gravedad ( $9.81 \text{ m/s}^2$ )   |
| GUI   | Interfaz Gráfica de Usuario ( <i>Graphical User Interface</i> )   |
| HACD  | Descomposición jerárquica aproximada en elementos convexos ( <i>Hierarchical Approximate Convex Decomposition</i> ) |
| HAMS  | Sistema háptico de manufactura y ensamblajes ( <i>Haptic Assembly and Manufacturing System</i> )                    |
| HLAPI | <i>High Level API</i>   |
| HMD   | Display montado en la cabeza  |
| Hz    | Hertz   |
| k     | Rigidez del resorte   |
| m     | Masa (kilogramos)   |
| M1    | Matriz de transformación del espacio háptico a espacio virtual  |
| M2    | Matriz de transformación de coordenadas del espacio físico al espacio virtual                                       |
| MEF   | Método de Elemento Finito   |
| MFC   | <i>Microsoft Foundation Classes</i>   |
| MRA   | Sistema Masa Resorte Amortiguador   |
| N     | Newton  |
| NP    | No Posible  |
| $O_a$ | Orientación actual de cámara ( $^\circ$ )   |
| $P_a$ | Posición actual de cámara (mm)  |
| PAL   | <i>Physics Abstraction Layer</i>  |
| PBM   | Modelado basado en física ( <i>Physics Based Modeling</i> )   |
| $P_i$ | Posición inicial de cámara (mm)   |
| PIP   | Panel de Interacción Personal   |

|                  |   |
|------------------|---|
| $P_n$            | Posición nueva de cámara (mm)   |
| PSE              | Simulador físico ( <i>Physics Simulation Engine</i> )                                       |
| PVD              | <i>Physics Visual Debugger</i>  |
| $r$              | Rotación por aplicar  |
| $\vec{r}_h$      | Posición de la representación háptica (mm)  |
| $\vec{r}_p$      | Posición de la representación física (mm)   |
| RV               | Realidad Virtual.   |
| SDK              | <i>Software Development Kit</i>   |
| STL              | Estereolitografía o lenguaje de teselado estándar ( <i>Standard Tessellation Language</i> ) |
| $T$              | <i>Offset</i> para traslación de la cámara (mm)   |
| $t_0$            | Tiempo inicial  |
| TFT              | Tiempo de finalización de la tarea (segundos)   |
| TSF              | Tiempo de simulación física (milisegundos)  |
| TSG              | Tiempo de simulación gráfica (milisegundos)   |
| TSH              | Tiempo de simulación háptica (milisegundos)   |
| $V$              | Volumen ( $m^3$ )   |
| VCG              | Guía por restricciones virtuales ( <i>Virtual Constraint Guidance</i> )                     |
| VPS              | <i>Voxmap Point Shell</i>   |
| VRML             | Lenguaje para modelado de realidad virtual ( <i>Virtual reality modeling language</i> )     |
| VTK              | <i>Visualization Tool Kit</i>   |
| $W$              | Peso (N)  |
| $x$              | Desplazamiento (mm)   |
| $\vec{\ddot{x}}$ | Aceleración ( $mm/s^2$ )  |
| $\vec{\dot{x}}$  | Velocidad (mm/s)  |

El desarrollo del presente trabajo de investigación ha permitido lograr las siguientes publicaciones:

- ✓ González, G., Govea, y E.H., Medellín, H.I. (2010). Performance evaluation of two different haptic devices in virtual assembly tasks. *XII Congreso Mexicano de Robótica, ComRob*, Mazatlan, Sin, Mex, Nov. 3-6.
- ✓ González-Badillo, G., Medellín-Castillo, H.I., Fletcher, C., Lim, T., Ritchie, J., y Garbaya, S. (2012). An evaluation of physics engines and their application in haptic virtual assembly environments. *Proceedings of the 37th International MATADOR Conference*, ISBN 978-1-4471-4480-9, No. 6(4), pp. 231-234.
- ✓ González-Badillo, G., Medellín-Castillo, H.I., Yu, H., Lim, T., Ritchie, J., y Garbaya, S. (2012). Effect of weight perception on human performance in a haptic-enabled virtual assembly platform. *Proceedings of the 37th International MATADOR Conference*, ISBN 978-1-4471-4480-9. No. 6(5), pp. 235-238, 2012.
- ✓ González, G., Medellín, H.I., Lim, T., Ritchie, J.M., y Sung, R. (2012). 3D object representation for physics simulation engines and its effect on virtual assembly tasks. *Proceedings of the ASME 2012 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, August 12-15, Chicago, IL, USA, 2012.
- ✓ González-Badillo, G., Medellín-Castillo, H.I., y Lim, T. (2013). Development of a Haptic Virtual Reality System for Assembly Planning and Evaluation. *3rd Iberoamerican Conference on Electronics Engineering and Computer Science, CIIIECC2013, Procedia Technology, Elsevier*, Vol.7, pp. 265-272.
- ✓ Gonzalez-Badillo, G., Medellín-Castillo, H.I., Lim, T., Ritchie, J.M., Garbaya, S. (2014). The development of a physics and constraint based haptic virtual assembly system”, Publicado en el Journal Assembly Automation, Vol. 34(1).
- ✓ Gonzalez-Badillo, G., Medellín-Castillo, H.I., Lim, T., Ritchie, J.M., Sung, R.C.W, Garbaya, S. (2014). The development of a methodology to evaluate physics simulation engines performance in haptic virtual assembly. Aceptado en el Journal Assembly Automation Vol. 34, No. 2 Marzo 2014.

Debido a la gran competencia que existe hoy en día entre las empresas de bienes y productos, se tiene la necesidad de procesos de diseño y manufactura innovadores que permitan crear nuevos productos con la mejor calidad, en el menor tiempo posible, con el menor esfuerzo y por lo tanto con el menor costo. La computación es una herramienta indispensable en tareas de diseño, planeación y manufactura, ya que su uso adecuado fomenta la creatividad, permite incrementar la productividad y por lo tanto habilita la optimización de productos y procesos.

Los ambientes virtuales generados por computadora, donde el usuario tiene un alto nivel de inmersión e interacción, están siendo estudiados en la actualidad para agilizar y optimizar los procesos de producción, diseño, planeación, análisis y entrenamiento, no solo en el sector de manufactura o diseño, sino también en diversas disciplinas como la medicina, la rehabilitación, la psicología, la arquitectura, etc.

### **Conceptos generales**

#### *1. Realidad virtual*

Existen diversas definiciones referentes a Realidad Virtual (RV), por ejemplo, Ellis (1994) [1], la define como la proyección interactiva de imágenes virtuales mejoradas y el uso de modalidades no visuales, como la retroalimentación háptica y auditiva, para hacer creer al usuario que está interactuando con objetos reales. Jayaram y Connacher (1997) [2] la definen como el uso de ambientes sintéticos generados por computadora y el hardware asociado para proveer al usuario con la ilusión de presencia física dentro de dicho ambiente. En resumen, las diversas definiciones coinciden en que la realidad virtual es un ambiente sintético generado por computadora y otros dispositivos externos que permiten la interacción con el usuario con el fin de crear la sensación de que dicho ambiente es real.

Por otro lado la inmersión se refiere a que tan profundamente el usuario esta sensorialmente involucrado al ambiente virtual, es decir, que tanto puede ver, oír, sentir e incluso oler o saborear los objetos virtuales. Existen por lo tanto diferentes niveles de inmersión para cada sistema de RV; por ejemplo, en el más alto nivel de inmersión el usuario estaría completamente aislado del mundo real y utilizaría los cinco sentidos (vista, oído, tacto, olfato y gusto) para interactuar con el ambiente virtual. A la fecha no existe ningún ambiente virtual que permita una inmersión completa de todos los sentidos. Un nivel de inmersión bajo puede involucrar únicamente uno o dos sentidos, siendo la vista el más común, donde por medio de proyecciones estereoscópicas el usuario puede ver objetos virtuales como si fueran objetos reales. También es cada vez más común el uso de dispositivos hápticos los cuales permiten sentir la forma, textura, peso e inercia de los objetos virtuales.

La Realidad Virtual (RV) ha cobrado recientemente gran importancia en el campo de la ingeniería como una poderosa herramienta de diseño debido a su capacidad de proveer un ambiente tridimensional que permite interactuar con representaciones digitales de un producto utilizando los movimientos naturales del cuerpo gracias a dispositivos periféricos adicionales como guantes, sistemas de tele-presencia, sistemas hápticos, etc.

## 2. *Sistemas hápticos*

El término háptico proviene del griego *haptesthai* que significa tocar, o lo que es relacionado al sentido del tacto y se puede interpretar bajo diversos conceptos como son:

- *Percepción háptica*, que se refiere al proceso mediante el cual se perciben los objetos a través del sentido del tacto.
- *Retroalimentación háptica*, que se refiere a recibir una fuerza o señal externa para estimular el sentido del tacto.
- *Interfaz háptica*, que se refiere a los dispositivos encargados de generar la señal de retroalimentación háptica.

Las interfaces hápticas son comúnmente utilizadas para habilitar el sentido del tacto y cinestesia al manipular objetos virtuales en sistemas de RV. También pueden ser utilizadas para la manipulación a distancia de objetos reales; por ejemplo, controlar un brazo robótico para el manejo de residuos peligrosos o brindar terapias de rehabilitación a distancia. De acuerdo con Burdea (1997) [3] los dispositivos hápticos permiten obtener de manera general tres tipos de retroalimentación:

1. Retroalimentación táctil o cutánea, la cual es útil para reconocer la textura, forma o incluso la temperatura de los objetos virtuales.
2. Retroalimentación de fuerza cinemática, la cual permite reconocer el peso, inercia y dureza de los objetos virtuales, y en su caso, sentir cuando el objeto manipulado por medio de la interfaz háptica colisiona con otros objetos virtuales.
3. Retroalimentación propioceptiva la cual permite capturar la posición y orientación del objeto virtual en un espacio tridimensional.

Estas características de retroalimentación que brindan los sistemas hápticos permiten crear aplicaciones más intuitivas donde el usuario tiene un mayor nivel de participación e influencia en ambientes virtuales. La retroalimentación háptica es crucial para incrementar el nivel de inmersión e interacción del usuario en las aplicaciones de realidad virtual. Algunas tareas como el entrenamiento de procesos repetitivos, planeación quirúrgica, diseño, escultura, prototipado, entre otros, pueden ser optimizadas al utilizar plataformas o sistemas que involucren el uso de dispositivos hápticos.



### 3. *Comportamiento basado en física*

Con el fin de lograr que los objetos creados dentro de un ambiente de RV tengan un comportamiento similar a los objetos reales se utiliza el modelado basado en física, el cual permite que los objetos virtuales actúen bajo el efecto de la gravedad, tengan respuesta a colisiones y estén sujetos a fuerzas de fricción, restitución, entre otras propiedades físicas. Para lograr esto se utiliza software conocido como simuladores físicos, los cuales utilizan la mecánica Newtoniana o dinámica del cuerpo rígido para resolver el movimiento de cada uno de los objetos virtuales. Las funciones contenidas en estos simuladores requieren principalmente la definición de la masa y la forma de los objetos virtuales para calcular su velocidad, aceleración, y posición en cada paso o instante de la simulación.

El principal objetivo de los simuladores físicos es lograr el comportamiento dinámico de los objetos virtuales, además de calcular la respuesta a colisiones cuando dos o más objetos se encuentran en contacto para evitar que se traslapen, tal como sucede en el mundo real. Este tipo de simuladores fueron originalmente creados para ser utilizados en la industria de los videojuegos con el fin de aumentar el nivel de realismo e interacción con el usuario; sin embargo, han tenido gran aceptación en muchas otras disciplinas como simulaciones dinámicas, planeación de procesos, entrenamiento, e incluso planeación quirúrgica y simulaciones científicas. Existen diversas marcas y tipos de simuladores físicos, dependiendo del desarrollador que las crea. Entre los más comunes se encuentran: Havok [4], Bullet [5], PhysX [6], Newton [7] y Open Dynamics Engine [8]. Cada uno de estos simuladores posee características diferentes, por lo tanto su selección e implementación dependerá de la aplicación a desarrollar.

### 4. *Sistemas hápticos y ensambles virtuales*

Debido a los avances recientes en la computación y por consiguiente en los gráficos por computadora, así como en software y hardware periférico, las tecnologías de realidad virtual han alcanzado cierto grado de madurez y son cada vez más comunes. Actualmente diversas compañías e instituciones académicas se encuentran investigando el desarrollo y aplicación de técnicas de realidad virtual en procesos de análisis, diseño, entrenamiento y manufactura, entre otros.

El ensamble de componentes mecánicos es altamente compatible con las simulaciones en ambientes virtuales, ya que el usuario puede interactuar con los objetos virtuales de manera natural y llevar a cabo el ensamble de piezas tal y como en el mundo real. Así, el ensamble virtual (EV) se define como el proceso de ensamblar modelos generados mediante paquetes de diseño asistido por computadora, CAD, utilizando una interfaz inmersiva y los movimientos humanos naturales (Chang y Vance, 2004 [9]).

Con el fin de utilizar los movimientos naturales del cuerpo en una aplicación de realidad virtual y lograr así un mejor control sobre el movimiento de los objetos virtuales, es necesario que el usuario pueda reconocer la forma, rigidez, colisiones, peso e inercia de los objetos virtuales

manipulados. Para lograr esto se integran interfaces hápticas a los sistemas de realidad virtual, añadiendo así el sentido del tacto y cinestesia en respuesta a los movimientos realizados por el usuario.

Las interfaces hápticas crean un sistema cerrado que permite ligar datos sintéticos generados por la computadora con movimientos humanos y viceversa, es decir, el usuario especifica posiciones a la computadora por medio del dispositivo háptico y la computadora responde con señales que se traducen en una fuerza aplicada al usuario a través de la interfaz háptica. Los sistemas hápticos permiten una manipulación natural de los objetos virtuales y por lo tanto son ideales para interactuar con estos objetos dentro de sistemas de ensamble virtual, ayudando así a comprender los procedimientos asociados a las tareas de ensamble (Hayward, *et al.*, 2004 [10]).

El desarrollo de prototipos, incluyendo su proceso de ensamble, es una parte integral del ciclo de diseño y creación de un nuevo producto. Mediante el uso de plataformas de realidad virtual apoyadas por sistemas hápticos es posible examinar un prototipo incluso antes de construirlo, observando su forma, textura, secuencia de ensamble y otras propiedades que pueden ayudar a crear mejores productos en un tiempo menor (Lennart y McLaughlin, 2002 [11]).

### **Justificación del tema**

El ensamble de componentes es uno de los procesos de manufactura más importantes en la industria debido a que puede representar hasta un 60% del costo total de un producto (Boothroyd, 1992 [12]). Durante la fase de diseño de un nuevo producto se pueden utilizar diversos sistemas como el diseño asistido por computadora, CAD, o la planeación de ensamble asistida por computadora, CAAP, para planear las secuencias, métodos y trayectorias de ensamble. Sin embargo, estos sistemas utilizan algoritmos como el reconocimiento de formas y características de los objetos para realizar de forma automatizada el ensamble. Estos procesos automatizados no siempre llevan a una solución óptima o correcta del ensamble, ya que pueden generar secuencias o trayectorias de ensamble que no son posibles de implementar en la realidad; por ejemplo, en los sistemas CAD es posible traslapar o atravesar objetos para realizar un ensamble, además de que el movimiento de los objetos no se realiza de manera natural. En estos sistemas la experiencia, conocimiento, intuición y destreza humana no son tomados en cuenta, además no consideran factores como la calidad y funcionalidad del ensamble, distribución del espacio de trabajo, o ergonomía (Xia, *et al.*, 1992 [13]). La falta de las restricciones físicas que el usuario experimenta en el ambiente real, como colisiones e interferencias con otros objetos, son otras de las carencias de los sistemas CAD / CAAP.

La planeación y evaluación de los procesos de ensamble puede ser mejorada al proveer un sistema de ensamble donde los usuarios puedan tener un mayor nivel de inmersión e interactividad, que les permita utilizar pistas más intuitivas como las colisiones entre objetos, colisiones con obstáculos, fricción, inercia, peso, restitución, sonido, estereovisión, etc.

(Gutierrez, *et al.* 1998 [14]). Además, la evaluación de los procesos de ensamble/desensamble en las primeras etapas del proceso de diseño puede ayudar a reducir dramáticamente el tiempo, costo y materiales asociados a la construcción de prototipos físicos (Liu, *et al.*, 2012 [15]).

Varios autores han desarrollado distintas plataformas de ensamble virtual utilizando dispositivos hápticos, por ejemplo el sistema HAMMS (Lim, *et al.*, 2007) [16]), el sistema SHARP (Seth, *et al.*, 2007 [17]), el sistema HVAS (Xia, *et al.*, 2011 [18]), el sistema HIVEx (Bhatti, *et al.*, 2009 [19]) y el sistema HIDRA (Coutee, *et al.*, 2001 [20]), por mencionar solo algunos. Dentro de las diferentes plataformas de ensamble virtual reportadas en la literatura se han identificado tres aplicaciones principales para estos sistemas: (i) planeación y evaluación de ensambles de nuevos componentes, (ii) análisis para operaciones de mantenimiento y (iii) entrenamiento de operaciones repetitivas.

Los investigadores que diseñan, desarrollan y evalúan las plataformas de ensamble virtual han reportado diversas dificultades o limitaciones en su implementación; entre las más frecuentes se encuentran:

- ✓ Dificultad de sincronización entre los módulos gráfico, físico y háptico.
- ✓ Baja eficiencia de la detección de colisiones.
- ✓ Uso de modelos de baja complejidad.
- ✓ Movimientos poco intuitivos al realizar el ensamble.
- ✓ Alto costo computacional.
- ✓ Baja estabilidad de los objetos virtuales.

Estas limitaciones conducen a que el proceso de ensamble tenga cierto grado de dificultad y por lo tanto el tiempo para realizar el ensamble virtual será mayor que el tiempo requerido para realizar el mismo ensamble en el mundo real, Figura 1. Una forma de medir el desempeño de un sistema de ensambles virtuales es por medio del tiempo requerido para completar o terminar el ensamble.

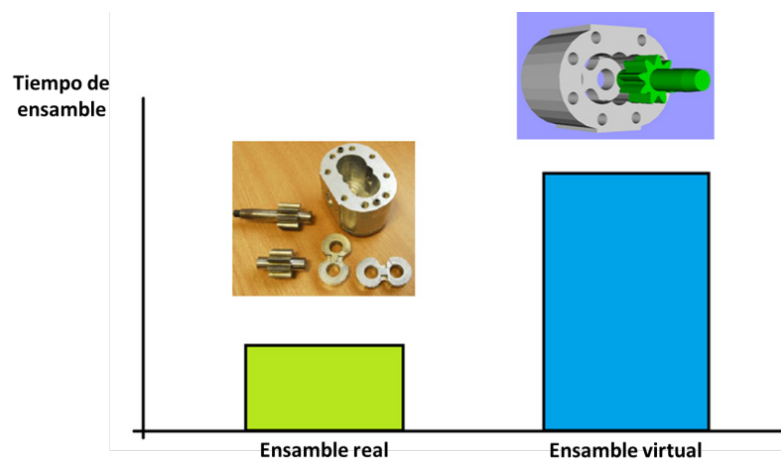


Figura 1. Ensamble real vs ensamble virtual.

Aunque a nivel de prototipo se han desarrollado diversas plataformas de ensamble virtual, los efectos que los diversos parámetros y componentes de la simulación tienen en el desempeño de las tareas de ensamble virtual aún no han sido clarificados; por ejemplo, el efecto de la fuerza de retroalimentación, las características de los simuladores físicos, el método para la creación de los modelos, los algoritmos para la detección de colisiones, etc. Por lo tanto resulta importante y necesario identificar y cuantificar el efecto que tienen estos y otros componentes de la simulación en el desempeño de tareas de ensamble virtual, con el fin de proponer las mejoras necesarias para crear sistemas de ensambles virtuales que sean más eficientes, intuitivos y que puedan llegar a tener una aplicación real.

En el presente trabajo de investigación se presenta el desarrollo y uso de una plataforma de ensambles hápticos que permite ajustar y evaluar los parámetros y componentes de la simulación durante tiempo de ejecución del programa, con el fin de determinar su influencia en el desempeño de las tareas de ensamble virtual.

### **Objetivo general**

Desarrollar un sistema de realidad virtual asistido por dispositivos hápticos en el cual se pueda analizar y evaluar la influencia de diversos factores de simulación, diseño y configuración en el desempeño de tareas de ensamble virtuales. De esta manera se pretende incrementar el conocimiento en sistemas de realidad virtual que permita el desarrollo de aplicaciones computacionales para agilizar el proceso de diseño y manufactura de componentes mecánicos, haciendo énfasis en el proceso de ensamble.

### **Objetivos específicos**

Para lograr analizar y evaluar el efecto que diversos componentes y parámetros de simulación tienen sobre el desempeño de tareas de ensamble asistidas por dispositivos hápticos, se plantean los siguientes objetivos específicos:

- ✓ Desarrollo de una plataforma háptica para ensambles virtuales.
- ✓ Análisis y evaluación de diferentes simuladores físicos.
- ✓ Análisis y evaluación de la influencia del modo de representación de objetos 3D.
- ✓ Análisis y evaluación del uso de restricciones de ensamble.
- ✓ Análisis y evaluación de diversos parámetros de diseño y simulación.
- ✓ Evaluación general del sistema desarrollado en tareas de ensamble.

## Metodología

El presente trabajo de investigación comprende tres etapas, Figura 2, las cuales se detallan a continuación:

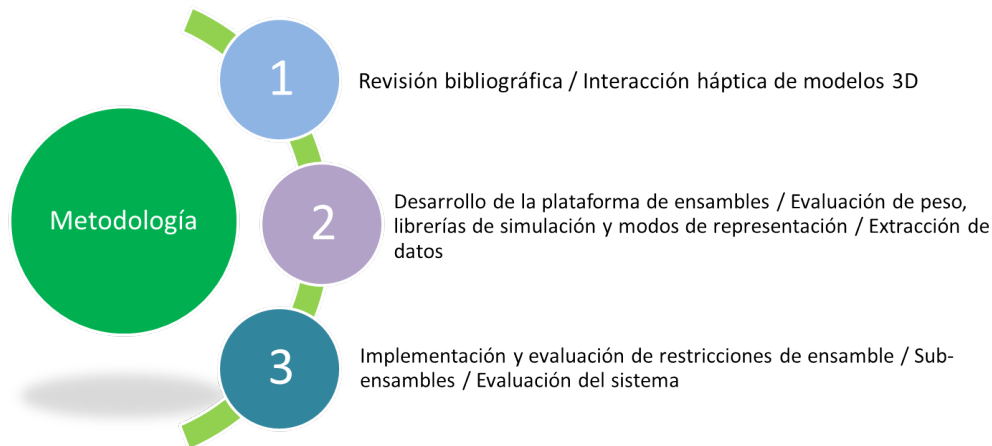


Figura 2. Metodología para el desarrollo del trabajo.

En la primera fase se contempla realizar una revisión bibliográfica extensa para identificar las investigaciones actuales y así poder enfocar el trabajo en ciertas áreas de oportunidad que puedan representar una contribución real a la materia. En esta misma fase se deberán crear las bases para el desarrollo de una plataforma virtual háptica que permita la manipulación de modelos CAD 3D en ambientes virtuales.

En la segunda fase se considera el desarrollo de la plataforma virtual que permita llevar a cabo ensambles virtuales de manera natural e intuitiva, los datos generados a partir de la interacción del usuario para realizar el ensamble podrán ser grabados para su posterior análisis. Asimismo se llevará a cabo un estudio que permita identificar el efecto de diversos elementos y parámetros de simulación sobre el desempeño de la plataforma y los ensambles virtuales. El desempeño será evaluado en términos de precisión geométrica, representación de modelos, precisión y repuesta de colisiones, estabilidad de los objetos virtuales, estabilidad de la simulación y tiempo en completar la tarea de ensamble.

En la tercera fase el trabajo se enfocará en la creación y evaluación de restricciones de ensamble, las cuales permitan la reducción de los grados de libertad del objeto manipulado cuando este se encuentra cerca de la posición de ensamble. Las restricciones de ensamble asistirán al usuario para alcanzar la posición objetivo, y por lo tanto se prevé que el desempeño de la simulación mejore. También se implementará la creación y uso de sub-ensambles al restringir los grados de libertad de la pieza ensamblada, con el fin de que solo tenga movimiento relativo respecto a la pieza base. Finalmente se realizará una comparativa teórica con sistemas similares, resaltando las diferencias, capacidades adicionales y contribuciones del sistema propuesto.

### **Objetivo final**

La finalidad que tiene el desarrollar un sistema de ensambles virtuales es que pueda ser utilizado para realizar las siguientes funciones:

- ✓ Analizar el ensamble/desensamble de componentes en la etapa de diseño.
- ✓ Entrenar a usuarios en operaciones nuevas y repetitivas.
- ✓ Generar de forma automática planes de ensamble (manuales y robóticos).

La plataforma desarrollada en el presente proyecto de investigación se enfoca únicamente en las primeras dos funciones.

### **Aportaciones y alcances**

El presente proyecto se enfoca en el desarrollo de una plataforma de ensambles virtuales con la cual se pretende analizar el efecto que diversos parámetros y componentes de la simulación tienen en el desempeño de ensambles virtuales. Al analizar diversos casos de ensamble en la plataforma virtual se podrá generar información valiosa acerca del desarrollo eficiente de sistemas de ensamble virtuales. Lo anterior llevará a generar el conocimiento para crear aplicaciones que puedan ser utilizadas en procesos industriales reales, teniendo como objetivo la reducción en el tiempo de desarrollo y manufactura de nuevos componentes, aprovechando las características intuitivas que brindan los sistemas hápticos. La aplicación desarrollada no pretende ser una interfaz directa entre el usuario y una máquina de ensamble y/o maquinado real (p.ej. brazo robótico), sin embargo será capaz de brindar al usuario información útil acerca de procedimientos y trayectorias de ensamble.

Mediante el uso de esta aplicación se podrán evaluar factores que pueden afectar el proceso de ensamble de componentes mecánicos, tales como:

- ✓ Espacio de trabajo.
- ✓ Movimientos.
- ✓ Facilidad de ensamble.
- ✓ Trayectorias de ensamble.
- ✓ Tolerancias y ajustes de las piezas ensambladas.

La plataforma de ensambles virtuales desarrollada permitirá trabajar con modelos 3D creados mediante cualquier paquete de diseño asistido por computadora (CAD). Los ensambles analizados en este trabajo de investigación representan casos reales de ensamble de componentes.

El presente documento de tesis está organizado de la siguiente manera:

En el Capítulo I se muestra un estudio bibliográfico donde se tratan los temas involucrados en el presente trabajo de investigación. Se muestra el estado del arte de la aplicación de técnicas de

realidad virtual en la simulación de procesos de diseño y manufactura. Asimismo se estudian los sistemas de ensamble virtual háptico desarrollados por diversos autores, con enfoque en su desarrollo, aplicación y características principales.

En el Capítulo II se expone el desarrollo, características y funciones del sistema de ensambles virtuales HAMS (*Haptic Assembly and Manufacturing System*), describiendo cada uno de los módulos que lo integran. Se muestra la secuencia de cada paso de la simulación para formar el ciclo actualización total y se describe el método utilizado para realizar la interacción háptica entre el usuario y los objetos virtuales.

En el Capítulo III se muestra la metodología utilizada para la simulación del ensamble. Se describen los conceptos e integración del modelado basado en física, así como los diferentes métodos para crear la representación de los modelos dentro del simulador físico. Se presentan las restricciones de ensamble dinámicas (DACs), las cuales son un método híbrido que combina el realismo del modelado basado en física con la eficiencia del modelado basado en restricciones.

En el capítulo IV se describe cada una de las pruebas realizadas en el sistema HAMS para determinar la influencia de diversos parámetros de simulación en el proceso de ensamble virtual, incluyendo la metodología seguida para evaluar los simuladores físicos, los métodos de representación, el efecto del peso de los objetos virtuales, las restricciones de ensamble dinámicas y el desempeño general del sistema HAMS. Además se plantea la comparativa teórica de la plataforma HAMS contra sistemas similares. Los resultados de cada una de estas evaluaciones se muestran y analizan en el capítulo V.

Finalmente, en la sección de conclusiones y trabajo a futuro se exponen los resultados más relevantes y las dificultades encontradas que pueden generar una contribución en el desarrollo de sistemas de realidad virtual aplicados a procesos de ingeniería; así como las funciones y evaluaciones que deberán realizarse en el sistema para llevarlo a una aplicación real.

### 1.1 Introducción

Durante la etapa de diseño de un nuevo producto se requiere del análisis de todos los procesos necesarios para su creación, desde la selección de materiales y procesos de manufactura para elaborar las piezas individuales, hasta el proceso de ensamble de cada una de las partes para formar un producto final. Existen diversas soluciones que permiten llevar a cabo el análisis y planeación del ensamble, como el uso de sistemas de diseño asistido por computadora, CAD, sistemas de planeación de ensamble asistidos por computadora, CAAP, o el uso de prototipos físicos. Sin embargo, en años recientes diversos autores han investigado diferentes técnicas y sistemas de realidad virtual para poder simular el proceso de ensamble con mayor eficiencia y naturalidad. Los sistemas de realidad virtual pueden ser utilizados para simular, analizar y evaluar diversos procesos de ingeniería, así como entrenar al personal involucrado. A continuación se describen algunos de los sistemas desarrollados por diversos autores, sus características principales, sus aplicaciones y los conceptos básicos de estos sistemas de realidad virtual.

### 1.2 Realidad virtual en ingeniería

#### 1.2.1 Simulación de procesos de diseño y manufactura

Existen diversas herramientas informáticas para simular los procesos de diseño y manufactura, por ejemplo, la ingeniería asistida por computadora (CAE) y la manufactura asistida por computadora (CAM). Sin embargo, aún existe la necesidad de evaluar de una manera más intuitiva diversos aspectos como la facilidad de manufacturar un componente durante la fase de diseño con el fin de asegurar que el producto pueda ser producido exitosamente en términos de calidad, costo y tiempo. Con este propósito Henderson y Kattethota (1999) [21] realizan dos tipos de análisis, uno para procesos por sustracción de material, específicamente el fresado estándar, y el otro para un proceso de adición, la manufactura por capas. En el primer caso la pieza es subdividida en unidades más pequeñas llamadas primitivos, donde cada primitivo es analizado para accesibilidad de la herramienta, con esta información se calcula una trayectoria de corte para cada primitivo y junto con los parámetros de corte sirve para estimar el tiempo y el costo del proceso. En el segundo caso se evalúa específicamente el efecto de la orientación de la parte y el espesor de la capa sobre la rugosidad superficial de la pieza.

Con el propósito de lograr una contribución significativa y llamativa para la industria, surge la necesidad de integrar diversas funciones como son el modelado virtual, manufactura virtual, la planeación de trayectorias y fabricación de prototipos rápidos en una sola aplicación que pueda ofrecer todas estas características. Xin (2003) [22] propone una plataforma de este tipo, la cual



soporta el diseño conceptual de dispositivos mecánicos, el diseño de la estructura, la simulación de comportamiento mecánico y la realización de ensambles virtuales de manera automática.

Song, *et al.* (2004) [23] propusieron un centro de prototipado interactivo el cual incluye la síntesis del modelo en plastilina virtual, escaneo tridimensional, maquinado robótico y herramientas geométricas avanzadas para el maquinado robótico. Un método basado en escaneado 3D es utilizado para calibrar el sistema coordenado de la pieza de trabajo y la sujeción, como resultado el prototipado físico y virtual pueden ser realizados simultáneamente. En el maquinado robótico las fuerzas de corte son limitadas debido a la poca rigidez que puede presentar el brazo robótico. Además se restringe la complejidad de las piezas debido a que no siempre se pueden alcanzar todas las superficies a maquinar (Chan y Tan (2005) [24]).

Diversos autores han desarrollado aplicaciones de realidad virtual para facilitar la programación intuitiva de centros de maquinado; por ejemplo, Aras y Yip-Hoi (2007) [25] presentan un sistema de maquinado virtual que hace uso de la geometría de la herramienta y la geometría de la pieza de trabajo para predecir las fuerzas de corte. Pérez y Sanz (2007) [26] proponen una fresadora de 3 ejes virtual, con propósitos educativos y de entrenamiento; el sistema es capaz de adaptar la máquina para añadir nuevos dispositivos o trabajar como una celda de manufactura, Figura 1.1.

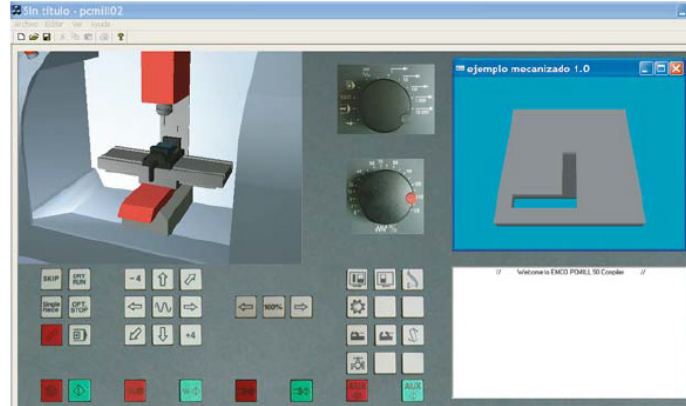


Figura 2.1. Fresadora virtual desarrollada por Pérez y Sanz (2007) [26].

Medellín, *et al.* (2007) [27] presentan un análisis de manufacturabilidad para prototipos rápidos creados por deposición de material. Evalúa tres aspectos: identificación de paredes delgadas, identificación de vértices, e identificación de volúmenes no deseados, o muy pequeños. Se propone la creación de operadores más completos para la etapa de análisis de manufacturabilidad, los cuales puedan incorporar la evaluación del acabado superficial en el volumen construido.

El cálculo de fuerza de corte, potencia, vibración y torque en el proceso de fresado son predichos y optimizados en un ambiente virtual desarrollado por Merdol y Altintas (2008) [28], los datos obtenidos son verificados mediante un caso de estudio. Chen y Cai (2008) [29] presentan un método para la predicción de errores de maquinado sobre la trayectoria de la herramienta en un proceso de acabado utilizando una máquina virtual de tres ejes. Cybercut (Sundararajan y Wright, 2008) [30] es un sistema integrado de planeación del proceso que convierte el diseño del usuario en instrucciones para una fresadora CNC, incluye extracción de características geométricas, planeación de métodos de sujeción, selección de herramientas y planeación de la ruta de la herramienta.

Dos herramientas para asegurar la eficiencia y efectividad del proceso de diseño de un producto son propuestas por Molcho, *et al.* (2008) [31]. La primera considera un análisis CAM que facilita la validación de manufactura en la etapa de diseño al identificar características que son difíciles, caras o imposibles de maquinar. La segunda incluye un módulo CAD que permite al usuario adaptar el diseño a las máquinas herramientas disponibles y estimar el efectos de las decisiones de diseño en la precisión del producto final.

Las aplicaciones de las tecnologías de realidad virtual en los procesos de diseño, análisis de funcionalidad, ergonomía, mantenimiento, ensamble mecánico y entrenamiento han llevado a obtener resultados positivos en su aplicación en la industria automotriz. De esta manera se confirma que el uso de estas herramientas puede conducir a la mejora de los procesos de desarrollo de nuevos productos, así como al desarrollo de nuevas metodologías y/o procedimientos de entrenamiento (Gironimo y Lanzotti, 2009 [32]).

### 1.2.2 Aplicaciones hápticas

#### *Renderizado háptico*

Los sistemas hápticos son una tecnología emergente que permite al usuario interactuar mediante el sentido del tacto con los objetos virtuales. Mediante el uso de interfaces hápticas se obtiene un mejor entendimiento de las representaciones virtuales de procesos y objetos, ya que se puede interactuar con ellos de una manera más natural e intuitiva. El sentido del tacto es fundamental en la manera en que se perciben los objetos (Thurfiell, *et al.*, 2002) [11].

La fuerza de reacción que siente el usuario al tocar objetos en un ambiente virtual háptico puede ser calculada utilizando diferentes métodos, los más conocidos son el método de elemento finito (MEF), Figura 1.2a, y el sistema masa-resorte-amortiguador (MRA), Figura 1.2b. El primero divide la pieza en subregiones representadas por elementos finitamente pequeños y que tienen una geometría definida. Este método se basa en aproximar una función continua de un modelo que está compuesto por diversos elementos, cada uno de los cuales está a su vez definido por otra función continua. En el segundo método el objeto se divide en mallas (por ejemplo mallas triangulares), a cada vértice del elemento se le asigna una partícula de masa, y sobre cada lado se asigna un resorte y un amortiguador que conectan a cada

partícula de masa. El modelo masa-resorte-amortiguador es muy popular debido a su simplicidad y facilidad de modelado computacional (Payandeh, 2001 [33]).

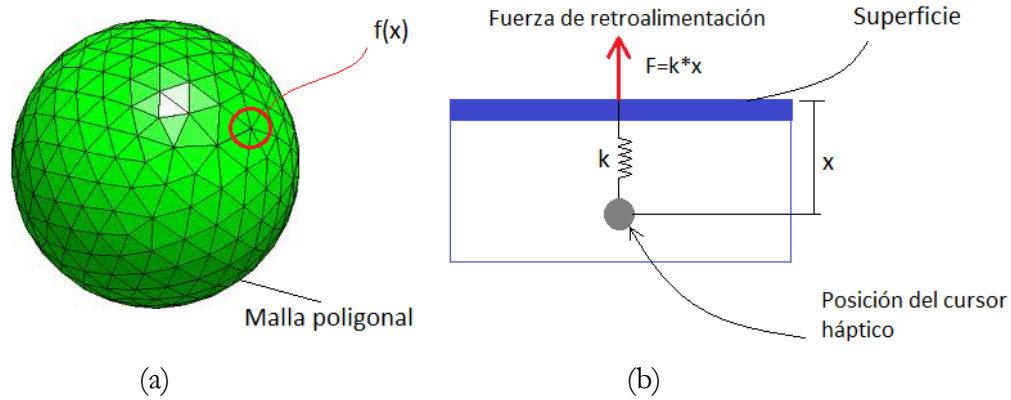


Figura 2.2. Cálculo de fuerza de reacción: (a) modelado por elemento finito, (b) modelado por masa-resorte.

Para manipular un objeto virtual por medio del dispositivo háptico se puede utilizar también el modelo MRA. De esta manera se puede aplicar una fuerza al objeto virtual y moverlo dentro de la escena. Cuando el objeto manipulado colisiona con otro objeto, el simulador físico calcula las fuerzas de reacción a través del mismo modelo MRA y las aplica al objeto manipulado, de esta manera el usuario puede sentir las fuerzas de la colisión entre objetos, Figura 1.3 (Vance y Dumont, 2011, [34]).

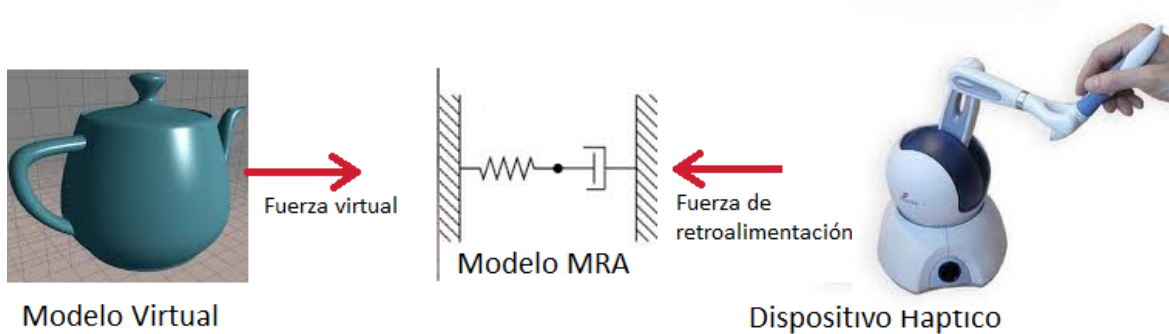


Figura 2.3. Acoplamiento virtual para manipulación de objetos y retroalimentación de fuerza.

*Diseño*

Potter, *et al.* (2004) [35] demuestran que un dispositivo háptico puede ser utilizado en tareas de ingeniería inversa. Se observa que se pueden preservar las características del modelo ya que se logra obtener un filtrado apoyado por sistemas hápticos como una herramienta para modificar con buena calidad una pieza. Por otro lado, Yang y Chen (2005) [36] también proponen una metodología para realizar ingeniería inversa a través de sistemas hápticos. El entorno de la

pieza es tratado como un bloque de plastilina sobre el cual se esculpe la pieza, al momento de que el usuario talla la plastilina virtual para obtener la forma del modelo, la interfaz háptica le permite sentir la fuerza de tallar la plastilina y la forma del modelo, Figura 1.4. Yang, *et al*, (2005) [37] presentan una plataforma que permite agilizar el proceso de creación de un nuevo componente utilizando el apoyo de un dispositivo háptico, el cual es empleado como un mecanismo central que permite hacer ingeniería inversa, modelado de forma, análisis en tiempo real de propiedades mecánicas, planeación de trayectorias de maquinado y de inspección de tolerancias de manera similar a una máquina de medición por coordenadas. Debido al uso del dispositivo háptico el sistema se vuelve más intuitivo, eficiente y amigable al usuario. Un sistema similar es presentado por He (2008) [38] donde se proponen métodos de modelado háptico para manufactura virtual, cuyas aplicaciones incluyen ingeniería inversa, torneado y taladrado virtual apoyado por sistemas hápticos, ensamble robótico virtual basado en tele-operación virtual háptica, planeación háptica de trayectorias y ensambles virtuales hápticos.

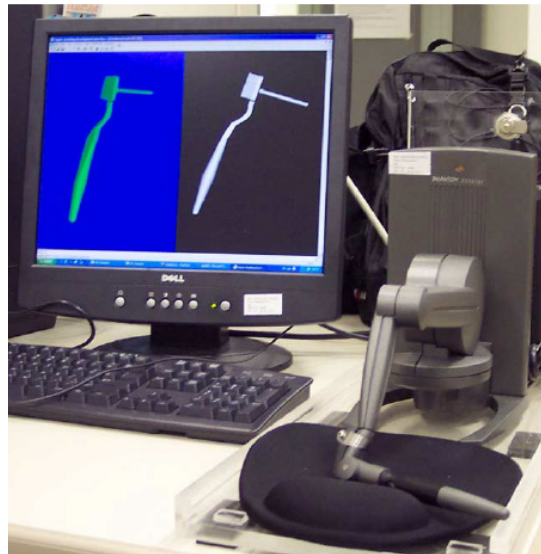


Figura 2.4. Sistema desarrollado por Yang y Chen (2005) [36] para actividades de ingeniería inversa.

Una interfaz háptica para el modelado y evaluación del diseño de piezas es propuesta por Yang, *et al*. (2009) [39]. En este sistema se puede aplicar una carga al objeto utilizando el dispositivo háptico, al mismo tiempo la reacción del objeto en términos de desplazamiento y reacciones puede ser visualizada y sentida por el usuario, obteniendo así una evaluación instantánea de la rigidez de la parte diseñada. El sistema propuesto puede ayudar a mejorar la eficiencia del diseño al proveer una evaluación funcional intuitiva en las etapas tempranas del proceso de diseño.

*Manufactura*

Por otro lado, el uso de sistemas hápticos en la planeación de procesos de manufactura también ha sido estudiado. Una interfaz intuitiva que permite la generación de trayectorias para una fresadora de 5 ejes es propuesta por Balasubramaniam, *et al.* (2002) [40]. El sistema está basado en un dispositivo háptico de 5 grados de libertad y un modelo tridimensional sólido impenetrable. En el proceso de sentir o “pintar” el objeto el usuario enseña al centro de maquinado a tallar el objeto virtual. Un sistema similar es presentado por Zhu y Lee (2004) [41], donde se propone una metodología analítica para un sistema de esculpido háptico para prototipos virtuales. El movimiento de la herramienta y la intersección del volumen con el barrido analítico de la herramienta son utilizados para actualizar el material virtual remanente. La trayectoria de la herramienta generada en el sistema puede ser procesada por algoritmos de planeación de maquinado para generar el código de control numérico. El uso de un dispositivo háptico de 5 grados de libertad para simular la manipulación de una máquina fresadora de escritorio de 3 ejes es presentado por Ramesh, (2004) [42]. En este trabajo se crean e implementan algoritmos para detección de colisiones entre los objetos virtuales y la herramienta.

Ren, *et al.* (2006) [43] proponen el uso de un modelo volumétrico llamado tri-dexel y retroalimentación de fuerza háptica para prototipado virtual y planeación de manufactura mediante el cual se generan trayectorias de herramienta para el maquinado virtual de superficies esculpidas, Figura 1.5. El resultado se verifica en una máquina CNC virtual de 5 ejes. Por otro lado Cheng y Tang (2008) [44] estudian un sistema háptico que permite la modificación interactiva de la orientación del cortador durante la operación de acabado en un proceso de 5 ejes con el fin de mejorar la calidad del acabado superficial de la pieza. Además el sistema incluye el uso de una guía háptica activa para asistir en la generación de trayectorias de la herramienta libres de colisiones.



Figura 2.5. Sistema háptico para planeación de manufactura y prototipado desarrollado por Ren, *et al.* (2006) [43].

*Otras aplicaciones*

En el trabajo realizado por Chen, *et al.* (2004) [45] se presenta una máquina de medición por coordenadas háptica (HVCMM), Figura 1.6. El sistema utiliza las técnicas de modelado háptico para la programación de una máquina de medición por coordenadas (CMM) fuera de línea. El dispositivo háptico es utilizado para controlar la sonda de medición y apuntarla en la superficie de un modelo CAD para generar la trayectoria de inspección. Con el sistema HVCMM es mucho más fácil generar una trayectoria de medición libre de colisiones, además puede ser utilizada para el entrenamiento en la operación de una máquina de medición por coordenadas real.

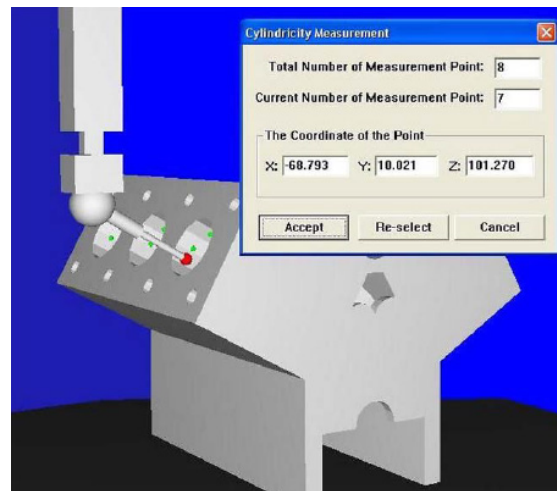


Figura 2.6. Máquina de medición por coordenadas (HVCMM) desarrollado por Chen, *et al.* (2004) [45].

Arbabtafti, *et al.* (2011) [46] presentan un simulador de maquinado de hueso con fines de entrenamiento, Figura 1.7. El sistema aplica los principios de la mecánica del hueso para calcular la energía requerida para remover una unidad de volumen de hueso y el resultado es utilizado para calcular la fuerza de retroalimentación háptica. El sistema utiliza un dispositivo háptico Phantom y la representación de modelos voxel, en la cual los objetos virtuales están construidos por pequeños cubos tridimensionales. Similarmente en el trabajo presentado por Zerbato D. *et al.* (2011) [47] se propone un simulador quirúrgico basado en física que incluye retroalimentación háptica. El simulador utiliza la unidad procesadora de gráficos para obtener un comportamiento realista de los órganos durante el proceso de corte, logrando mantener las altas frecuencias de actualización requeridas por la simulación háptica. El simulador permite la actualización de cambios de información topológica en tiempo real y la interacción con modelos deformables. Se concluye que la simulación de modelos deformables basados en física incrementa el realismo del ambiente virtual.

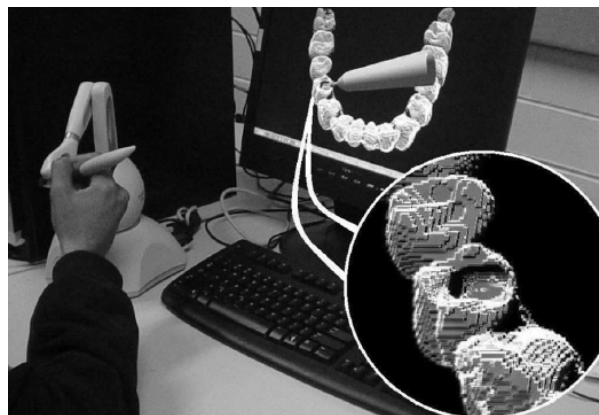


Figura 2.7. Simulador para maquinado dental desarrollado por Arbabtafti, et al. (2011) [46].

Las tecnologías hápticas se han estudiado en muchas disciplinas como ingeniería (diseño, manufactura, mecanizado, y entrenamiento) y medicina (planeaciones quirúrgicas y entrenamiento). La planeación y evaluación de ensambles es una de las aplicaciones de ingeniería que pueden ser ampliamente exploradas al utilizar dispositivos hápticos en un ambiente virtual. La manipulación háptica de objetos virtuales reduce la brecha entre la realidad virtual y el mundo real, ya que permite una manipulación de objetos más natural durante el proceso de ensamble.

## 1.3 Ensamblés

### 1.3.1 Sistemas de ensamble virtual

#### *Definiciones*

Dentro del área de manufactura virtual una gran parte de los trabajos se enfocan en el estudio del proceso de ensamble. Existen diversas definiciones de ensamble virtual (EV), por ejemplo, Jayaram, *et al.* (1997) [2] definen el ensamble virtual como “... el uso de herramientas de computación para asistir o proveer al usuario con decisiones de ingeniería relacionadas el ensamble a través del análisis, modelos predictivos, visualización y presentación de datos sin la realización física del producto o procesos de soporte”.

Por otro lado Seth, *et al.* (2011) [48] lo definen como “... la capacidad de ensamblar representaciones virtuales de modelos físicos en un ambiente virtual inmersivo generado por computadora al simular el comportamiento realista del entorno, la dinámica de los objetos virtuales y la interacción entre las partes, reduciendo así la necesidad de construir prototipos físicos para probar el ensamble, resultando en mejores decisiones de diseño y ensamble”.

Finalmente, Xia, *et al.* (2013) [13] definen el EV como “... utilizar tecnología de realidad virtual, gráficos por computadora, inteligencia artificial, teoría y métodos de ensamble para

construir un modelo virtual de un producto, el ambiente virtual del espacio de trabajo y la distribución del ensamble con el fin de simular y analizar de manera interactiva el resultado del diseño del producto y el proceso de ensamble”.

#### *Importancia del ensamble virtual*

Durante el proceso de diseño de un nuevo componente, la tarea de ensamble generalmente se deja como la última parte del proceso, asignando esta tarea a un experto que determina, con base a su experiencia y lo que ya está hecho, un plan de ensamble. Esta tarea puede ser muy costosa en términos de tiempo y recursos, además no siempre lleva a soluciones eficientes. Las técnicas de prototipado rápido proveen una manera física de evaluar y planear el ensamble de componentes antes de crear un producto definitivo. Sin embargo, modificar y construir prototipos físicos requiere de tiempo, además de que es un proceso de baja eficiencia y de alto costo (Seth, *et al.* 2011 [48], Xinhua, *et al.* 2011 [49]).

Los sistemas CAD también permiten realizar planeación de ensambles, pero en estos sistemas la experiencia y conocimientos son difíciles de interpretar de manera intuitiva y algunos factores como la evaluación de la calidad, distribución del espacio de trabajo y la ergonomía no pueden ser tomados en cuenta durante la evaluación del ensamble (Xia, *et al.* 2013 [13]). La falta de restricciones físicas que el usuario sí experimenta en el mundo real como colisiones e interferencias entre objetos es otro de las desventajas de tales sistemas. El uso de restricciones de movimiento para reducir los grados de libertad de la pieza manipulada y producir movimientos precisos puede ofrecer una potencial solución al problema, sin embargo, para lograr esto la posición y orientación final del ensamble deben de ser conocidas con anterioridad a la realización del ensamble.

Las técnicas de realidad virtual han demostrado ser una herramienta de utilidad en el diseño de productos. La realidad virtual provee un ambiente intuitivo donde el diseñador puede interactuar fácilmente con las partes y llevar a cabo revisiones de diseño sin la necesidad de utilizar prototipos físicos. Para lograr una planeación de ensamble exitosa se debe considerar la intuición y la interacción del humano (Jayaram, *et al.*, 1999 [50]). Los sistemas de ensamble virtual pueden también habilitar la generación de planes de ensamble de manera automática y la optimización del proceso de diseño a través del diseño basado en el ensamble (DFA) (Lim, *et al.*, 2007 [16]). La realidad virtual permite al usuario interactuar con modelos CAD tridimensionales con el mismo número de grados de libertad que los objetos reales. El ensamble virtual permite al diseñador evaluar si existen limitaciones al momento de ensamblar piezas para construir un componente, y así poder realizar cambios en el diseño que permitan mejorar el proceso de ensamble, ahorrando recursos al no fabricar partes de prueba o prototipos que resulten poco útiles. Además el uso de realidad virtual facilita la integración y participación de todo el equipo de trabajo en el proceso de diseño.

De acuerdo a Zhu, *et al.* (2010) [51], los pasos generales del proceso de planeación de ensambles virtuales son: (i) modelado del producto en un sistema CAD, (ii) ensamble interactivo; (iii) generación automática de un plan del proceso de ensamble basado en la



experiencia; y (iv) el empleo del plan de ensamble en un proceso real. La actividad fundamental relacionada al proceso de planeación de ensambles es el ensamble interactivo.

### *Aplicaciones*

Cada una de las plataformas de ensamble desarrolladas por diversos autores tienen objetivos diversos, por ejemplo, el sistema presentado por Simmons (1998) [52] ofrece como resultado de la planeación del ensamble un conjunto de instrucciones que describen precisamente la secuencia de ensamble, los métodos y materiales de unión, herramientas, moldes y cualquier método de prueba e inspección relevante. Se establece también que cualquier herramienta de ensamble virtual requiere de incluir al menos dos atributos relevantes del proceso: la secuencia de ensamble y los métodos de unión entre las partes. Por su parte Yang, *et al.*, (2004) [53] presentan una propuesta de planeación interactiva de ensamble. El sistema incluye una interfaz intuitiva para desarrollar tareas de ensamble directamente en un mundo virtual. El planeador interactivo extrae la información del ensamble mecánico y determina trayectorias de movimiento utilizando una red neuronal; este método es capaz de producir secuencias de ensamble alternativas automáticamente y generar instrucciones para programación robótica.

El estudio realizado por Shen, *et al.*, (2005) [54] evalúa el efecto de las tolerancias utilizando el caso de ensamble “perno en agujero” (*peg in hole*), además proponen un nuevo algoritmo para operaciones de ensamble basado en tolerancias. De manera similar, Shen, *et al.*, (2005) [55] estudian las tolerancias de ensamble basando su investigación en las características de ranuras y bordes de modelos tridimensionales. Sung y Ritchie (2009) [56] presentan dos herramientas virtuales de ensamble llamadas COSTAR (*Cable Organization System Trough Alternative Reality*), Figura 1.8a, y HAMMS (*Haptic Assembly Manufacturing and Machining System*), Figura 1.8b, utilizando estos sistemas se analizan los ensambles de un arnés y una bomba de aceite virtual en cada sistema respectivamente con fines de planeación y evaluación en la etapa de diseño, la principal contribución de este trabajo es el estudio de movimientos y trayectorias de manera gráfica en el mismo ambiente virtual.

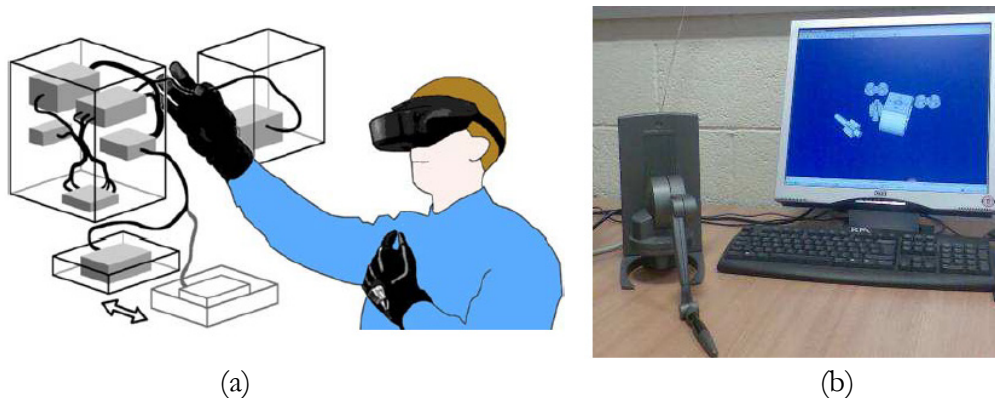


Figura 2.8. Sistemas de planeación de ensambles, (a) COSTAR y (b) HAMMS presentados por Sung y Ritchie (2009) [56].

El sistema VTS (*Virtual Training Studio*) (Brough, *et al.* 2007) [57], Figura 1.9 es un ambiente virtual que permite el entrenamiento de supervisores en operaciones de ensamble. El sistema está enfocado en el entrenamiento cognitivo, de tal modo que los entrenados pueden aprender a reconocer partes, secuencias de ensamble y orientaciones durante la operación. El sistema utiliza tres modos de funcionamiento: simulación interactiva, animación tridimensional y video. Las características clave del sistema son: manejo de simetrías, detección de errores, animaciones dinámicas, anotaciones de instrucciones ambiguas, generación automática de instrucciones en texto para el entrenamiento.



Figura 2.9. Escena virtual del sistema VTS desarrollado por Brough, et al. (2007)[57].

Fischer, *et al.*, (2009) [58] presentan una plataforma de realidad virtual que permite realizar tareas comunes de diseño de productos, como la evaluación de prototipos, ensambles virtuales y visualización de resultados de análisis de ingeniería. Mediante una propuesta similar, Lim, *et al.*, (2009) [59] desarrollan el proyecto VARP (ensamble virtual y prototipos rápidos), el cuál es la integración de diferentes plataformas de simulación para procesos de ingeniería (HAMMS, COSTAR y RPBioX), Figura 1.10. El sistema VARP permite al usuario descomponer interactivamente el modelo, evaluar y realizar cambios en el diseño, analizar la tarea de ensamble, y producir planes de manufactura y ensamble. Como primer paso, el usuario es capaz de observar el modelo en un ambiente virtual y descomponerlo en elementos más pequeños para que puedan ser manufacturados por una máquina de prototipos rápidos, esos elementos pueden ser analizados mediante la herramienta DFRP (diseño para prototipos rápidos), la cual analiza cada una de las geometrías en busca de posibles problemas de manufactura. Una vez realizado el análisis, las piezas pueden ser hechas una por una para su posterior ensamble, el sistema genera como salida las instrucciones para ensamble manual o robótico.

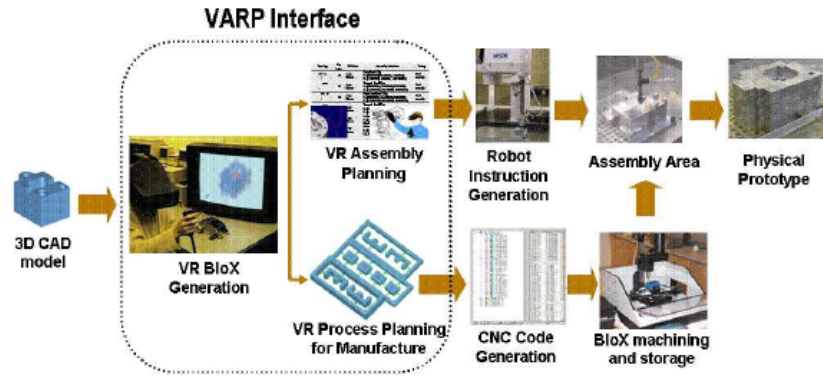


Figura 2.10. Sistema de ensamble virtual y prototipado rápido VARP, desarrollado por Lim, et al., (2009) [59].

Diversos autores han desarrollado e investigado múltiples plataformas de ensamble virtual utilizando diferentes técnicas, por ejemplo, Iacob, *et al.*, (2011) [60] y Sato, *et al.*, (2011) [61] estudian el ensamble automático por reconocimiento de características geométricas, Figura 1.11 ; Zaldivar-Colado y Garbaya, (2009) [62], Gutierrez, *et al.*, (2009) [63] y Tching, *et al.*, (2010) [64] emplean el modelado basado en restricciones; Lim, *et al.*, (2007) [16], Aleotti y Caselli, (2011) [65] y Xia, *et al.*, (2012) [66] emplean el modelado basado en la física; Jayaram, *et al.*, (1999) [50], Chen, *et al.*, (2010) [67] y Cheng-jun, *et al.*, (2010) [68] utilizan la extracción y uso de restricciones de ensamble de sistemas CAD; Finalmente Ritchie, *et al.*, (2008) [69], Bordegoni, *et al.*, (2009) [70], Christiand y Yoon, (2011) [71] y Ji, *et al.*, (2011) [72] investigan el uso de sistemas de ensamble háptico. Una breve descripción de las características principales de estos y otros sistemas se presenta en la Tabla 1.1.

De manera general se identifican cuatro aplicaciones principales para los sistemas de ensamble virtual: (i) planeación y optimización de trayectorias y secuencias de ensamble manuales o robóticas; (ii) análisis de diseño basado en el ensamble (DFA); (iii) evaluación, planeación y análisis de tareas de mantenimiento, y (iv) entrenamiento en procesos de ensamble repetitivos.

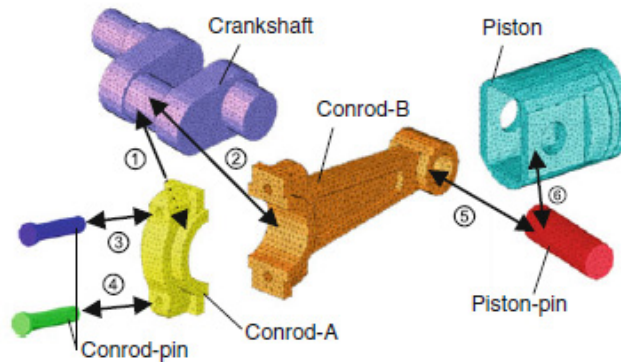


Figura 2.11. Ensamble por reconocimiento de geometrías, Sato, et al., (2011) [61].

Tabla 2.1. Plataformas de ensamble virtual.

| Sistema (auto, año)   | Características principales  |
|---|--|
| VADE [50] (Jayaram, 1999)   | Hace uso de herramientas como desarmadores, pinzas etc.; Está ligado de manera permanente a un sistema CAD; Permite al usuario realizar cambios en el diseño. Seguimiento y edición de trayectorias por medio de representación tipo barrido de volumen.   |
| HIDRA [20] (Coutee, 2001)   | Integra la retroalimentación háptica en un ambiente de simulación del (des)ensamble; Utiliza dos dispositivos hápticos para la manipulación de partes utilizando dos dedos.  |
| VDAS [72] (Ji, 2002)  | Permite el diseño, modificación y ensamble de productos mecánicos; Base de datos con partes de sujeción estandarizadas; No utiliza háptico, utiliza un guante de realidad virtual.   |
| MIVAS [73] (Wan, 2004)  | Emplea técnicas de optimización para operaciones de ensamble; Registra los movimientos del usuario y utiliza comandos por voz; Inmersión gráfica completa; Interacción de la mano virtual de forma realista para sujetar las piezas; Detección de colisiones en tiempo real y retroalimentación de la fuerza; Documentación del resultado del plan de ensamble.                          |
| SHARP [74] (Seth, 2006)   | Capacidad de crear sub-ensambles; Barrido de volúmenes; Módulo de red para ensamble colaborativo; Soporta diversos sistemas de realidad virtual como CAVE, HMD, muros de proyección, etc.  |
| HAMMS [75] (Lim, 2007a)   | Sistema de ensamble virtual basado en física; Un dispositivo háptico para manipular objetos; Utiliza PhysX para el comportamiento dinámico de los objetos y la detección de colisiones.  |
| COSTAR [76] (Ritchie, 2007)   | Sistema para la planeación de diseño y ensamble de arneses de cables. Gráficos inmersivos por medio de un <i>Head Mounted Display</i> (HMD); utiliza dos guantes para la manipulación sin retroalimentación de fuerza; Registra todos los movimientos del usuario.   |
| VEDAP-II [77] (Garbaya, 2009)   | Orientado a planeación y evaluación de ensambles; Enfocado en el modelado del comportamiento dinámico de las partes durante las operaciones de ensamble virtual. Múltiples grados de libertad de retroalimentación de fuerza.  |
| Mixed reality application for MA [70] (Bordergoni, 2009)                      | Utiliza tecnologías de bajo costo; Ensamble con las dos manos; Proyección a escala real; Sistema de rastreo de la posición para cambiar el punto de vista del usuario; Utilizado para demostrar procedimientos de ensamble; Retroalimentación de fuerza y torque 6DOF.   |
| HIVEx [19] (Bhatti, 2009)   | Utilizado para aprendizaje cognitivo y entrenamiento del ensamble virtual; Aprovecha el procesamiento en múltiples procesadores; HMD y dispositivos hápticos acoplados a un guante de RV; Cuatro niveles de entrenamiento; Utiliza el modelado basado en física.   |
| VCG [64] (Tching, 2010)   | Orientado a la planeación y entrenamiento del ensamble; Utiliza guías de restricción cinemática; Utiliza referencias virtuales; Activación de restricciones cinemáticas en tiempo de ejecución.  |
| IMA-VR [63] (Gutierrez, 2010)   | Sistema de entrenamiento virtual para transferencia de habilidades motoras y cognitivas; Combina retroalimentación háptica y visual; Comportamiento dinámico de los objetos.   |
| VA system based on force feedback [68] (Cheng-jun, 2010)                      | Estudia las interacciones entre piezas basado en la retroalimentación de fuerza; Modulo para transformar datos CAD a archivos con datos de la parte y de las restricciones de ensamble; Valida que los componentes puedan satisfacer las restricciones de ensamble pre-definidas.  |
| GCVAE [78] (Hu, 2010)   | Soporta escenarios complejos; Ensamble colaborativo; Restricciones de ensamble importadas del sistema CAD; Los modelos gráficos y de detección de colisiones son utilizados para llevar a cabo revisión de interferencias de manera dinámica.  |
| Simulador MAD [79] (Hassan, 2010)   | Simulador para procesos de ensamble/desensamble en operaciones de mantenimiento; Provee la trayectoria de ensamble óptima mediante guiado háptico y secuencias de ensamble; Orientado a la planeación de secuencias de ensamble.   |
| Virtual assembly system with an optimized haptic path [71] (Christiand, 2011) | Algoritmos de ensamble óptimos para permitir la interacción háptica durante el ensamble virtual; Soporta tres modos de simulación: Sin guías, guía de secuencias y guiado de secuencias por medio del háptico; Permite al usuario alcanzar un proceso de ensamble óptimo al seguir una secuencia y trayectorias de ensamble háptico óptimas.   |
| Physics-based virtual reality for task learning [65] (Aleotti, 2011)          | Explora los beneficios del modelado basado en física para el aprendizaje automático de tareas de ensamble y planeación de secuencias de desensamble; Metodología para encontrar todas las posibilidades de sub ensamble; Algoritmo para evaluar la estabilidad de los objetos manipulados.   |
| HVAS[18] (Xia, 2011)  | Integración automática de datos para transferir información referente a la topología, geometría, ensamble y datos físicos desde un sistema CAD; Combina el modelado basado en física con el modelado por restricciones para realizar ensambles de manera realista; Utiliza un dispositivo háptico Phantom. Utiliza estereovisión para visualización tridimensional del ambiente virtual. |
| HITsphere system [66] (Xia, 2012)   | Ambiente virtual inmersivo, con capacidad para realizar caminado en el ambiente virtual; Manipulación libre de objetos virtuales; Interfaz para integración de datos automática; Reconstrucción de datos de restricciones para construir el ambiente de ensamble virtual.  |

### 1.3.2 Sistemas de ensamble virtual háptico

El uso de sistemas hápticos resulta en un gran apoyo a los sistemas de realidad virtual, pues el diseñador puede ver, tocar y manipular las partes que diseña sin necesidad de construir prototipos físicos. El uso de dispositivos hápticos puede mejorar el desempeño de las plataformas de ensamble virtual. Esto ha sido demostrado por diversos autores, por ejemplo, Gupta, *et al.* (1997) [80], evalúan la facilidad de maniobra y ensamble mediante un ambiente virtual. En este mismo trabajo se presenta el sistema VEDA (*Virtual Environment Design for Assembly*), en el cual el usuario puede observar representaciones virtuales de los objetos y escuchar sonidos cuando los objetos chocan. El sistema permite sentir y manipular los objetos a través de una interfaz háptica utilizando dos dedos, permite interacciones dinámicas entre objetos virtuales y utiliza representaciones en dos dimensiones de los objetos con el fin de mantener tasas de actualización altas en la simulación. Como caso de evaluación utiliza el ensamble de perno y agujero. Se reporta que el tiempo de ensamble virtual es aproximadamente dos veces mayor que el tiempo de ensamble real, sin embargo las tendencias en la variación en el tiempo de ensamble virtual con respecto a parámetros como fricción, tolerancias, distancias de manipulación y redondeo de bordes son iguales a las tendencias de variación del tiempo del ensamble real. Se concluye que el uso de un sistema con retroalimentación de fuerza acerca más a la realidad la situación simulada.

Un sistema de diseño asistido por computadora ligado a un dispositivo háptico es descrito por Gutierrez, *et al.*, (1998) [14]. El objetivo de esta aplicación es la simulación de operaciones de ensamble y mantenimiento. El usuario puede interactuar con modelos tridimensionales de una manera más realista que los sistemas CAD tradicionales, pues además de ver los objetos también puede tocarlos y moverlos en el espacio virtual. Asimismo habilita la detección de colisiones con otros objetos virtuales, los cuales pueden ser importados desde otro sistema CAD mediante los formatos STEP o VRML. El sistema permite el ensamble utilizando dos métodos: el ensamble de dos objetos mediante tres pares de superficies coincidentes o la inserción de perno-agujero.

HIDRA (*Haptic Integrated Dis/Reassembly Analysis*) (McDermott, *et al.*, 1999 [81]), Figura 1.12, es una aplicación que integra la retroalimentación háptica en un ambiente de simulación para procesos de ensamble y desensamble. Los autores establecen que el proceso de (des)ensamble es un proceso tanto visual como táctil y requiere de la navegación en ambientes tridimensionales complejos, además debe existir una extensa relación entre el razonamiento y la realización de la tarea. La simulación del ensamble con retroalimentación háptica brinda a los diseñadores información táctil tridimensional con la cual se pueden generar secuencias de (des)ensamble de una manera más efectiva. El sistema HIDRA utiliza dos dispositivos hápticos Phantom, de modo que el usuario puede utilizar los dedos índice y pulgar, uno en cada dispositivo, para manipular los objetos virtuales. Utiliza el formato VRML para importar las piezas o incluso un ensamble completo. La representación háptica de los objetos es mediante geometrías primitivas como cilindros, cubos, conos, etc. Además, se utilizan diversos modos para la optimización del proceso de simulación: i) creación dinámica de representaciones

hápticas, es decir, solo se crean los objetos hápticamente cuando el cursor está cerca de ellos; ii) procesamiento de objetos, utiliza representaciones primitivas para objetos que no están en movimiento, reduciendo así la carga computacional; y iii) Disminución de la frecuencia con que se actualiza la posición de objetos en movimiento, ya que la percepción kinestésica del brazo humano responde a una tasa de actualización de alrededor de 10Hz. Con estas técnicas de optimización se logra una tasa de actualización total de hasta 2kHz.



Figura 2.12. Sistema de ensamble virtual HIDRA desarrollado por McDermott, *et al.*, 1999 [81].

Chen E. (1999) [82] identifica tres requerimientos clave que debe tener un dispositivo háptico utilizado en tareas de análisis de mantenimiento: 1) permitir a los usuarios mover su mano de manera libre sobre todo el espacio de trabajo; el cuál debe ser lo suficientemente grande 2) habilitar el sentido del tacto, y 3) tener seis grados de libertad activos, es decir, todos con retroalimentación de fuerza y/o torque. Estas y otras características han sido identificadas en el dispositivo Phantom Premium 6DOF, un dispositivo con gran espacio de trabajo, y desempeño aceptable para aplicaciones de prototipado virtual.

Bloomfield, *et al.*, (2003) [83], llevaron a cabo una comparativa de diferentes interfaces para ensambles virtuales. En esta investigación se muestra la ventaja, en términos de tiempo de ensamble, que tiene el utilizar retroalimentación de fuerza sobre aplicaciones que no involucran dispositivos hápticos. Por otro lado, Ritchie, *et al.*, (2004) [76] presentan el sistema HAMMS (*Haptic Assembly, Manufacturing and Machining System*) para la planeación de ensambles asistido por dispositivos hápticos. El sistema permite al usuario realizar ensambles mediante el sentido del tacto y la cinestesia, además registra todos los movimientos del usuario mientras lleva a cabo el ensamble. Los datos registrados son graficados como un perfil dependiente del tiempo describiendo el movimiento junto con la orientación, posición y velocidad mediante el estudio de tiempos y movimientos de Gilberth, Figura 1.13. Los movimientos innecesarios pueden ser identificados y eliminados para optimizar el plan de ensamble el cual puede ser utilizado en

tareas de ensamble reales. El sistema HAMMS incluye un dispositivo háptico Phantom Omni y el simulador físico Novodex para obtener el comportamiento dinámico a los objetos.

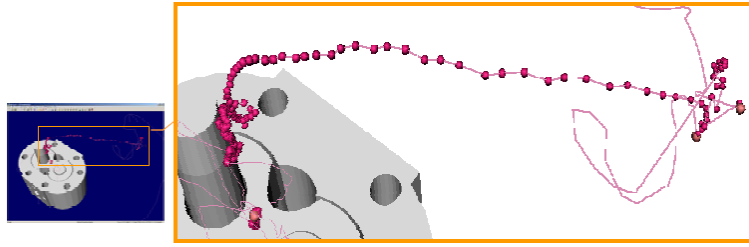


Figura 2.13. Estudio de movimientos en el proceso de ensamble, Ritchie, et al., (2004) [76].

Seth, Su y Vance (2005) [84] proponen un sistema de bajo costo capaz de importar modelos CAD complejos para su visualización tridimensional y retroalimentación háptica, Figura 1.14. El sistema permite realizar tareas de ensamble en un ambiente virtual de escritorio y además posee un módulo para internet. Incluye la detección de colisiones entre objetos virtuales y el comportamiento dinámico. El módulo de comunicación de red permite que la plataforma se conecte con otros sistemas de realidad virtual en diferentes locaciones geográficas para la demostración y análisis de las secuencias de ensamble utilizando un canal de red no exclusivo. La retroalimentación de fuerza es llevada a cabo mediante un dispositivo háptico Phantom Omni. El sistema utiliza dos archivos de entrada: un archivo CAD para la representación gráfica y un archivo STL para la representación háptica. La detección de colisiones y retroalimentación de fuerza para el dispositivo háptico son calculadas utilizando el método *Voxmap PointShell* (VPS), el cual está basado en la representación de modelos por medio de voxels. Se utilizan diferentes tamaños de voxel para cada una de las partes a ensamblar; las partes pequeñas requieren tener baja tolerancia para el ensamble, por lo tanto utilizan tamaños de voxel pequeños, las partes grandes utilizan tamaño de voxel grandes. El sistema puede ser utilizado en una gran variedad de dispositivos como monitores de escritorio, displays montados en la cabeza (HMD), muros de proyección y sistemas inmersivos.



Figura 2.14. Sistema de ensamble virtual háptico propuesto por Seth, Su y Vance (2005) [84].

Los sistemas automáticos para planeación de trayectorias en el área de robótica, ensamble y control numérico empleados en la actualidad presentan algunos problemas, por ejemplo, la generación de trayectorias libres de colisiones. He y Chen (2009) [85] sugieren que la intuición humana puede ser utilizada para facilitar el proceso de planeación de trayectorias de ensamble. Con este principio desarrollaron una aplicación en donde se combinan el instinto humano con las herramientas informáticas para la planeación semiautomática de trayectorias que permite la manipulación eficiente de un robot con seis grados de libertad.

La posibilidad de utilizar un ambiente virtual háptico de escritorio y el modelado basado en física como una herramienta de diseño para evaluar operaciones de ensamble, es investigada por Howard y Vance, (2007) [86]. El sistema desarrollado brinda retroalimentación háptica en las dos manos utilizando dos dispositivos hápticos Phantom Omni debido a su relación costo beneficio, pues tiene 6 grados de libertad de entrada y 3 de retroalimentación de fuerza.

En el trabajo presentado por Jayaram, *et al.*, (2007) [87] se utiliza el sistema VADE (*Virtual Assembly Design Environment*), el cual consta de gráficos con proyección estereoscópica y capacidad de ensamble con dos manos. El propósito es analizar dos casos de estudio de ensambles industriales reales con la finalidad de demostrar si la representación virtual de estos ensambles puede aportar información en áreas como ergonomía, distribución del ensamble, planeación del proceso, etc.

La evaluación del desempeño de una tarea de ensamble en un ambiente virtual utilizando retroalimentación de fuerza háptica es llevada a cabo por Lim, *et al.*, (2007) [75]. El objetivo es llevar a cabo la evaluación de factores humanos en un sistema de ensamble háptico mediante el uso de una evaluación comparativa de tareas virtuales/hápticas contra tareas industriales de referencia. Una rama de esta investigación consiste en cuantificar ensambles mediante la secuencia de operaciones requerida para construir un componente. La otra rama es el estudio de estrategias de diseño que no solo simplifiquen el producto si no que reduzca el costo del ensamble y mejore su funcionalidad y facilidad de ensamble, que es la meta del diseño para el ensamble (DFA). Los resultados sugieren que el sistema probado solo refleja aproximadamente los tiempos de ensamble requeridos en el mundo real y que ensambles con redondeo de bordes son realizados de una manera más rápida que aquellos que no los tienen.

Una aplicación para llevar a cabo ensambles virtuales en una ambiente virtual colaborativo es presentada por Iglesias, *et al.*, (2007) [88]. El simulador de ensamble háptico incluye el uso del mouse, teclado y dispositivos hápticos, reconoce las colisiones entre objetos, utiliza restricciones de ensamble y replica fuerzas en los dedos del usuario. Diferentes tipos de dispositivos hápticos se utilizan para tocar y mover los objetos, como el Phantom, Omni, con un punto de contacto, o el dispositivo GRAB, con dos puntos de contacto. En un trabajo posterior, Iglesias, *et al.*, (2008) [89], estudian un simulador de ensamble colaborativo, Figura 1.15, donde dos usuarios en locaciones diferentes llevan a cabo una tarea de ensamble utilizando cada uno un dispositivo háptico Phantom Omni. Se identifican dos retos principales: la sincronización de la escena virtual, y la provisión de fuerza de retroalimentación



confiable, estable y efectiva. Se desarrolla un algoritmo de suavizado de la fuerza con el fin de mejorar la calidad de la retroalimentación háptica bajo condiciones de red adversas.

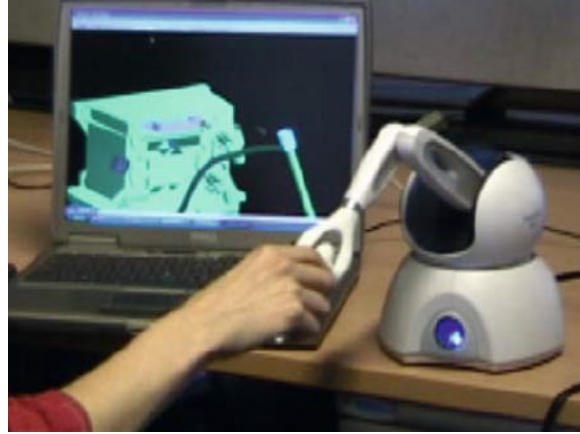


Figura 2.15. Sistema colaborativo de ensambles virtuales, Iglesias, et al., (2007) [88].

En el sistema de ensamble virtual presentado por Joshi y Jayaram, (2008) [90] se hace uso de herramientas mecánicas (llaves, desarmadores, etc.) embebidas en un sistema CAD para asistir en el análisis del ensamble y desensamble utilizando realidad virtual inmersiva y un dispositivo háptico. La secuencia y trayectorias de ensamble de partes mecánicas en un sistema de simulación de mantenimiento virtual son estudiada por Yoon, *et al.* (2009) [91] y [92]. El sistema genera la distancia más corta para cada parte durante la operación de ensamble y minimiza el número de cambios de direcciones del manipulador. Bhatti, *et al.*, (2008) [93] y (2009) [94] presentan un sistema de realidad virtual inmersivo habilitado hapticamente, el cual permite la interacción del usuario en escenarios de entrenamiento virtuales, Figura 1.16. El sistema es capaz de proveer información acerca de secuencias de ensamble, además ayuda en aprendizaje y desarrollo de habilidades mediante procedimientos predefinidos.

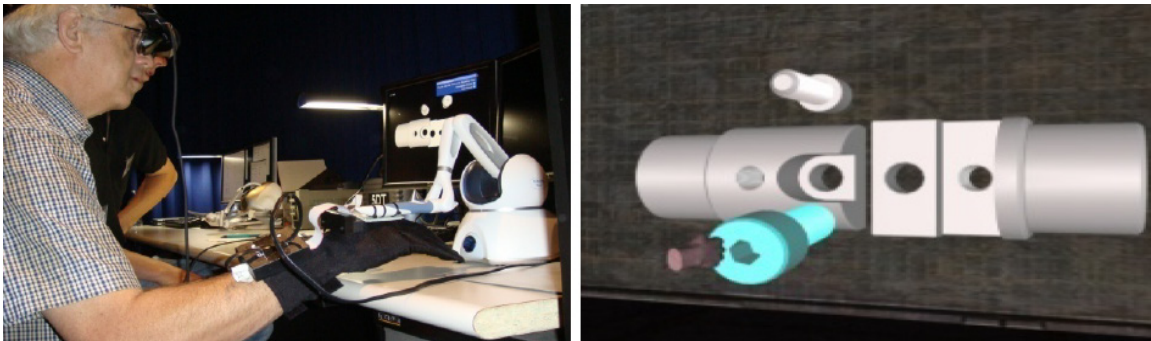


Figura 2.16. Sistema de RV inmersivo para ensambles virtuales, Bhatti, et al., (2009) [94].

Radi y Reinhart (2009) [95] proponen el sistema de guía robótica háptica con el propósito de combinar el sentido háptico humano y las capacidades de un robot industrial en el proceso de ensamble. Los resultados preliminares muestran que la integración de la retroalimentación háptica en la manipulación de robots industriales sería de gran beneficio para el proceso de ensamble ya que el usuario estaría informado acerca de las fuerzas de contacto, minimizando el riesgo de colisiones no intencionales y deformaciones de las piezas, elevando así la calidad de los productos.

Abate, *et al.*, (2009) [96] presentan una evaluación de la efectividad de la retroalimentación háptica en el entrenamiento de tareas de mantenimiento en la industria aeroespacial. El sistema implementa un ambiente interactivo en el cual cada una de las actividades principales puede ser simulada utilizando un dispositivo háptico y un módulo de simulación de realidad virtual que soporta la simulación dinámica de los objetos. Los resultados confirman el potencial del componente háptico en tareas donde la reducción del tiempo está relacionada a la habilidad háptica del técnico. Se identifican tres elementos clave de un ambiente virtual: autonomía, interacción y presencia.

Chen, *et al.*, 2010 [67] modelan un brazo robótico virtual manejado mediante un dispositivo háptico con el fin de interactuar con prototipos virtuales en un ambiente de ensamble virtual, Figura 1.17. El brazo robótico virtual es utilizado en el modelado interactivo, en el planeado de trayectorias libres de colisiones y en la planeación de trayectoria de ensamble. El usuario puede modificar interactivamente la trayectoria de ensamble mientras es guiado por las fuerzas de retroalimentación háptica.

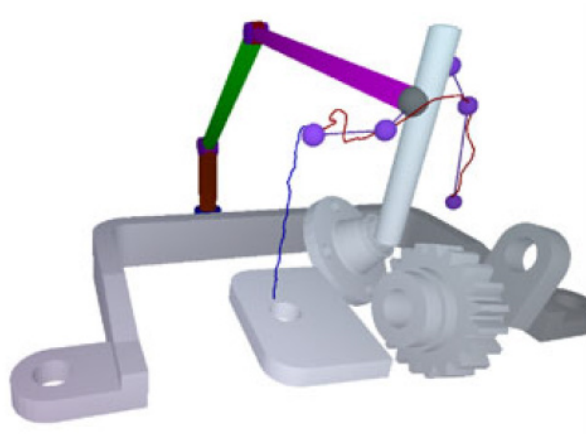


Figura 2.17. Brazo robótico virtual para ensambles virtuales desarrollado por Chen, *et al.*, 2010 [67].

Ladevez, *et al.*, (2010) [97] describen un esquema interactivo en el cual se provee al usuario con una guía háptica que describe la trayectoria de ensamble o desensamble de partes CAD tridimensionales en tiempo real. El usuario puede influenciar al planeador siguiendo o no la trayectoria.

Dos metodologías principales para crear plataformas de ensamble virtual son identificadas por Seth, *et al.*, [48] (2011) y Xia, *et al.*, (2013) [13]: modelado basado en física (PBM) y modelado basado en restricciones (CBM). El modelado basado en física (PBM) simula propiedades físicas reales, como el efecto de la gravedad, la fricción, la restitución y fuerzas de contacto para lograr el comportamiento dinámico de los objetos y la interacción de todos los objetos virtuales por medio de la respuesta a colisiones. La respuesta al contacto previene que los objetos virtuales se traslapen habilitando así el proceso de ensamble de componentes. Para lograr esto, se utilizan simuladores de física (*Physics Simulation Engines*, PSEs). En este caso los objetos se comportan tal y como lo harían en la realidad. Por otro lado, el modelado basado en restricciones (CBM) utiliza restricciones geométricas previamente definidas para ubicar los objetos en su posición y orientación de ensamble final, similar a la forma en cómo se realizan los ensambles en los sistemas CAD. Mediante este método, el ensamble es llevado a cabo reduciendo los grados de libertad de la pieza manipulada con respecto a la posición final de ensamble, resultando en un movimiento restringido del objeto. Esta metodología tiene la ventaja de un bajo costo computacional. Sin embargo, tiene la desventaja de disminuir el realismo y el nivel de interacción de la simulación.

### 1.3.3 Ensamble basado en física (PBM)

Muchos de los sistemas virtuales para planeación, evaluación y entrenamiento de ensambles utilizan el modelado basado en física para llevar a cabo el ensamble, debido a que es la técnica que ofrece un mayor nivel de realismo y por lo tanto facilita la simulación y el proceso de ensamble. A continuación se presentan algunos de los sistemas que utilizan esta técnica, así como sus principales observaciones.

Wang, *et al.*, (2001) [98] investigan los problemas del modelado basado en física en ambientes de ensamble virtual. Se enfoca en el comportamiento dinámico de la parte mientras es sostenida por el usuario con la mano. Se observa que para lograr un movimiento realista de la parte en movimiento, se requiere escalar el valor de la gravedad utilizando un factor entre 0.2 y 0.3, el cual es obtenido por prueba y error. La simulación dinámica de las partes en el ambiente virtual puede mejorar enormemente la percepción realista de los espacios virtuales. Sin embargo, cuando el usuario comienza a realizar las operaciones de ensamble, se concentra en el proceso de ensamble y no en los comportamientos individuales de cada parte, por lo tanto resta importancia a la simulación dinámica. Cabe mencionar que este tipo de interacción dinámica es solo aplicable y útil en casos de ensambles relativamente pequeños. Si la pieza es grande y/o pesada no será posible moverla de manera libre en una operación de ensamble real. Finalmente, se debe procurar que la tasa de simulación sea lo más alta posible, a pesar de que las simulaciones dinámicas tienden a disminuir la velocidad de actualización.

En el sistema desarrollado por Howard y Vance, (2007) [86], el comportamiento basado en física se logra utilizando el simulador de física ODE (*Open Dynamics Engine*), y para la detección de colisiones se utiliza OPCODE (*Optimized Collision Detection*). Los modelos pueden ser representados utilizando mallas triangulares arbitrarias. Se observa que las mallas triangulares

tienen buen desempeño al colisionar contra primitivos; sin embargo, cuando ocurren colisiones malla contra malla no se obtiene una respuesta a colisiones realista, particularmente cuando los objetos tienen contacto continuo uno con el otro, pues se observa que las dos superficies tienden a pegarse y a traslaparse. La tasa de actualización de la simulación física se encuentra directamente relacionada al número de contactos generados entre geometrías en colisión. Se utiliza una sola malla para representar objetos complejos o sub-ensambles, mientras que los primitivos se pueden utilizar para representar partes pequeñas o herramientas. Para lograr un mejor desempeño del sistema, las partes se mantienen estáticas y se cambian a dinámicas hasta que son seleccionadas. El sistema puede ser utilizado para evaluar ensambles y operaciones de mantenimiento.

Seth, *et al.*, (2008) [99] presentan el desarrollo de SHARP, un sistema basado en física para simular de forma realista la interacción parte con parte y la manipulación manual de objetos en un ambiente virtual. El sistema permite el ensamble de objetos virtuales complejos utilizando las dos manos mediante dos dispositivos hápticos.

Las animaciones basadas en física, utilizadas en aplicaciones para el aprendizaje automático de tareas secuenciales son estudiadas por Aleotti y Caselli (2011) [65], Figura 1.18. El objetivo es evaluar la estabilidad de sub-ensambles y la planeación del ensamble. El sistema desarrollado utiliza ODE y OPCODE para simular objetos dinámicos y detección de colisiones. Las pruebas presentadas fueron hechas utilizando objetos sencillos y no se compara contra ensambles reales, sin embargo, se menciona que los resultados pueden ser extrapolados para cualquier tipo de superficies.

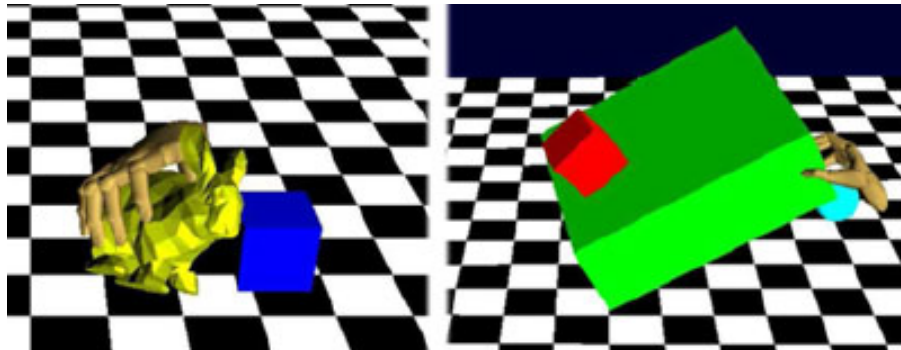


Figura 2.18. Escenario de animación basada en física, Aleotti y Caselli (2011) [65].

Seth, *et al.*, (2011) [48] identifican tres retos que los ensambles virtuales deben de superar: (i) detección de colisiones, (ii) detección de restricciones entre los objetos y (iii) optimización del modelado basado en física. Estos retos fueron identificados debido a la dificultad que representa mantener las altas tasas de actualización háptica (1kHz) y al mismo tiempo mantener una alta precisión de las colisiones y los cálculos físicos para simulaciones interactivas. Asimismo, también es difícil ensamblar partes con bajas tolerancias utilizando el

modelado basado en física. Por lo tanto, la mejora de los métodos para la interacción entre los objetos virtuales es fundamental para lograr una experiencia de ensamble virtual más realista.

Otros retos que se presentan cuando el renderizado háptico es integrado a sistemas basados en física son: la dificultad de sincronización de cada módulo (gráfico, háptico y físico), detección de colisiones poco efectiva y alto costo computacional, lo cual genera un impacto negativo en el desempeño de la aplicación (Seugling y Rölin, 2006 [100]). Estos problemas son debidos a que los simuladores físicos no están adaptados al renderizado háptico, principalmente porque la frecuencia de actualización de las simulaciones hápticas es 1kHz mientras que la actualización para la simulación física es de alrededor de 100Hz (Ritchie *et al.*, 2009 [101]; Glondu, *et al.*, 2010 [102]). Sin embargo el uso del modelado basado en física resulta en un mejor entendimiento y apreciación de la funcionalidad de las partes y puede también llevar a un entrenamiento mejorado de tareas manuales (Wang *et al.*, 2001 [98] y Zerbato *et al.*, 2011[47]).

Aunque se han desarrollado una gran variedad de aplicaciones que usan el modelado basado en física, la mayoría de los autores reportan problemas con la detección y respuesta a las colisiones, ocasionando el traslape entre los objetos virtuales o reacciones poco realistas. Estos problemas están relacionados a los simuladores físicos y a los algoritmos de detección de colisiones que se utilizan.

#### 1.3.3.1. Simuladores físicos

Los simuladores de física se utilizan para dar un comportamiento dinámico a los objetos virtuales. Estos simuladores han sido utilizados en muchas aplicaciones desde videojuegos por computadora hasta películas, y recientemente en aplicaciones de realidad virtual. En el caso de los ensambles virtuales, los simuladores de física son utilizados para crear el modelado basado en física (PBM). El PBM es una alternativa al problema de la síntesis de movimiento, ya que permite calcular la posición de los objetos virtuales en cada ciclo al simular numéricamente las leyes de la física, sin embargo tiene un alto costo computacional y presenta cierta dificultad para lograr el control de los objetos, por lo que tiene potencial para producir resultados erróneos o no exactos.

#### *Estudios relacionados*

Nourian, *et al.*, (2005) [103] presentan la aplicación de un simulador de física enfocada a la cirugía de ojos llamada xPheve, la cual permite la adaptación de la simulación física por medio de la conexión y desconexión con las leyes requeridas, por ejemplo, de acuerdo a la fuerza aplicada la simulación puede activar dos leyes diferentes: el corte o la deformación del material. Se identifica que la precisión y el desempeño de la simulación parecen estar en conflicto una con la otra. Al incrementar la eficiencia se disminuye la precisión y viceversa. Al activar o desactivar las leyes físicas dependiendo de los requerimientos de la simulación se puede mejorar significativamente la calidad y escala de las aplicaciones de realidad virtual.

En el trabajo desarrollado por Yeh, *et al.*, (2006) [104] se presenta una aplicación llamada PhysicsBench, la cual es una plataforma que utiliza el simulador ODE para simular diversos escenarios con comportamiento físico. En este trabajo se observó que hasta el 10% del total del tiempo disponible para actualizar la simulación completa puede ser utilizado para la simulación física.

Kaufmann, *et al.*, (2007) [105] presentan una aplicación de realidad virtual inmersiva, llamada PhysicsPlayground, Figura 1.19, para la enseñanza de física, la cual pretende ayudar a los estudiantes a comprender los conceptos de mecánica. La aplicación provee al usuario un mundo virtual 3D con comportamiento físico en el cual los estudiantes pueden construir y evaluar sus propios experimentos. El hardware incluye un display para montaje en la cabeza, un panel personal de interacción (PIP) y un marcador, estos dos últimos son utilizados para controlar el ambiente virtual. Utiliza formas primitivas (cilindros, cubos, esferas) y también formas arbitrarias, las cuales pueden ser cargadas en la plataforma en cualquier instante. También permite la definición de juntas cinemáticas de diferentes tipos. El sistema utiliza PhysX SDK como simulador de física, cuya precisión es suficiente para propósitos educativos.

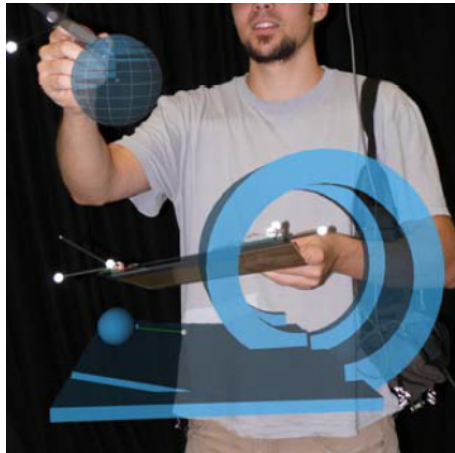


Figura 2.19. Aplicación PhysicsPlayground, Kaufmann, *et al.*, (2007) [105].

De acuerdo a Laurell (2008) [106] un simulador de física puede ser dividido en cinco subsistemas:

1. Detección de contactos, para conocer si dos objetos chocan u ocupan el mismo espacio. Existen diferentes algoritmos y métodos para conocer si dos objetos virtuales están colisionando, por ejemplo, el algoritmo *Axis Aligned Bounding Boxes* (AABB), donde se asume que los marcos que rodean a cada uno de los objetos tienen la misma orientación respecto a los ejes coordenados, o el algoritmo *Aligned Bounding Boxes* (ABB), en el cual el marco que rodea a cada objeto puede ser orientado en cualquier dirección. En ocasiones se requiere una detección de colisiones más precisa para

objetos complejos, por ejemplo una flecha, un engrane o una carcasa. Para estos casos se utilizan conjuntos de polígonos que representan la superficie de estas formas. La figura resultante será tan precisa como número de polígonos tenga, pero entre más polígonos se tendrán más bordes y vértices, los cuales necesitan ser analizados para colisiones uno por uno en cada ciclo, lo que incrementa el tiempo de cómputo. Una de las soluciones más comunes para minimizar este problema es dividir el análisis de detección de colisiones de los objetos en dos fases: (i) la fase burda, donde el objeto es rodeado por una esfera o una caja y se prueban colisiones solo para esta forma primitiva. Si se detecta una colisiones entonces se pasa a la segunda fase; (ii) la fase fina, donde se lleva a cabo una detección de colisiones del objeto más detallado utilizando bordes y vértices de la malla poligonal.

2. Resolución de contactos, para determinar la fuerza requerida para prevenir que los objetos se traslapen. El método más común es el principio del impulso y el momento, el cual se funda en la física Newtoniana. Conociendo la masa de los objetos, su momento de inercia, el coeficiente de restitución y las velocidades lineales y angulares antes del impacto es posible calcular las velocidades después del impacto.
3. Cálculo de fuerzas: existen dos fuerzas fundamentales: (i) fuerzas de contacto, las cuales son aplicadas en la superficie de contacto en cada objeto en colisión y (ii) fuerzas de campo, como la gravedad, que trabajan a distancia. Los simuladores de física deben de calcular el efecto combinado de ambas fuerzas sobre cada objeto y en cada paso de la simulación.
4. Integración del movimiento, se refiere al cálculo de la posición de los objetos en cada paso de la simulación, basado en las fuerzas que actúan sobre cada objeto y en la segunda ley de Newton:  $f = m * a$ . La aceleración se integra dos veces para encontrar la posición de los objetos en un instante de tiempo utilizando métodos numéricos de integración.
5. Impacto del tiempo (paso de la simulación): en simulaciones dinámicas el tiempo se vuelve un factor de gran influencia, ya que se debe considerar tanto el tiempo de ejecución de un ciclo y el tiempo de respuesta del sistema. Durante un ciclo de actualización el sistema debe resolver contactos, calcular fuerzas e integrar el movimiento para manejar las entradas hechas por el usuario y generar imágenes tridimensionales en el menor tiempo posible. La tasa de actualización total del sistema, incluyendo tanto la simulación gráfica como la simulación física, debe ser de al menos 25 Hz (40 milisegundos por ciclo). Una actualización por debajo de este parámetro se percibe como lenta y sin continuidad. La detección de colisiones y la generación de gráficos tridimensionales usualmente toman la mayor parte del tiempo de procesamiento del CPU, y por lo tanto restan tiempo disponible para otros cálculos.

La detección y resolución de contactos son los puntos críticos para las tareas de ensamble, por lo tanto, se debe balancear la precisión al detectar las colisiones con la precisión en llevar a cabo respuestas físicas correctas o realistas.

*Evaluaciones Comparativas*

Choi, *et al.*, (2004) [107] evalúan el desempeño de tres métodos de integración numérica: el método de Euler, el método de Euler mejorado y el método de Runge-Kutta. La integración numérica es utilizada para determinar la posición de un objeto al calcular la posición en función del tiempo. Se utiliza la simulación de un auto en movimiento para realizar la evaluación, y se graban los cuadros por segundo de la simulación para evaluar su desempeño. Cada método fue evaluado bajo dos condiciones: una con detección de colisiones y la otra sin detección de colisiones. El ambiente de simulación sin detección de colisiones se divide a su vez en velocidad uniforme y velocidad variable. El método de Euler produjo los mejores resultados en términos de cuadros por segundo en un ambiente de simulación sin detección de colisiones.

Seugling y Rölin (2006) [100] comparan tres simuladores de física: - Newton, ODE y PhysX – con las siguientes pruebas realizadas en tiempo de ejecución: (i) fricción en un plano inclinado, (ii) fuerzas giroscópicas, (iii) restitución, (iv) precisión contra lo real, (v) estabilidad del apilado, (vi) estabilidad y escalabilidad de restricciones, (vii) estabilidad de contactos, y (viii) contactos complejos entre modelos primitivos, convexos y representados por malla triangular. En la mayoría de las pruebas el simulador mejor evaluado fue PhysX, excepto en las pruebas de estabilidad del apilado y colisiones de malla triangular contra malla triangular, donde se observaron comportamientos no deseados. Similarmente una evaluación cualitativa de simuladores de física fue realizada por Coumans y Victor (2007) [108], en la cual se comparan los simuladores PhysX, Havok, ODE y Bullet. Los criterios de comparación incluyen la detección de colisiones y las características de cuerpo rígido. Los resultados sugieren que PhysX es el simulador más completo.

Por otro lado, Boeing y Bräunl (2007) [109] realizaron una comparativa entre PhysX, Bullet, JigLib, Newton, ODE, Tokamak y True Axis. En este trabajo se desarrolla y utiliza PAL (*Physics Abstraction Layer*), el cual permite integrar y utilizar varios simuladores de física en una misma aplicación. El criterio de comparación incluye: (i) desempeño del integrador, (ii) propiedades de los materiales, (iii) fricción, (iv) estabilidad de restricciones, (v) sistema de detección de colisiones, (vi) juntas mecánicas disponibles y (vi) prueba de apilado. Concluyen que PhysX tiene el mejor método integrador mientras que Bullet provee el sistema de detección de colisiones más robusto. Adicionalmente se identifican seis características esenciales en el desempeño general de un simulador de física:

1. Paradigma del simulador. Los métodos más comunes para resolver las fuerzas de contacto son métodos basados en la penetración (penalty based methods), métodos basados en impulso (impulse based methods), y métodos basados en restricciones.
2. Desempeño del integrador. El integrador determina la precisión numérica del simulador, y es responsable de calcular la posición de un cuerpo utilizando las fuerzas que actúan sobre él. Para probar su desempeño se realiza la prueba de un objeto en caída libre, comparando los resultados obtenidos en la simulación contra resultados ideales.



3. Representación de los objetos. Comprende la forma de representar los objetos para la detección de colisiones. Aunque un objeto tenga una representación gráfica dada, la representación para detección de colisiones puede ser completamente diferente, (un engrane puede ser representado como un cilindro). La representación de los objetos contribuye a la eficiencia y precisión de las colisiones en la simulación.
4. Detección de colisiones y de contactos. Para su evaluación se mide la penetración cuando ocurren contactos de diferentes tipos entre varios objetos, e incluso entre contactos acumulativos como el apilado de cajas.
5. Propiedades del material. Considera el determinar los parámetros físicos que pueden ser reproducidos por el simulador de física (fricción, gravedad, restitución, masa, etc.). Se prueba comparando los resultados de la simulación contra movimientos ideales.
6. Implementación de restricciones. Comprende el identificar cuáles restricciones o juntas mecánicas son soportadas, y con cuanta precisión son simuladas. La estabilidad de las restricciones se prueba al formar una cadena de juntas mecánicas en la cual se unen dos cajas con una masa 400 veces más grande que el número de restricciones, se evalúa la deformación de las restricciones.

Las comparativas descritas anteriormente evalúan el desempeño de los simuladores físicos pero no consideran su integración con el renderizado háptico. Glondu, *et al.*, (2010) [102] introducen las posibilidades de implementar un sistema modular háptico que utiliza simulación háptica y física. Con este propósito evalúan cuatro simuladores físicos: Havok, PhysX, Bullet y OpenTissue. El criterio de desempeño se basa en el tiempo de cómputo, estabilidad y precisión. Se presentaron cuatro pruebas de evaluación: una pila de 40 cubos, una casa de cartas de siete niveles, 8000 cubos en un recipiente y un cubo muy pesado (mucha masa) sobre un cubo muy ligero (poca masa). PhysX mostró penetración en algunas de la pruebas, mientras que Havok mostro el mejor tiempo promedio de computo, estabilidad y precisión al representar el efecto de la fricción. Aunque el renderizado háptico fue considerado en esta evaluación, las pruebas fueron realizadas sin utilizar el dispositivo háptico. Se menciona que la frecuencia háptica se mantuvo a 1kHz en escenas de hasta 50 cuerpos rígidos dinámicos. Se muestra también un método escalonado para acoplar el renderizado háptico y el físico. Los simuladores de física evaluados utilizan métodos iterativos tipo Gauss-Seidel para resolver la penetración y las restricciones. Todos los simuladores de física presentan tiempos de cómputo y precisión compatibles con el renderizado háptico.

#### *Integración con el renderizado háptico*

Un método para integrar el renderizado háptico con simuladores de física es presentado por Chan y Choi (2009) [110]. La plataforma desarrollada tiene su aplicación en la cirugía virtual. Se presentan dos metodologías para determinar la fuerza de retroalimentación correspondiente a la detección de colisiones: (i) fuerza basada en el error, la cual utiliza la profundidad de penetración, el punto de contacto, la normal unitaria del punto de contacto y la rigidez de los dos objetos en colisión para calcular la fuerza de reacción utilizando la ley de Hooke, y (ii) fuerza obtenida por plano de contacto, en este método la interfaz háptica es restringida a

moverse solo en un lado del plano de contacto, cualquier penetración a través del plano será resistida por una fuerza que está relacionada a la profundidad de la penetración y la rigidez del material. Para la manipulación háptica de los objetos virtuales se acopla un resorte virtual al cursor háptico con el fin de controlar el objeto y obtener la retroalimentación de fuerza. Se observó que la rapidez de los simuladores de física no es suficiente para la tasa de actualización requerida para un renderizado háptico realista, traduciéndose en oscilaciones del objeto controlado al acercarse a un estado estable o estático.

Diversos autores como, Howard (2007) [86], Chen, (2010) [111] y Zerbato D, (2011) [47] han identificado la importancia de representar lo más realista posible el comportamiento físico de los objetos virtuales en tareas de ensamble y/o entrenamiento. Existen diversos simuladores físicos como PhysX, Bullet, Open Dynamics Engine, Havok, Newton, etc., sin embargo el uso y desempeño de este tipo de simuladores en aplicaciones que incluyen sistemas hápticos es un tema que debe estudiarse con más detalle.

### 1.3.3.2. Detección de colisiones

Cuando ocurre una colisión entre dos objetos el simulador físico debe calcular las fuerzas de reacción que serán aplicadas a ambos objetos. La reacción de los objetos está gobernada por las ecuaciones diferenciales de movimiento que son derivadas de las leyes de movimiento de Newton o de la mecánica Lagrangiana (Vance y Dumont, 2011, [34]). De esta manera la detección de colisiones es un aspecto clave en el procedimiento de ensamble virtual y está directamente relacionada a la forma en que el modelo u objeto virtual se representa dentro del simulador de física. El uso de modelos con poca precisión geométrica puede resultar en una simulación deficiente del proceso de ensamble virtual, incluso en un ensamble que no puede ser llevado a cabo en el ambiente virtual pero sí en el ambiente real (Chen, 2010 [111]). Las tareas de ensamble pueden incluir objetos con formas diversas, los cuales pueden ser clasificados en dos grupos: objetos convexos Figura 1.20a y objetos cóncavos (no convexos) Figura 1.20b, siendo los últimos los más comunes en las tareas de ensamble y los que presentan mayor dificultad para la detección de colisiones.

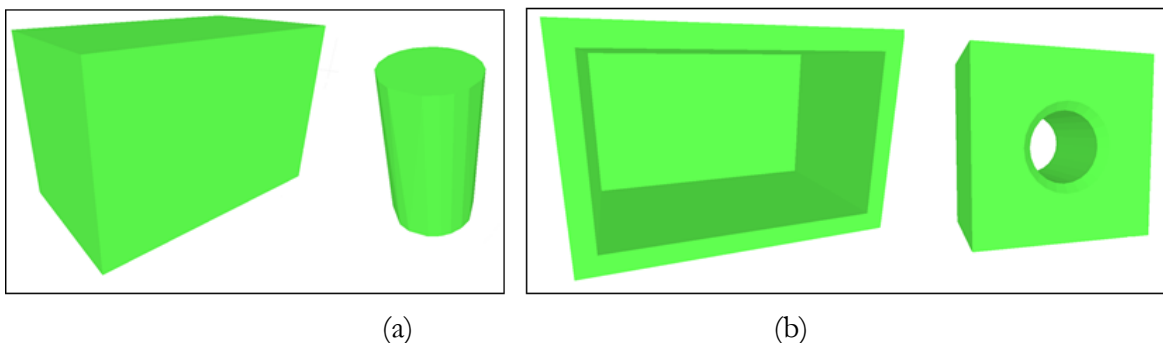


Figura 2.20. Clasificación de modelos: (a) objetos convexos, (b) objetos cóncavos (no convexos).

Cuando se requiere utilizar un modelo CAD en aplicaciones de realidad virtual, el intercambio de datos es difícil debido a la relativa complejidad matemática de las funciones polinomiales que definen la geometría del modelo (Tching, *et al.*, 2010 [64]). Con el fin de utilizar los modelos CAD en aplicaciones de realidad virtual se utilizan las representaciones teseladas las cuales aproximan el modelo mediante una malla poligonal, o malla triangular, que cubre la superficie del modelo. Esta conversión puede ser hecha exportando el modelo CAD en formatos de representación por malla, como STL, OBJ, IGES, etc. El problema principal al utilizar representaciones teseladas en tareas de ensamble es que la descripción poligonal resultante no siempre tiene suficiente precisión geométrica como el modelo CAD original, esto debido a la pérdida de algunas características geométricas (Tching, *et al.*, 2010 [64]). La precisión o calidad de la representación dependerá de la resolución dada. En la Figura 1.21 se muestra el modelo de una esfera en donde se puede apreciar que una baja resolución resulta en una forma diferente a la esfera (Figura 1.21a) y conforme se incrementa la resolución el modelo tiene mayor precisión (Figura 1.21b y 1.21c).

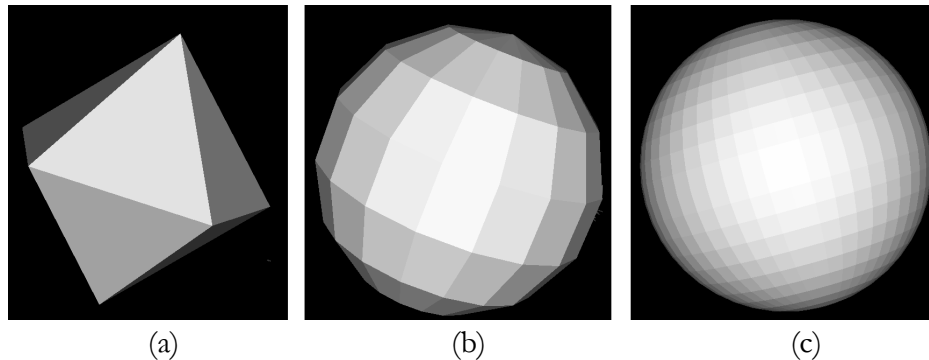


Figura 2.21. Esfera teselada: (a) baja resolución, (b) media resolución, (c) alta resolución.

#### *Estudios relacionados*

El uso de modelos representados por mallas triangulares (teselado) es la manera más común de crear objetos dentro de un sistema de realidad virtual, así como en los simuladores de física y en la detección de colisiones. Chandru y Manohar, (1997) [112] presentan una revisión crítica a cuatro métodos de representación de objetos con fines de detección de colisiones: Constructive Solid Geometry (CSG), Boundary Representation (B-rep), Non-manifold y modelos voxel. Cada representación es analizada observando un conjunto de atributos tales como facilidad de renderizado, destreza morfológica, heterogeneidad, facilidad de edición (local y global), requerimientos de memoria, facilidad de análisis, facilidad de obtener representaciones volumétricas desde una parte física, y precisión geométrica. Se encontró que el modelado por voxel ofrece el mejor desempeño global.

Con el fin de mejorar el desempeño de las aplicaciones de realidad virtual y el comportamiento dinámico de los objetos virtuales se han llevado a cabo diversos estudios enfocados a mejorar los algoritmos para detección de colisiones dentro de ambientes virtuales. Por ejemplo, en el

trabajo realizado por McDermott, *et al.*, (1999) [81], la detección de colisiones es utilizada para prevenir que los objetos virtuales se traslapen durante el proceso de ensamble. Se utilizan los algoritmos para detección de colisiones incluidos en el software V-Clip, los cuales solo soportan objetos convexos. Se pueden utilizar objetos cóncavos siempre y cuando sean representados por un conjunto de piezas convexas. El sistema propuesto incluye una opción que permite al usuario descomponer objetos no convexos en partes convexas al seleccionar un conjunto de vértices del objeto original. En este sistema la retroalimentación háptica es realizada en base a representaciones primitivas de los objetos, por ejemplo cilindros, cubos, conos, esferas, etc.

La representación por voxel es analizada por Lock y Wills (2000) [113], donde se mezclan dos representaciones del objeto: (i) una representación de volumen, llamada Voxmap, que consiste en voxels llenos o vacíos, y (ii) una capa de puntos (PointShell), la cual es definida por puntos uniformemente distribuidos sobre la superficie del modelo. Una colisión es detectada cuando un punto del PointShell del objeto manipulado se encuentra dentro de un voxel relleno del segundo objeto, Figura 1.22. Los resultados muestran una detección de colisiones rápida, la cual puede alcanzar desempeño de tiempo real y tasas de actualización para la detección de colisiones por encima de 20Hz. Sin embargo, conforme se incrementa la resolución de la capa de puntos y del mapa de voxels (voxmap) el desempeño, en términos de rapidez, disminuye.

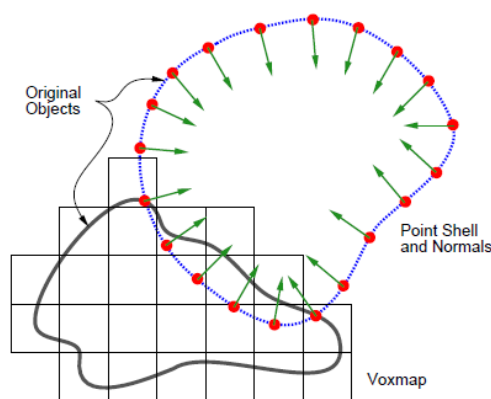


Figura 2.22. Representación en dos dimensiones de la detección de colisiones para representación por voxel, Lock y Wills (2000) [113].

Tesic y Banerjee (2001) [114], representan los objetos virtuales mediante elementos convexos que forman un modelo compuesto, el cual es definido por una malla triangular. Esta representación permite la detección rápida y exacta de colisiones para modelos no convexos, y está diseñada para objetos que tienen esquinas y bordes bien definidos.

Johnson y Willemsen (2004) [115], [116] desarrollan el algoritmo de distancias mínimas locales (LMD) para prevenir colisiones entre objetos virtuales. Sus estudios permiten al usuario marcar

las trayectorias de objetos evitando colisiones, pues el sistema genera una fuerza de repulsión al usuario mediante la interfaz háptica antes de que los objetos colisionen.

Varios programas para detección de colisiones como V-collide, PQP, SWIFT++ y VPS, que utilizan representaciones poligonales de los modelos, son analizados por Kim y Vance, (2004) [117], haciendo énfasis en aplicaciones de ensamble virtual. VPS detecta colisiones entre un objeto dinámico en movimiento y un objeto estático que se define como una colección de todos los objetos en la escena virtual, excepto el objeto manipulado. VPS incluye la capacidad de brindar el modelado basado en física a los objetos virtuales, con lo que se calcula directamente la respuesta a colisiones y el movimiento de los objetos después del choque. Se concluye que VPS es el paquete de detección de colisiones más eficiente para tareas de ensamble.

Hippman (2004) [118] presenta el algoritmo de modelo de contacto poligonal (PCM) el cual representa las superficies de un cuerpo por medio de mallas poligonales y jerarquías de volúmenes de delimitación. La detección de colisiones para volúmenes de delimitación lleva a cabo pruebas de colisión únicamente para objetos convexos en pequeñas áreas de contacto, lo que resulta en una detección de colisiones 10 veces más rápida que el método tradicional de fuerza bruta (prueba de cada uno de los vértices).

El algoritmo “*Clustered Hierarchy of Progressive Meshes*” (CHPM) propuesto por Yoon, *et al.*, (2004) [119], funciona como una representación con multi-resolución del modelo original y como un volumen jerárquico de delimitación. El algoritmo puede manejar detección de colisiones para modelos grandes o modelos con un gran número de triángulos. También es capaz de acelerar los cálculos de cómputo utilizando diferentes niveles de detalle en el área de la malla donde ocurren las colisiones.

Jimenez y Segura, (2008) [120] proponen una representación basada en poliedros para modelar objetos complejos utilizando objetos más simples sin ninguna descomposición previa en piezas convexas. A los elementos resultantes se les llama *simplices*, los cuales son obtenidos al descomponer el poliedro en elementos tetraédricos alineados.

Choi, *et al.*, (2010) [121] proponen un algoritmo eficiente, robusto y computacionalmente simple para contactos entre cuerpos de geometría compleja. El algoritmo, llamado *Triangle Soup Average Plane Contact* (TSAPC), consiste de dos partes: 1) pre-búsqueda, donde todos los triángulos que se intersectan son encontrados al utilizar una superposición recursiva entre los marcos delimitantes de los objetos, y 2) búsqueda detallada, donde se identifica la penetración, profundidad y ubicación de cada contacto individual. Como resultado se observa que aunque el número de triángulos se incrementa para aumentar la precisión de la superficie de contacto, el tiempo de solución solo incrementa un poco.

En el trabajo desarrollado por Weller, *et al.*, (2010) [122] se presenta un análisis comparativo de los algoritmos de detección de colisiones de objetos rígidos. Se analizan y comparan los algoritmos VPS (*Voxmap PointShell*) e IST (*The Inner Sphere Thre*) en términos de su desempeño

y la calidad de la respuesta a colisiones mediante la fuerza y el torque. Las fuerzas, torques, penetraciones y tiempo de cómputo son las variables medidas en cada paso de la simulación. Los algoritmos de detección de colisiones deben proveer fuerzas y torques continuos y estables, incluso en situaciones extremas como velocidades de impacto altas o grandes superficies de contacto.

#### 1.3.4 Ensamble basado en restricciones (CBM)

Un problema importante relacionado a la interacción del usuario en los sistemas virtuales de ensamble es la falta de restricciones físicas, como colisiones con los objetos, que el usuario sí experimenta en el mundo real y depende de ellas para lograr un ensamble efectivo. Aunque el ensamble basado en física trata de solucionar esta situación mediante la detección y respuesta a las colisiones, se requiere encontrar métodos más eficientes y menos costosos en términos de tiempo de cómputo, además la falta de precisión en la representación de modelos teselados y la baja eficiencia de la respuesta a colisiones provocan un bajo desempeño del sistema. Para compensar esto es necesario brindar al usuario con restricciones de movimiento para que los objetos puedan ser manipulados con precisión (Jayaram, *et al.*, 1999 [50]).

El ensamble basado en restricciones consiste en definir ciertas restricciones de movimiento que indican la posición de la pieza a ensamblar respecto a una pieza base. Estas restricciones pueden ser de diversos tipos como: concetricidad, paralelismo, distancia fija, ángulo fijo, perpendicularidad, etc. Para utilizar estas restricciones en un sistema de realidad virtual se requiere definir previamente dichas restricciones. La información requerida es comúnmente obtenida del ensamble previamente realizado en un sistema CAD.

##### *Estudios relacionados*

Con el fin de ayudar al usuario a realizar el ensamble virtual, Simmons, *et al.*, (1998) [52], proponen dos métodos: ensamble por colisión y ensamble por proximidad. El ensamble por proximidad requiere que la posición de ensamble final sea definida con el fin de aplicar restricciones de movimiento a la pieza manipulada cuando el usuario intenta ensamblar dicha pieza. Cuando el usuario manipula un objeto, el sistema determina si las posiciones de ensamble de la pieza manipulada respecto a la pieza base son cercanas, en caso positivo, el sistema pregunta al usuario si esa es la posición de ensamble final, en caso positivo se aplica el método de unión seleccionado por el usuario.

En el trabajo realizado por Jayaram, *et al.*, (1999) [50] se describen los métodos para verificar y aplicar restricciones axiales y planares. La mayoría de las restricciones en los ensambles reales involucran el alineamiento axial y/o el acoplamiento de superficies. En el modelado basado en restricciones cada restricción de la parte manipulada y la restricción correspondiente en la pieza base son monitoreadas durante la simulación para identificar si las restricciones están lo suficientemente cerca para ser aplicadas. Si se cumplen ciertas condiciones el movimiento del objeto manipulado es entonces restringido mediante la reducción de los grados de libertad y solo podrá moverse en ciertas direcciones.

Wang, *et al.*, (2001) [98] investigan los problemas del modelado basado en física en ambientes de ensamble virtual. Se enfoca en el comportamiento dinámico de la parte mientras es sostenida por el usuario con la mano y mientras la parte está restringida respecto a la pieza base. Se estudian tres tipos de movimientos que están cercanamente relacionados al modelado del ensamble virtual: (i) movimiento en el espacio libre, (ii) translación en un plano y sobre un eje, y (iii) rotación alrededor de un eje. Estos movimientos son tratados como restricciones mientras se lleva a cabo el ensamble. El mecanismo de restricciones guía al usuario en el proceso de ensamble, lo cual lleva a trazar la conclusión de que en la mayoría de los casos la simulación dinámica incrementa enormemente el sentimiento realista de ambiente virtual pero no contribuye significativamente al análisis y evaluación de ensamble. Cuando se simulan restricciones de ensamble no solo se representa la intención del diseñador, sino también se muestra el proceso mediante el cual el ensamble se realiza físicamente. Esto también provee un método directo e intuitivo para la evaluación del ensamble ya que se simula el proceso físico de ensamble.

La guía por restricciones virtuales (*Virtual Constraint Guidance, VCG*) es una técnica para realizar ensambles virtuales utilizando el CBM presentada por Tching, *et al.*, (2010) [123]. Este método utiliza tanto restricciones cinemáticas como guías virtuales para llevar a cabo un ensamble con interacción háptica. Las guías virtuales son modeladas como muros o planos virtuales que representan guías geométricas que limitan los movimientos del usuario cuando se acerca a la posición de ensamble, Figura 1.23. Las restricciones cinemáticas son juntas cinemáticas (juntas prismáticas, de bola, de pasador, etc.) que limitan el movimiento relativo de las partes después de que han sido ensambladas. Cuando se activan las guías virtuales el sistema deshabilita la detección de colisiones en el tiempo y lugar donde se está llevando a cabo el ensamble virtual. El sistema utiliza la información topológica de los objetos CAD para identificar las áreas de ensamble. Esta parte se realiza mediante un pre-proceso en el cual se definen manualmente las restricciones para cada zona de interés.

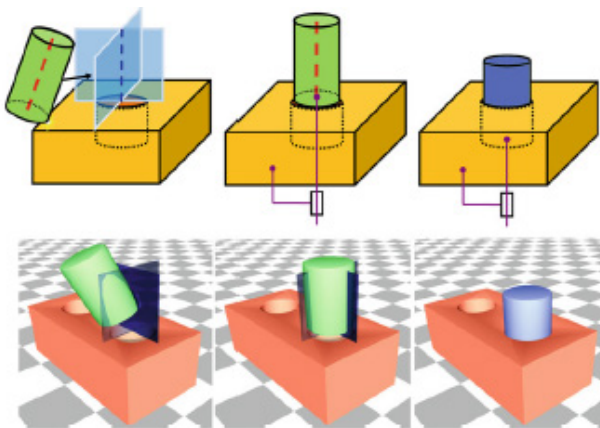


Figura 2.23. Aplicación de la guía por restricciones virtuales, propuesta por Tching, *et al.*, (2010) [123].

En el trabajo realizado por Vance y Dumont, (2011) [34] la tarea de ensamble es dividida en dos fases: (i) fase de movimiento libre, que involucra el tiempo desde que un objeto es seleccionado hasta que se detecta una colisión, y (ii) fase de posicionamiento fino, que se refiere a la inserción final del objeto. También se presenta un resumen de tres técnicas de ensamble basado en restricciones: (i) restricciones automáticas de geometría (ACG), donde las restricciones geométricas de las piezas son utilizadas para guiar el proceso de ensamble, una vez que una restricción ha sido identificada las dos partes son alineadas y los grados de libertad para el movimiento son reducidos; (ii) guía de restricciones virtuales (VCG), en el cual se utilizan guías virtuales para guiar el movimiento de los objetos con el fin de moverlos a una posición específica para después crear una restricción cinemática; y (iii) descomposición dinámica de los grados de libertad (DIOD) que está basada en detectar los cambios de velocidad de un objeto en movimiento. Cuando la velocidad es alta, el objeto se mueve con una relación 1.1 de la velocidad real, si la velocidad se disminuye entonces también la relación se disminuye.

Seth, *et al.*, (2011) [48], Patil, *et al.*, (2009) [124] y Borgergoni, *et al.*, (2009) [70] sugieren que un método híbrido entre algoritmos PBM y algoritmos CBM podría generar una simulación más real y eficiente al brindar un comportamiento natural de la pieza y ser capaz de posicionar las piezas manipuladas de manera más rápida y precisa durante el proceso de ensamble. Es estos trabajos se ha investigado la combinación del PBM y del CBM en una misma plataforma. De manera general se ha observado que se pueden simular ensambles con tolerancias muy bajas manteniendo el realismo en el comportamiento de los objetos. Se ha logrado ensamblar partes con tolerancias de hasta 0.0001% y se han detectado colisiones acertadamente con una precisión de 0.0001 mm (Seth, *et al.*, 2007 [17]).

#### 1.4 Fuerzas de contacto

La sensación de fuerzas de contacto cuando el objeto manipulado colisiona con algún otro objeto en la escena virtual es fundamental en el proceso de ensamble virtual háptico, pues dicha sensación de contacto es la que permite realizar el proceso de ensamble de manera natural. Con el fin de proveer una fuerza de reacción estable y realista el simulador físico debe proveer resolución de contactos de manera continua cuando los objetos son manipulados. Los simuladores físicos utilizan la integración numérica para moverse de un estado actual,  $y(t_0)$ , a un nuevo estado,  $y(t_0 + \Delta t)$ , donde  $\Delta t$  es el paso o tiempo de la simulación. El componente de detección de colisiones del simulador físico debe revisar cada nuevo estado para encontrar posibles intersecciones entre objetos. Si no se encuentran traslapes el sistema adopta este nuevo estado como el estado actual. Pero por el contrario, si se encuentra un traslape, el sistema debe calcular el tiempo donde ocurrió la primera colisión y mover la simulación a ese estado. Una vez que todos los puntos de contacto son determinados el sistema debe calcular las fuerzas de restricción que impiden el traslape o penetración de los objetos (Baraff, 1989 [125]).



De acuerdo a Baraff (1989) [125], se deben satisfacer tres restricciones cuando ocurre una colisión: (1) la velocidad después de la colisión debe de ser al menos  $-\epsilon$  veces la velocidad relativa al punto de contacto; donde  $\epsilon$  es el coeficiente de restitución; (2) las fuerzas de impulso en el punto de contacto pueden únicamente empujar a los objetos pero no jalarlos hacia adentro de los objetos mismos; y (3) las fuerzas de contacto ocurren solo en los puntos de contacto. Así, el resultado de una colisión entre dos cuerpos rígidos es una discontinuidad en la velocidad de los objetos, la cual puede ser modelada de manera precisa aplicando fuerzas impulsivas a los objetos, (Baraff, 1989 [125]. Estas fuerzas impulsivas pueden ser calculadas mediante métodos analíticos o métodos basados en la penalización (Ruspini y Khatib 1997) [126]. Los métodos analíticos resuelven de manera numérica las fuerzas de contacto exactas y los impulsos requeridos para garantizar que los objetos nunca se traslapen. Los métodos basados en la penalización calculan fuerzas de restauración, típicamente proporcionales a la cantidad de penetración cuando los objetos ya se han traslapado un poco.

Un algoritmo de renderizado háptico para modelos poligonales arbitrarios es propuesto por Johnson, *et al.*, (2005) [127]. El algoritmo calcula la distancia entre el modelo manipulado y el resto de los objetos en la escena virtual, con el fin de calcular la fuerza de retroalimentación háptica. Esta fuerza puede ser utilizada para prevenir el traslape entre objetos. La técnica utilizada es llamada *Spatialized Normal Cone Hierarchies* (SNCH) y tiene tres aplicaciones principales: generación de trayectorias libres de colisiones, detección de colisiones y estimación de la penetración.

Zhu, *et al.*, (2005) [128] proponen una metodología para la sujeción o agarrado de una pieza virtual por medio de cuatro dedos utilizando retroalimentación háptica en tareas de ensambles virtuales. El algoritmo VPS (*Voxmap PointShell*) es adaptado para detectar colisiones y la fuerza de sujeción se determina mediante la ley de Hooke.

Otuday y Lin (2006) [129] describen un algoritmo para renderizado háptico de 6 grados de libertad. El objetivo es lograr la interacción estable entre objetos rígidos con gran cantidad de triángulos y alta complejidad. Mediante el algoritmo de simplificación de preservación de la sensación (SPSA), que consiste en utilizar varios niveles de detalle del objeto, se logran altas tasas de actualización para la detección y respuesta a colisiones. Se utilizan también la técnica de agrupamiento de contactos, la cual limita el número de contactos a calcular al modelar cada contacto como una restricción planar.

El problema de los contactos entre partes durante la fase de emparejamiento en una tarea de ensamble es estudiado por Garbaya y Zaldivar (2007) [130]. El sistema propuesto en este trabajo calcula las fuerzas de contacto y de retroalimentación háptica al hacer la intensidad de la fuerza dependiente de la profundidad de penetración. Se utilizan dos representaciones del objeto virtual, una visible y otra invisible, las cuales están conectadas por medio de un arreglo masa-resorte-amortiguador (MRA), haciendo la penetración invisible al usuario, Figura 1.24. La fuerza calculada es renderizada al usuario a través del dispositivo háptico Phantom 6DOF. Se analiza el efecto de la sensación de fuerzas de contacto en el desempeño del usuario durante el

proceso de ensamble. Los resultados muestran un efecto signifiante de la sensación de fuerzas de contacto sobre el tiempo de terminación de la tarea. Se observó que los usuarios tienen un mejor desempeño cuando se provee retroalimentación de fuerza generada por el contacto entre objetos virtuales durante el proceso ensamble.

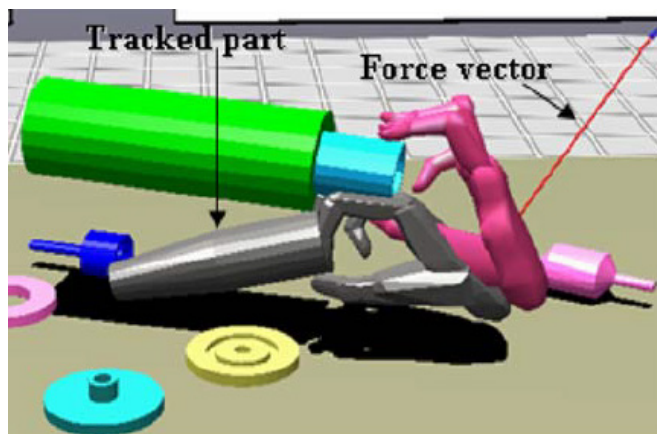


Figura 2.24. Representaciones de los objetos virtuales acoplados en el sistema desarrollado por Garbaya y Zaldivar (2007)[130].

Existen otros trabajos que buscan optimizar la estimación de fuerzas de contacto por medio del desarrollo e implementación de nuevos algoritmos, por ejemplo, Barbic y James (2008) [131] presentan un algoritmo basado en el CPU para el renderizado háptico. El objetivo es calcular los contactos entre dos modelos tridimensionales complejos, cada uno de los cuales puede tener deformaciones. Los contactos distribuidos son esenciales para el renderizado háptico de 6 grados de libertad. Al utilizar este método la detección de colisiones, el cálculo de las deformaciones, y el cálculo de las fuerzas de contacto pueden ser actualizadas en un mismo ciclo a una frecuencia de 1000Hz, igual a la frecuencia de actualización háptica recomendada.

### 1.5 Importancia de la retroalimentación de fuerza

La importancia de la fuerza de retroalimentación durante el proceso de ensamble virtual ha sido analizada por diversos autores quienes han probado el desempeño de tareas de ensamble puede ser aumentado al utilizar retroalimentación de fuerza, por ejemplo, McNeely, *et al.*, (1999) [132] proponen un algoritmo de renderizado háptico capaz de mantener una tasa de actualización de 1 kHz, sin la necesidad de ciclos de simulación física y háptica asíncronos. De este modo se soporta la manipulación de un objeto rígido dentro de un ambiente con gran cantidad de objetos estáticos, Figura 1.25. Se integra la detección de colisiones, el modelo de fuerza y el comportamiento dinámico del objeto en un solo ciclo que actualiza las fuerzas directamente a 1kHz. Para la representación de modelos utiliza el algoritmo Voxmap PointShell descrito anteriormente. El añadir fuerza de retroalimentación a los ambientes

virtuales conduce a una mejora en las restricciones físicas del sistema virtual, logrando que el usuario pueda concentrarse en problemas relativos al ensamble, como la planeación de trayectorias y la satisfacción de reglas de ingeniería.

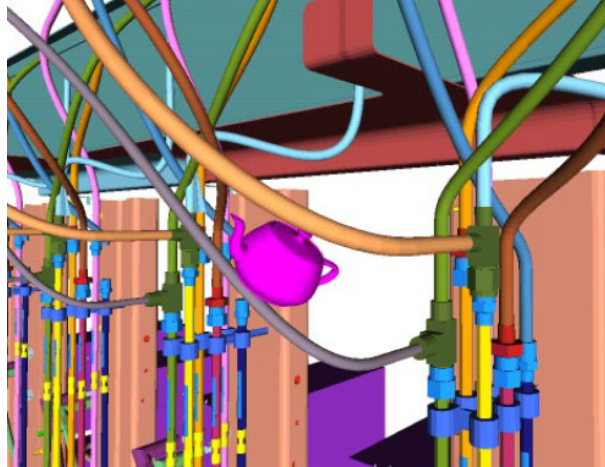


Figura 2.25. Objeto dinámico manipulado por dispositivo háptico en un escenario complejo, McNeely, et al., (1999) [132].

Adams (2001) [133] investiga los beneficios de la retroalimentación de fuerza para el entrenamiento virtual de una tarea real. Se utiliza un dispositivo háptico Excalibur con tres grados de libertad. La plataforma de simulación, llamada *Virtual Building Block* (VBB), tiene la finalidad de realizar ensambles de bloques tipo LEGO. Se realiza el entrenamiento a tres grupos de usuarios, el primero se entrenó utilizando retroalimentación háptica, el segundo sin retroalimentación háptica y el tercero sin entrenamiento. Después de la etapa de entrenamiento todos los sujetos construyeron el modelo real de LEGO con cinco repeticiones. El análisis revela que los operadores entrenados con retroalimentación háptica completaron el ensamble más rápido que aquellos que no recibieron entrenamiento. Aunque el tiempo promedio de los operadores entrenados con retroalimentación háptica es mejor que los entrenados sin retroalimentación, la alta varianza de los datos no permite identificar una diferencia significativa.

Por otro lado, Unger (2001) [134] describe un arreglo para la comparación del desempeño del usuario durante el ensamble real y virtual de un ensamble perno-agujero. Durante el ensamble virtual los objetos son renderizados háptica y gráficamente, mientras que en la tarea real un perno físico está pegado al efector final de un dispositivo háptico magnético. El perno es insertado al agujero de una placa que cuenta con sensores de torque y fuerza, las posiciones y orientaciones son medidas mediante el dispositivo háptico. Los resultados indican un mayor tiempo de ensamble, mayores fuerzas y más fallos en la tarea virtual, asimismo se observa que las estrategias del usuario para realizar la tarea son similares tanto en el ensamble real como en el ensamble virtual.

Huang, *et al.*, (2002) [135] estudian el efecto de la retroalimentación háptica en el desempeño del usuario cuando realiza una tarea dinámica. Los resultados muestran que una retroalimentación háptica de magnitud notable resulta en un mejor desempeño. Por su parte Ruffaldi, *et al.*, (2006) [136] sugieren que para comparar los resultados obtenidos en aplicaciones hápticas contra resultados de procesos reales se requiere aplicar las mismas entradas a los objetos virtuales que a los objetos reales. Además también es necesario medir las fuerzas resultantes en el proceso real, lo que requiere de dispositivos que no son de uso común. En el trabajo realizado por Lim *et al.*, (2007) [16] se investiga el impacto del renderizado háptico en la eficiencia del usuario en tareas de ensamble. El ensamble perno-agujero fue estudiado para determinar el efecto de la retroalimentación háptica en el tiempo de ensamble. Se observó que las características de ensamble pequeñas y visualmente insignificantes, como redondeos en los bordes de las piezas, afectan el tiempo de ensamble cuando se provee de retroalimentación háptica. Asimismo, el uso de detección de colisiones y el uso de estereovisión puede afectar los tiempos de ensamble en ambientes virtuales. La diferencia en el tiempo de ensamble debido a diferentes características es equivalente a los valores encontrados en el ensamble real.

En el trabajo realizado por Sagardía, *et al.*, (2009) [137] se presenta un conjunto de escenarios e indicadores para comparar algoritmos de renderizado háptico utilizando trayectorias de ensamble generadas automáticamente por computadora. Cada escenario consiste de un objeto dinámico y un objeto estático. Se analizan las fuerzas que son generadas por los algoritmos cuando se mueve el objeto dinámico sobre la trayectoria especificada. Las trayectorias son relacionadas a objetos delgados, los cuales son un problema en el renderizado háptico. En cada prueba se analiza la posición relativa del objeto dinámico respecto al estático, las fuerzas de colisión, los torques, la penetración y el tiempo de cómputo. Se utiliza el algoritmo VPS para la detección de colisiones.

Aleotti, *et al.*, (2010) [138] presentan una aplicación de realidad aumentada visual-háptica. El sistema está enfocado en tareas de manipulación y combina un dispositivo háptico de tres grados de libertad y animación basada en física. Los objetos virtuales son manipulados por el usuario utilizando el dispositivo háptico y coexisten con el mundo real sobre una representación virtual del espacio de trabajo. El sistema soporta la interacción con cuerpos rígidos y deformables. Los resultados muestran que la retroalimentación de fuerza reduce el tiempo de terminación de la tarea de manipulación.

Un análisis comparativo entre la retroalimentación visual y la retroalimentación de fuerza para la detección de colisiones en ensamble virtuales, es realizado por Sagardía (2012) [139]. Los resultados revelan una clara y significativa superioridad de la retroalimentación de fuerza sobre la retroalimentación visual. De acuerdo a Xia, *et al.*, [13] (2013), la retroalimentación háptica mejora el desempeño del ensamble virtual al reducir el tiempo de terminación de la tarea, incrementar la precisión al posicionar los objetos virtuales, y proveer movimientos de manipulación más estables sobre trayectorias tridimensionales.

En diversos estudios realizados para evaluar el desempeño del usuario en aplicaciones de ensambles virtuales se ha observado que el tiempo para completar el ensamble es mayor en el ambiente virtual comparado contra el ensamble real. Gran cantidad de factores contribuyen a esta diferencia, entre los cuales se destacan: el tipo de interfaz utilizada para la manipulación de objetos virtuales (guante, mouse, háptico, etc.), la forma y el peso de los objetos virtuales, la manipulación de la cámara virtual e incluso el tipo de proyección utilizada (monitores 2D, monitores con estereovisión, sistemas inmersivos, etc.).

## 1.6 Percepción del peso

Diversos autores como Huang [135], Giachristis [140], Tzafestas[141], Dominjon[142] y Hara[143] han concluido que la percepción del peso de los objetos virtuales influye en el desempeño del usuario en tareas virtuales, este desempeño es generalmente medido en términos del tiempo de terminación de la tarea. Wang, *et al.*, (2001) [98] clasifican los objetos virtuales en tres categorías determinadas por su peso: partes que pueden ser levantadas con una mano, partes que pueden ser levantadas con dos manos y partes que son levantadas con una herramienta. Se concluye que la simulación dinámica en un ambiente virtual intensifica el sentimiento realista del espacio virtual pero no es apta para partes demasiado grandes o pesadas y además no contribuye significativamente a la tarea de ensamble. Por otro lado, los estudios realizados por Tzafestas (2003) [141] muestran que la percepción humana del peso, cuando se manipula un objeto en el ambiente virtual, es similar a la percepción del peso cuando se manipula un objeto real. Se concluye también que las imperfecciones y limitaciones del dispositivo háptico pueden llevar a un pequeño decremento en el desempeño del usuario.

A la relación que existe entre la magnitud del movimiento que el usuario realiza con la mano y el movimiento resultante del objeto observado en el ambiente virtual se le conoce como relación control/display (C/D). Una relación C/D igual a uno indica que si el usuario mueve la mano un centímetro el objeto virtual también se moverá un centímetro. El efecto de la relación C/D en la percepción del peso de los objetos virtuales es estudiado por Dominjon, *et al.*, (2005) [142]. Los resultados muestran que cuando se utilizan relaciones C/D más pequeñas que 1 (esto es amplificación de los movimientos del usuario en el ambiente virtual) los participantes percibieron el objeto manipulado más ligero que su peso asignado; incluso en algunos casos fue incluso posible invertir la sensación del peso para hacer que los objetos pesados se sintieran más ligeros que un objeto ligero. Esto sugiere que la masa puede ser añadida a la lista de propiedades hápticas que pueden ser simuladas mediante retroalimentación pseudo-háptica (percepción del peso sin necesidad de un dispositivo háptico). Por su parte Hara (2008) [143] analiza la percepción del usuario cuando el peso de un objeto virtual cambia de manera repentina mientras está siendo manipulado por el dispositivo háptico. Los resultados indican que los usuarios perciben un cambio en el peso del objeto virtual solo cuando la diferencia entre el peso inicial y el peso actual es significativa. Se concluye que cuando no se percibe un cambio en el peso, el usuario inconscientemente ajusta sus músculos a ese cambio.

Los beneficios de las interacciones con retroalimentación de fuerza para realizar tareas de ensamble en ambientes virtuales son estudiados por Vo, *et al.*, (2009) [144]. En una tarea de discriminación del peso, se pidió al usuario identificar el objeto virtual más pesado de un par de objetos virtuales mostrados. Los resultados obtenidos mostraron que los usuarios compararon los modelos en menos tiempo utilizando la retroalimentación háptica que utilizando solamente la percepción visual. También se observó que la identificación del peso depende de la mano que se utiliza para manipular el objeto. Por otro lado, la percepción del peso cuando se levantan cajas virtuales con una o dos manos es evaluada por Giachristis, *et al.*, (2009) [140], donde se propone que la percepción adecuada y precisa del peso simulado deberá de conducir al usuario a ejecutar la tarea con alta precisión. Los resultados indican que en el levantamiento a dos manos los objetos virtuales tienden a sentirse más ligeros que los mismos objetos levantados con una sola mano. Los usuarios parecen ser cinco veces menos sensibles a la discriminación o diferenciación de pesos en el ambiente virtual que en el ambiente real.

### 1.7 Discusión

Los trabajos de investigación reportados en la literatura revelan que el ensamble interactivo virtual puede ser utilizado como una herramienta para acelerar el proceso de diseño de un producto. Se ha demostrado que la retroalimentación de fuerza produce una simulación más realista, permitiendo realizar las tareas con mayor precisión y en un menor tiempo. Por lo tanto, la retroalimentación de fuerza es ideal para procesos de ensamble virtual, el cuál debe ser un proceso interactivo que soporte la intuición humana para poder obtener resultados exitosos.

Varias técnicas y sistemas de ensamble virtual han sido desarrolladas para simular, analizar y evaluar el ensamble de componentes. Sin embargo, todavía hay mucho desconocimiento del comportamiento de estos sistemas debido a la complejidad que estos representan; existen diversos factores y componentes de un sistema de ensamble virtual que requieren de mayor estudio. Uno de los principales problemas al desarrollar plataformas de ensamble virtual es la sincronización entre los principales módulos del sistema -el simulador gráfico, el simulador de física y el simulador háptico-, pues cada uno trabaja a tasas de actualización diferente: la simulación háptica requiere de altas tasas de actualización de alrededor de 1kHz, mientras que la simulación gráfica y la simulación física se actualizan a una velocidad menor de 100hz.

La detección de colisiones representa una de las mayores dificultades dentro de las plataformas de ensamble virtual, pues es un proceso lento y con limitaciones en cuanto a la precisión geométrica de los modelos. Se requieren nuevos métodos y algoritmos para mejorar la resolución de contactos entre objetos virtuales en términos de rapidez y precisión. La detección de colisiones está directamente ligada a la forma en que se representan los objetos virtuales dentro de los simuladores, especialmente en el simulador de física, o en el método de detección de colisiones, por lo tanto se deben de proponer y utilizar nuevos métodos que permitan utilizar formas más complejas sin sacrificar el rendimiento del sistema.

Se han identificado los dos métodos principales para el desarrollo de plataformas hápticas de ensamble virtual: el modelado basado en restricciones y el modelado basado en física, cada uno con sus ventajas y desventajas. Sin embargo, utilizar un modelo híbrido que mezcle las dos metodologías puede resultar en un sistema de ensambles virtuales más eficiente y con alto nivel de realismo. En cuanto a los simuladores físicos, PhysX, Bullet y Havok, son los que muestran un mejor desempeño en las evaluaciones realizadas. Por otro lado, se ha observado que el dispositivo háptico Phantom Omni presenta una mejor relación costo beneficio y se adapta a los requerimientos de las tareas de ensamble virtual. Adicionalmente existen diversos factores como la percepción del peso, dispositivos, métodos de manipulación de los objetos virtuales, visualización, ente otros, que deben de ser estudiados para establecer su influencia en el proceso de ensamble virtual.

### 2.1 Introducción

La revisión bibliográfica presentada en el capítulo anterior reveló que existen ciertas características de los sistemas de ensamble virtual que aun necesitan ser estudiadas con el fin de desarrollar aplicaciones que puedan ser utilizadas en actividades reales como el diseño y la manufactura. Con este propósito se ha desarrollado un nuevo sistema llamado HAMS (*Haptic Assembly and Manufacturing System*) el cual se pretende utilizar como una herramienta para entrenamiento, análisis de diseño y planeación del ensambles de ensambles virtuales apoyado por dispositivos háptico. Además, la configuración y flexibilidad de HAMS permitirán su uso como una plataforma para evaluar el efecto de varios parámetros y componentes en el desempeño de la simulación de las tareas de ensamble. En este capítulo se describe el desarrollo del sistema HAMS así como sus principales características y funcionalidades.

### 2.2 Descripción del sistema

El sistema HAMS, desarrollado y propuesto en el presente trabajo, comprende tres módulos principales: el módulo gráfico, el módulo físico y el módulo háptico, Figura 2.1. Las características clave de cada módulo son:

1. Módulo gráfico: es el encargado del renderizado gráfico, el cual incluye la creación y representación gráfica de la escena virtual y de los modelos tridimensionales, la visualización de trayectorias de ensamble, la visualización de información en forma de texto, así como la creación y visualización de botones virtuales llamados *widgets* para modificar parámetros de la simulación. Para este módulo, el sistema HAMS emplea las librerías “*Visualization Toolkit (VTK) v5.10*” [145], cuyo principal objetivo es la visualización de volúmenes y elementos 3D. VTK Permite la creación y visualización de modelos primitivos como esferas, cilindros, cubos y conos, además de la visualización de modelos de forma arbitraria compleja representados por medio de mallas triangulares. VTK es de código abierto, lo que permite modificar las funciones contenidas en estas librerías para adaptarlas a las necesidades de programación.
2. Módulo físico: es el responsable del comportamiento físico de los objetos virtuales, dándoles un movimiento dinámico realista, además de detección y respuesta a colisiones. En la plataforma de ensambles desarrollada se utilizan tres simuladores de física – Bullet physics v2.81 [5], PhysX SDK v2.8 [6], y PhysX SDK v3.1 [6] – los cuales de acuerdo con la revisión bibliográfica son los que presentan un mejor desempeño en términos generales. Estos simuladores emplean la solución de ecuaciones de la mecánica Newtoniana para resolver el movimiento de cuerpos rígidos, cuerpos flexibles y juntas cinemáticas. El sistema HAMS permite seleccionar cualquiera



de los tres simuladores de física durante la operación del sistema, esto con el fin de utilizar el simulador que mejor se adapte a las condiciones de ensamblaje particulares.

3. Módulo háptico: es el encargado de calcular y proveer la fuerza de retroalimentación que habilita el sentido del tacto y cinestesia para que el usuario pueda reconocer y manipular los objetos virtuales. Se utilizan las librerías HLAPI de OpenHaptics v3.0 [146], las cuales son las librerías estándar para la programación de los dispositivos hápticos Phantom (ahora conocidos como Geomagic Touch). El sentido de tacto se crea por medio de detección de colisiones entre el cursor háptico y el objeto virtual. El sistema HAMS también permite la manipulación de objetos utilizando un dispositivo háptico Phantom Omni en cada mano.

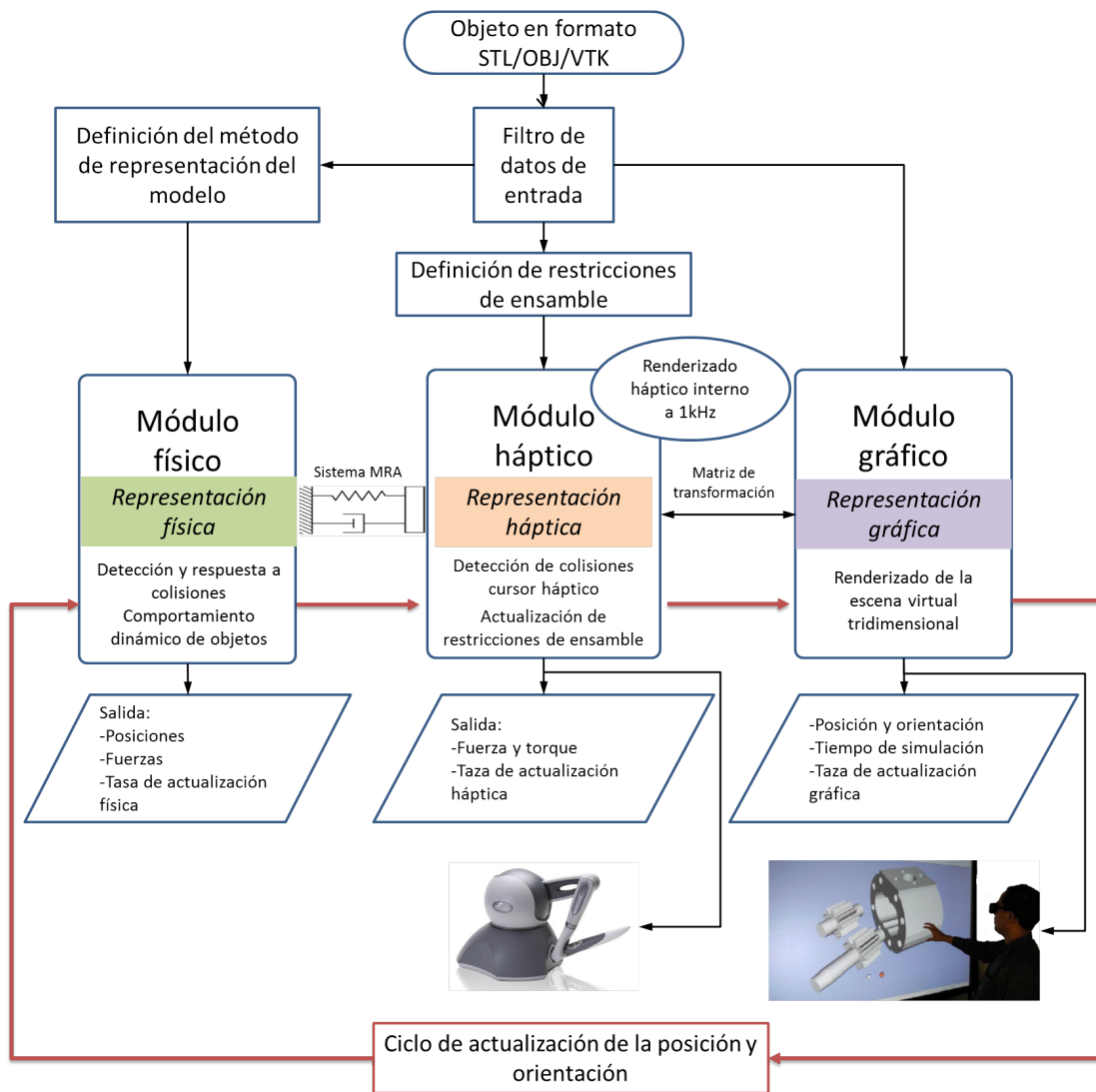


Figura 2.1. Interfaz del sistema de ensamblajes virtuales HAMS.

Estos tres módulos están integrados en la plataforma MFC (*Microsoft Foundation Classes*) de Visual Studio 2010, la cual permite crear interfaces de usuario gráficas (GUIs) en ambiente Windows basadas en la programación orientada a objetos. MFC permite crear una sistema virtual flexible donde existen controles para cargar nuevos objetos, encender o apagar el dispositivo háptico, activar opciones de análisis, orientar vistas de la cámara y modificar parámetros de simulación entre otras funciones. En la Figura 2.2 se observa la interfaz gráfica de usuario mientras que en la Figura 2.3 se muestra la interfaz háptica del sistema HAMS.

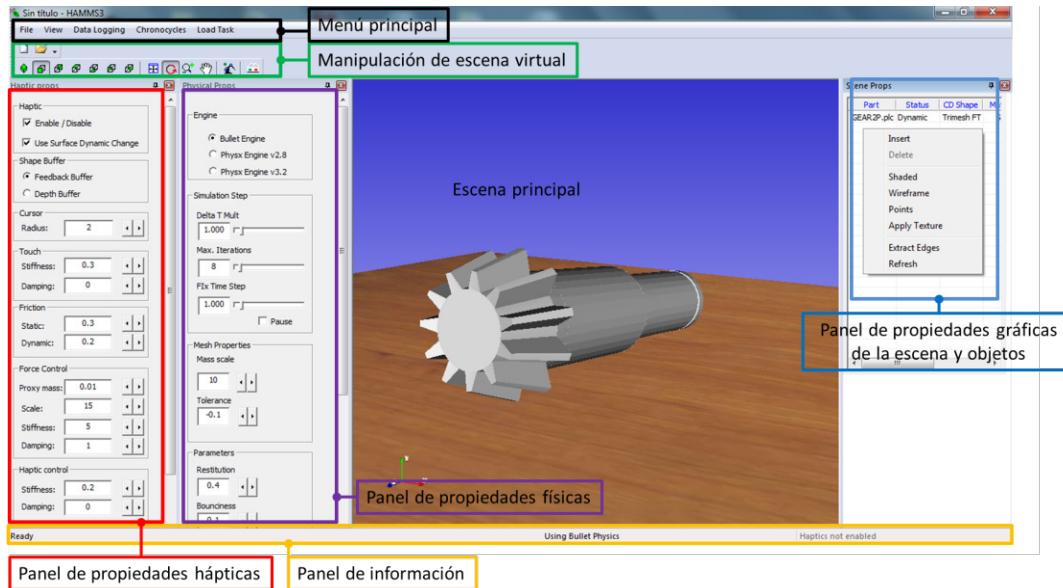


Figura 2.2. Interfaz gráfica de usuario del sistema HAMS.

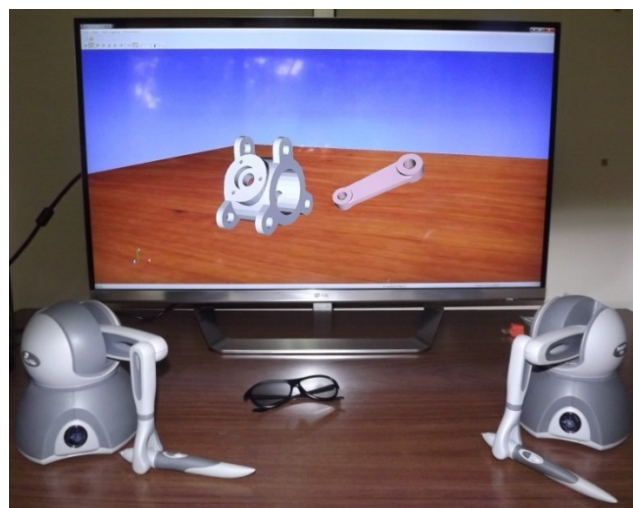


Figura 2.3. Interfaz háptica del sistema de ensamblajes virtuales HAMS.

### 2.3 Integración

Para que el usuario perciba que los objetos virtuales tienen un comportamiento dinámico fluido, la posición de dichos objetos debe ser actualizada cada cierto tiempo, generalmente un intervalo de tiempo muy corto. Por tanto, los módulos gráfico, físico y háptico deben ser sincronizados y actualizados en cada intervalo de tiempo para cambiar de manera dinámica la posición gráfica, física y háptica de cada uno de los objetos virtuales. Cada uno de los módulos debe cumplir con una tasa o rapidez de actualización basada en los requerimientos de la percepción humana.

El moduló háptico requiere de ser actualizado con una alta rapidez, alrededor de 1kHz o 1000 veces por segundo, ya que la percepción humana para reconocer objetos por medio del sentido del tacto trabaja a esta velocidad. Tasas de actualización háptica más bajas causarían la sensación de que el objeto no tiene suficiente rigidez, o incluso vibración al momento de tocar los objetos virtuales.

La simulación gráfica requiere de una actualización entre 50 y 60 Hz, que es la velocidad a la cual el ojo humano percibe los movimientos como continuos. En aplicaciones que requieren el uso de estereovisión para producir una visualización tridimensional la tasa de actualización gráfica puede ser hasta de 120 Hz.

Por último, la simulación física es la responsable de actualizar las posiciones de los objetos al tomar en cuenta las fuerzas que actúan sobre ellos. Si la posición de los objetos no es actualizada de manera continua su movimiento se percibirá como lento, con un efecto de “gravedad lunar” o “baja gravedad”, incluso se observará un movimiento no continuo con instantes en que los objetos se detienen, lo cual no concuerda con el movimiento real. Debido a esto, la rapidez de la actualización física debe ser cercana a la rapidez de actualización gráfica. Algunos investigadores han encontrado que la actualización física debe ser mayor de 25Hz, pues actualizaciones más bajas resultan en un comportamiento no realista de los objetos con pausas entre cada cambio de posición. Así, el ciclo completo de la simulación del sistema de ensamblajes virtuales debe ser actualizado tomando en cuenta las tasas de actualización antes mencionadas, Figura 2.4.

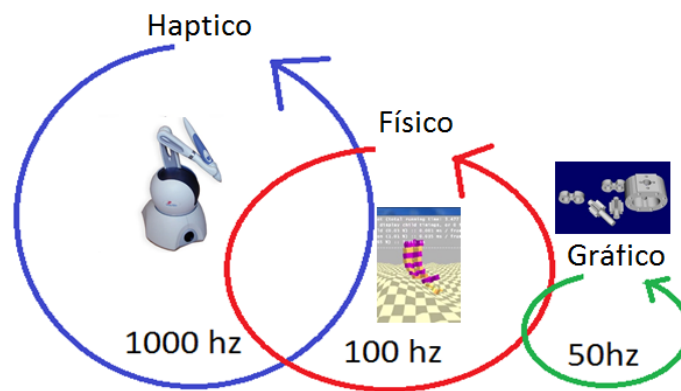


Figura 2.4. Tasas de actualización de cada módulo de simulación.

Con el fin de lograr la actualización de posiciones se crea un ciclo de actualización total, en el cual cada módulo es actualizado en un orden definido continuamente durante la ejecución del programa. Debido a que la posición de los objetos es calculada por el módulo físico este debe ser el primero en actualizarse. Así, los módulos siguientes utilizan los resultados del módulo anterior para su actualización. El ciclo de actualización total debe cumplir con las tasas de actualización requeridas para cada módulo de simulación con el fin de calcular de manera correcta la posición de los objetos y mostrarla a través de la pantalla y del dispositivo háptico. En la Figura 2.5 se muestra el diagrama de flujo del ciclo de actualización total.

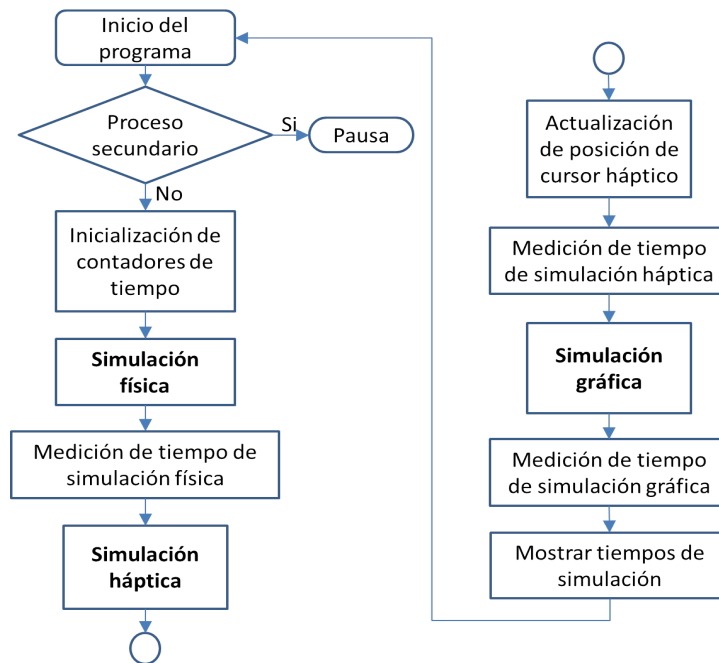


Figura 2.5. Ciclo de actualización total del sistema HAMS.

En el ciclo de actualización de la Figura 2.5 se muestra la actualización del módulo háptico en serie con los otros dos módulos. Lo cual indica que la rapidez de simulación total está determinada por la rapidez de actualización del módulo más lento, generalmente el módulo físico. Para lograr un funcionamiento correcto todos los módulos se deberían actualizar a tasas de alrededor de 1kHz (tasa de actualización háptica), lo cual es muy difícil de cumplir por los módulos gráfico y físico. Por lo tanto, se debe aclarar que este ciclo de simulación háptica no necesariamente se actualiza a esta velocidad; la tasa de actualización de 1 kHz es con fines de lograr que el usuario perciba los objetos virtuales de manera sólida y uniforme al momento de utilizar el tacto, es decir, cuando toca los objetos para reconocerlos, pero no manipularlos. Sin embargo, cuando el usuario manipula un objeto se utiliza la percepción kinestésica, la cual requiere de una actualización de alrededor de 10Hz, es decir, para que el usuario perciba objetos en movimiento de manera correcta y fluida se debe de actualizar su posición por lo menos 10 veces por segundo. Por lo tanto, el ciclo mostrado cumple con las tasas de

actualización dinámica para cada módulo. Para cumplir con el requisito de actualización de 1kHz de la percepción del tacto, HLAPI utiliza un procesamiento paralelo para llevar a cabo la detección de colisiones entre el cursor háptico y los objetos virtuales, y así poder ofrecer una sensación realista al tacto. Este procesamiento paralelo es independiente del ciclo de simulación principal.

## 2.4 Creación de modelos

Una vez que se han creado e inicializado los ciclos de simulación es posible cargar modelos CAD en el sistema para simular su proceso de ensamble. En HAMS los modelos CAD pueden ser cargados utilizando archivos en formato STL, OBJ o VTK. Para cargar un modelo se utiliza el comando “Nuevo” del menú de herramientas del GUI HAMS, el cual abre una ventana del explorador de Windows, Figura 2.6, en la cual se puede ver y seleccionar el archivo deseado.

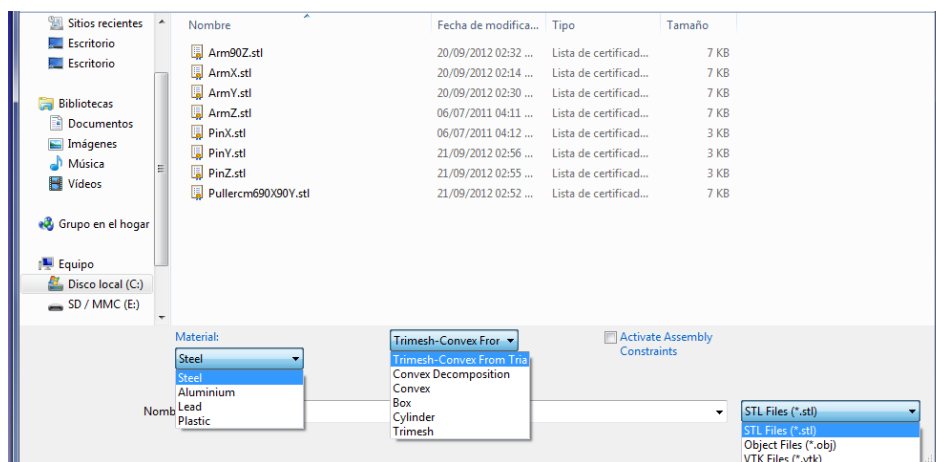


Figura 2.6. Ventana para cargar un nuevo modelo en HAMS.

La función principal de esta ventana es poder encontrar y abrir los archivos que contienen la descripción del modelo a cargar. Sin embargo, otras variables o parámetros referentes al modelo, como el material y tipo de representación, también pueden ser asignadas al momento de cargar el objeto.

Cuando se carga un nuevo modelo en el sistema HAMS todos sus datos son asignados a la variable *vtkActor*, la cual adquiere toda la información necesaria del modelo, como su forma, el método de representación, las restricciones de ensamble y la posición y orientación iniciales. Además a cada modelo cargado se le asigna un índice de colisión (número de actores u objetos en la escena), una densidad y un identificador háptico. Finalmente, cada objeto se agrega a una lista de ensamble (lista que contiene todos los objetos dinámicos en la escena), se ajusta la cámara y los límites del espacio de trabajo para que el actor quede al centro de la escena virtual.

Al cargar un modelo se le puede asignar el tipo de material - acero, plomo, madera o plástico - cada uno con densidad diferente, lo cual afecta la percepción del peso y el comportamiento dinámico del objeto virtual. En la sección de manipulación háptica se describe el mecanismo por medio del cual el usuario puede sentir el peso de los objetos virtuales.

Los archivos en formato OBJ o STL contienen la información del modelo tridimensional en forma de vértices e índices que forman triángulos, los cuales describen la superficie del modelo CAD, esta representación es conocida como malla triangular. Al cargar un objeto nuevo el programa extrae la información contenida en el archivo con el fin de aplicar los filtros adecuados y poder reconstruir la malla triangular de manera adecuada en el formato requerido por cada simulador. La malla triangular leída es utilizada para crear tres representaciones del objeto, una representación por cada módulo de simulación dentro del sistema HAMS, Figura 2.7. Cada representación es creada bajo los métodos propios de cada simulador. El proceso general para crear cada una de las representaciones en el sistema HAMS se describe en la Figura 2.8.

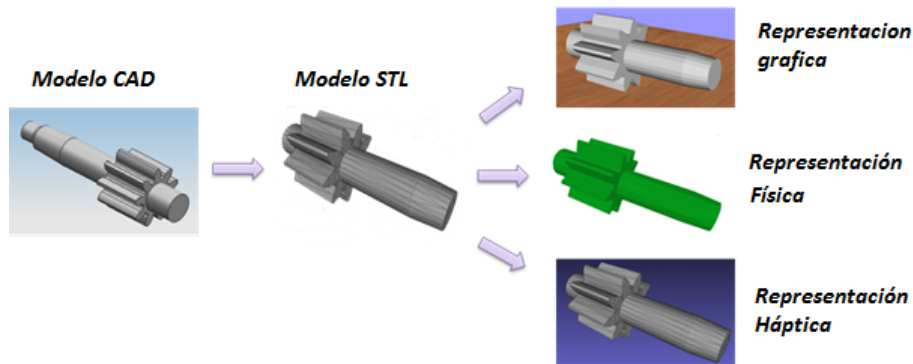


Figura 2.7. Representación del objeto virtual en cada módulo de simulación del sistema de ensamblés virtuales.

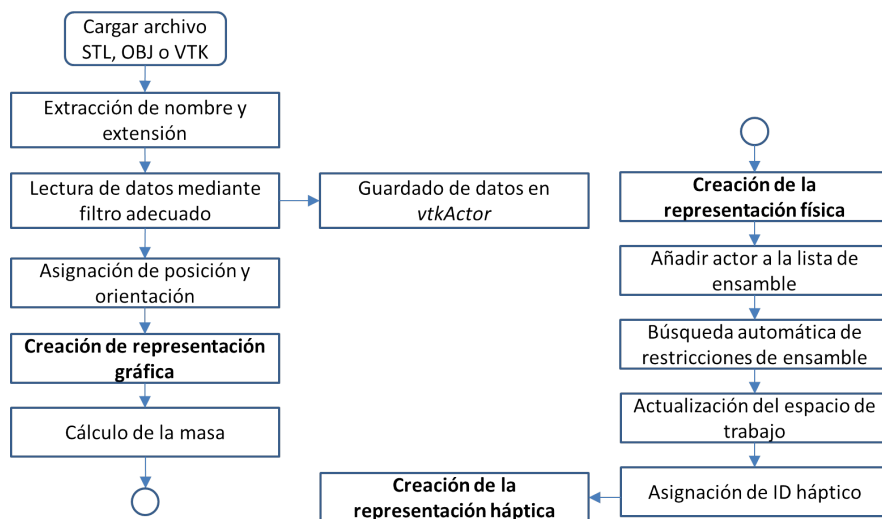


Figura 2.8. Proceso para la creación de objetos virtuales en HAMS.

### 2.4.1 Representación gráfica

Esta representación es utilizada para mostrar al usuario el modelo como un objeto virtual en la pantalla. Se crea por medio de funciones definidas en las librerías VTK, las cuales utilizan directamente la malla triangular del modelo CAD para crear esta representación y por lo tanto, su precisión respecto al modelo original depende de la resolución dada a la malla triangular al momento de generar el archivo STL. En la Figura 2.9 se puede observar el proceso específico para la creación del modelo gráfico.

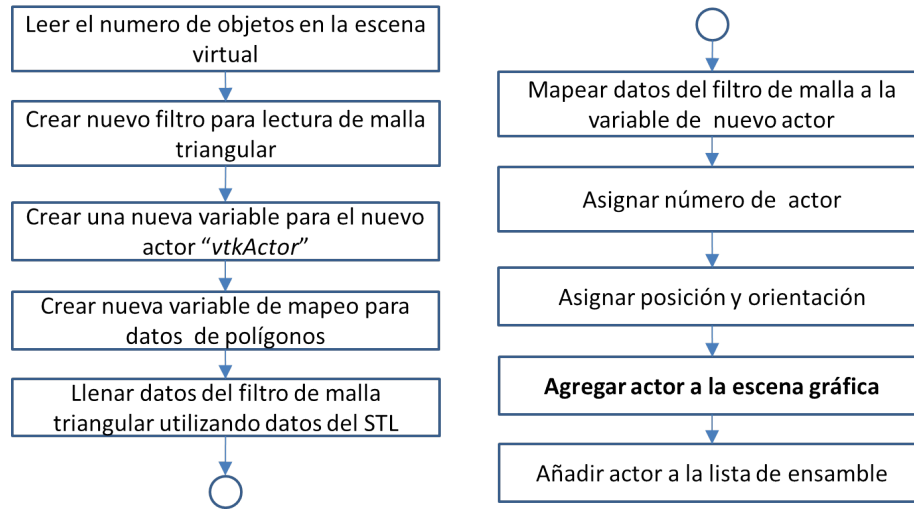


Figura 2.9. Proceso para la creación de la representación gráfica.

### 2.4.2 Representación física

El modelo físico habilita el comportamiento basado en la física real y la detección de colisiones entre los objetos virtuales. Esta representación se crea dentro de los simuladores de física, Bullet y PhysX, es invisible al usuario e independiente a la representación gráfica, por lo tanto puede ser completamente diferente; por ejemplo, si la representación gráfica es una malla triangular que representa un automóvil, la representación física de ese mismo modelo puede ser una esfera. Para el caso de ensamblajes virtuales, el modelo físico debe ser lo más parecido posible al modelo original con el fin de que los resultados de la simulación del ensamble sean comparables contra el ensamble real. Existen diversos métodos y algoritmos para crear la representación física, los cuales son presentados en el Capítulo III del presente documento. Asimismo, en los capítulos IV y V se presenta el análisis y la evaluación de la influencia de los diferentes métodos de representación física en el ensamble virtual. El proceso para crear la representación física en el sistema HAMS se muestra en la Figura 2.10.

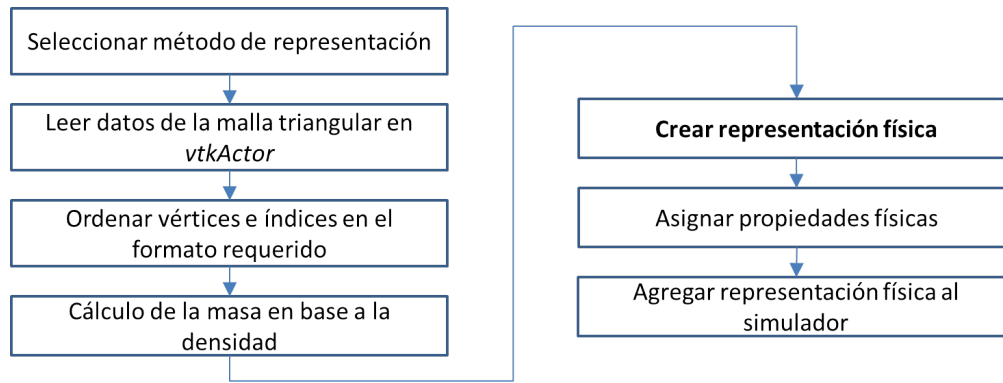


Figura 2.10. Proceso para la creación de la representación física.

### 2.4.3 Representación háptica

Esta representación permite que los objetos virtuales puedan ser tocados y manipulados utilizando el dispositivo háptico. El sentido del tacto es habilitado mediante la detección de colisiones entre la representación háptica y el cursor háptico. La representación háptica es creada utilizando funciones de las librerías HLAPI, las cuales capturan la geometría del modelo gráfico cuando es creado mediante comandos de VTK y OpenGL. Así, el modelo háptico es creado en base al modelo gráfico y por lo tanto ambos serán iguales. De manera similar a la representación física, este modelo también es invisible al usuario. Dentro de HLAPI existen dos métodos para capturar la geometría gráfica desde los comandos de OpenGL:

1. Renderizado por búfer de profundidad (*depth buffer*). En el búfer de profundidad gráfico se almacenan y administran todos los datos que contienen las coordenadas de profundidad del modelo tridimensional. HLAPI lee estos datos en cada ciclo de simulación y los utiliza para realizar el renderizado háptico, por lo tanto, cualquier cambio en la representación gráfica afecta a la representación háptica. Este método es recomendado cuando se cambia de manera dinámica la forma de los objetos virtuales. Sin embargo, tiene la desventaja de que algunas características del modelo que no son visibles en pantalla también son omitidas en el renderizado háptico, es decir solo se toca lo que se ve en pantalla. Este método es recomendado cuando el modelo es muy complejo y está integrado por una gran cantidad de polígonos.
2. Renderizado por búfer de retroalimentación (*feedback buffer*). Este método lee todos los puntos, líneas y polígonos que forman el modelo gráfico y los guarda en un búfer de retroalimentación, lo que requiere de reservar espacio de memoria cada vez que se carga un modelo. Estos datos son utilizados para realizar el renderizado háptico. Este método utiliza todos los datos que componen al modelo, por lo tanto, brinda una representación háptica con mejor precisión geométrica. Sin embargo, debido a la gran cantidad de información que maneja no es recomendable para modelos de gran complejidad o con un alto contenido de elementos en su malla triangular, tampoco es recomendable para simulaciones que involucren cambios dinámicos en la forma del modelo. Este método permite que todo el modelo sea renderizado hápticamente, sin



importar que existan secciones del modelo que no sean visibles en la pantalla. Se recomienda este método cuando los modelos son sencillos y con relativamente pocos triángulos (hasta 65,536 vértices).

El sistema HAMS permite al usuario seleccionar durante su ejecución el método de captura de la geometría, con el fin de ajustar la simulación háptica a las mejores condiciones de acuerdo a los modelos cargados. Adicionalmente se requiere generar un número identificador háptico cada que se carga un nuevo modelo en el sistema de ensamblés. Este número relaciona el modelo gráfico con el modelo háptico y además sirve para detectar cambios en el movimiento de un objeto en cada paso de la simulación y de este modo renderizar las fuerzas de contacto de manera correcta. Una vez generado el identificador háptico se deben asignar valores a ciertas propiedades hápticas para cada objeto cargado. Entre las propiedades hápticas se encuentran la rigidez, la fricción estática, la fricción dinámica, el amortiguamiento y el tipo de superficie de contacto (exterior, interior o ambas). El proceso de creación de la representación háptica se muestra en la Figura 2.11.

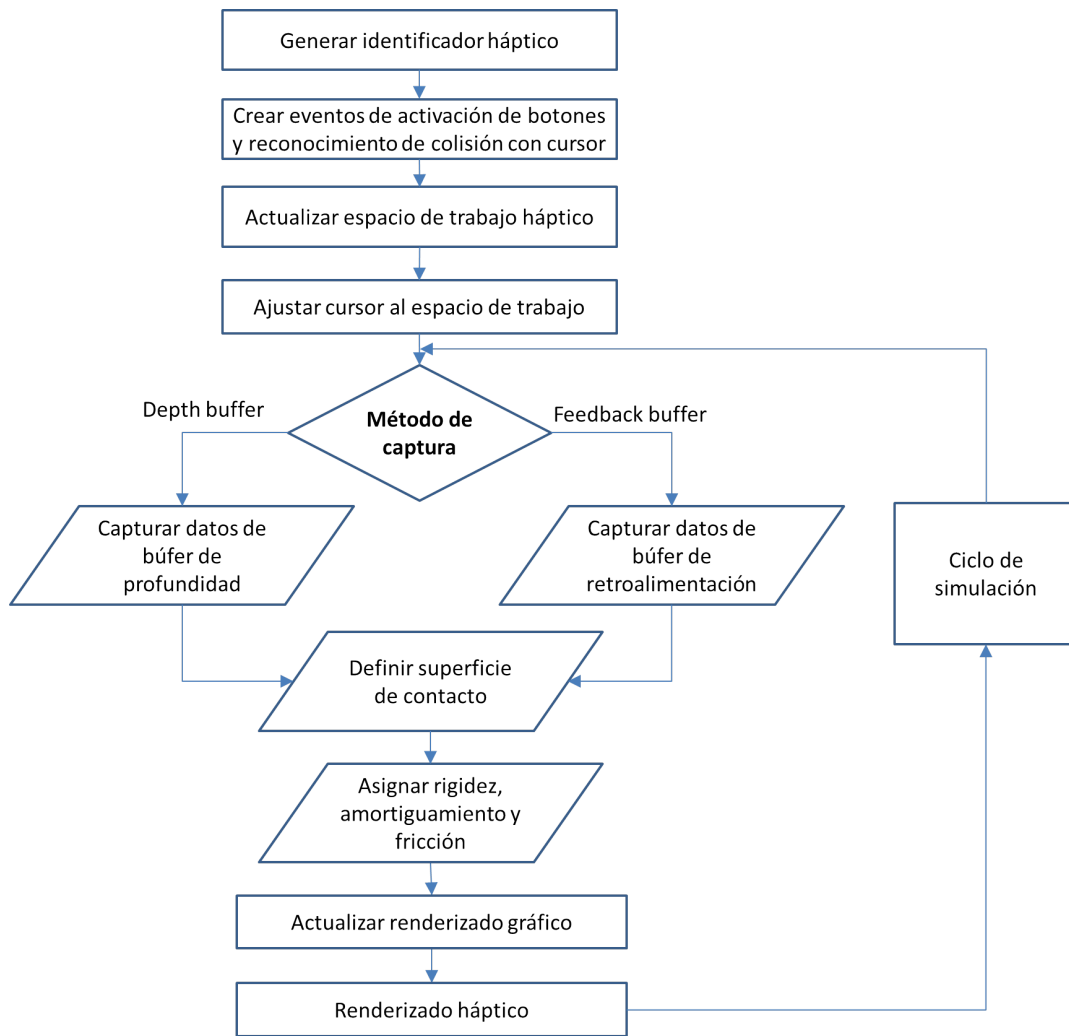


Figura 2.11. Proceso para la creación de la representación háptica.

Una vez que el ciclo de simulación total está activo y las representaciones de cada módulo han sido creadas es posible llevar a cabo el reconocimiento y la manipulación de los objetos virtuales por medio del dispositivo háptico, a esta fase se le llama la fase de interacción, en la cual se puede realizar el proceso de ensamble virtual.

## **2.5 Interacción háptica**

Una característica clave de los sistemas de ensamblajes virtuales es la interacción del usuario con los objetos virtuales, la cual debe permitir llevar a cabo el ensamble de una manera natural, intuitiva y eficiente. Para lograr este tipo de interacción los objetos virtuales deben tener un comportamiento dinámico, es decir, reaccionar a las fuerzas que se aplican sobre ellos produciendo su movimiento. Por otro lado, el usuario debe ser capaz de reconocer los objetos virtuales mediante el sentido del tacto, manipularlos, sentir su peso, su inercia y reconocer colisiones del objeto manipulado con otros objetos en el ambiente virtual. Para crear una simulación dinámica es necesario actualizar la simulación cada cierto tiempo, calculando las nuevas posiciones del objeto en base a las fuerzas que actúan sobre él en cada paso de la simulación. La simulación física debe ser la primera en actualizarse, pues es la que lleva a cabo el cálculo de las fuerzas sobre los objetos virtuales para determinar la posición y orientación que los objetos virtuales deben tener al final de cada ciclo de simulación. Estos resultados son entonces utilizados por los módulos de simulación gráfica y háptica para actualizar sus representaciones del objeto virtual y tener así consistencia en la posición y orientación de cada una de las representaciones - física, háptica y gráfica-.

### **2.5.1 Etapas de la interacción háptica**

Durante la ejecución del programa se pueden distinguir tres etapas principales de interacción; (i) la etapa de identificación del espacio, en la cual el usuario mueve el cursor háptico alrededor del ambiente virtual identificando sus movimientos y el espacio de trabajo, sin tocar o manipular los objetos; (ii) la etapa de reconocimiento, donde el usuario solo utiliza el dispositivo háptico para localizar y sentir los objetos existentes en la escena virtual; y (iii) la etapa de manipulación, donde el usuario selecciona un objeto y lo manipula utilizando el dispositivo háptico para realizar el ensamble.

En las etapas de identificación del espacio y reconocimiento, el modelo físico actualiza la ubicación del modelo gráfico a través de una matriz de transformación  $M_2$ . La posición y orientación del modelo háptico son actualizadas mediante la captura directa del modelo gráfico, este proceso se describe en la Figura 2.12. En la etapa de reconocimiento el usuario puede sentir y explorar la forma de los objetos virtuales para reconocerlos por medio de la fuerza de retroalimentación creada por la respuesta a colisiones entre el cursor háptico y la representación háptica.

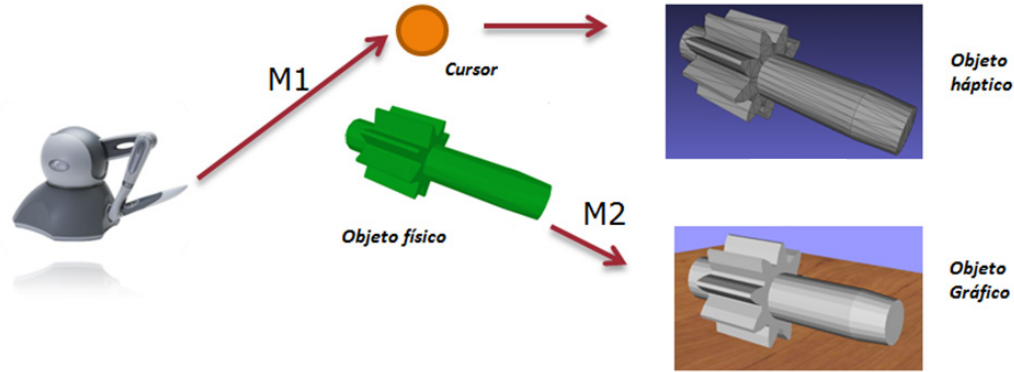


Figura 2.12. Fase de reconocimiento.

Cuando el usuario desea manipular un objeto virtual, debe de tocarlo y presionar el botón principal del dispositivo háptico, al hacer esto, el objeto virtual es seleccionado y cambia a color verde, el cursor desaparece e inicia la etapa de manipulación, la cual se lleva a cabo de la siguiente manera:

1. El cursor háptico es acoplado de manera directa a la representación háptica, es decir, al mover el dispositivo la representación háptica se moverá tal y como lo haría el cursor.
2. La representación háptica se acopla a la representación física por medio de un sistema masa-resorte-amortiguador (MRA) definido mediante la siguiente ecuación:

$$\vec{F} = m\ddot{\vec{x}} = k(\vec{r}_h - \vec{r}_p) - c\dot{\vec{x}} \quad (2.1)$$

donde:

- $\vec{F}$  Fuerza sobre el objeto.
- $m$  Masa del objeto.
- $\ddot{\vec{x}}$  Aceleración del objeto.
- $k$  Constante de rigidez del resorte.
- $\vec{r}_h$  Vector de posición de la representación háptica.
- $\vec{r}_p$  Vector de posición de la representación física.
- $c$  Constante de amortiguamiento.
- $\dot{\vec{x}}$  Velocidad del objeto.

3. Cuando se mueve la representación háptica, a través del dispositivo háptico, el sistema MRA crea una fuerza, la cual es aplicada al modelo físico para producir su movimiento.
4. Por último, la posición y orientación del modelo gráfico son actualizados de acuerdo al modelo físico utilizando una matriz de transformación M2. En la Figura 2.13 se observa el proceso de manipulación háptica de los objetos virtuales.

Mediante este método de acoplamiento es posible calcular la fuerza de retroalimentación háptica, la cual permite sentir el peso, inercia y colisiones del objeto virtual manipulado. Este modelo MRA es muy utilizado en aplicaciones hápticas y de ensamblaje virtual debido a que es sencillo, con baja complejidad algorítmica y permite obtener buenos resultados siempre y cuando los parámetros como masa, constante del resorte y amortiguamiento sean adecuados.

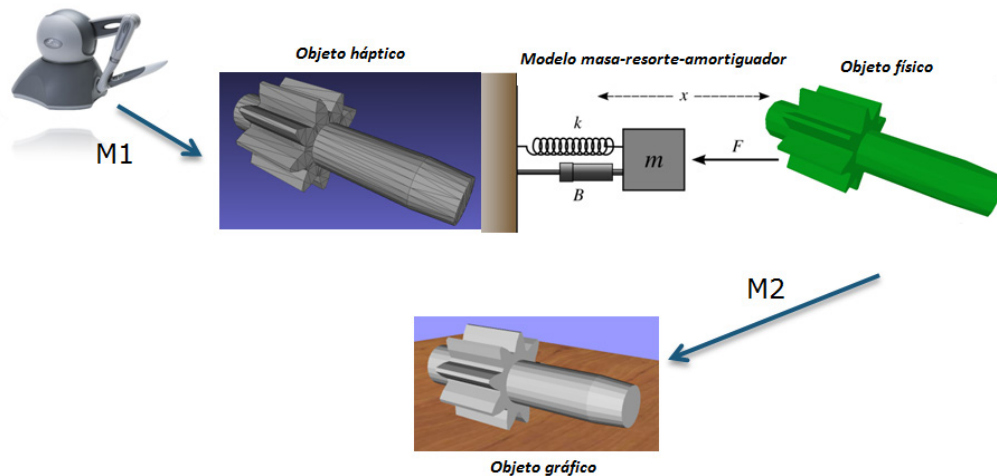


Figura 2.13. Fase de manipulación de objetos.

En el modelo MRA, ecuación (2.1) la constante de rigidez del resorte  $k$  depende de la masa de los objetos virtuales y de la precisión requerida en los movimientos. A mayor masa y mayor precisión, la rigidez deberá ser mayor. Sin embargo, valores muy grandes o no apropiados puede llegar a causar inestabilidad del sistema, haciendo que los objetos no puedan ser controlados. En el sistema HAMS se han asignado valores iniciales de rigidez y de amortiguamiento, los cuales producen una manipulación estable para la mayoría de los objetos virtuales. Estos valores pueden ser ajustados por el usuario según lo requiera de acuerdo al comportamiento del sistema y los modelos utilizados.

La estabilidad y precisión en la manipulación de los objetos virtuales dependen básicamente de dos parámetros: el cálculo de la fuerza por medio del sistema MRA, y el cálculo del movimiento resultante de aplicar dicha fuerza a la representación física. Este último paso es llevado a cabo por el simulador físico, por lo tanto, la correcta operación de este simulador es esencial dado que resultados no adecuados pueden ocasionar inestabilidad en el objeto manipulado y una respuesta a colisiones incorrecta, causando un desempeño pobre de la aplicación y dificultad para realizar el ensamble. El proceso para llevar a cabo la simulación física durante la etapa de manipulación se describe en la Figura 2.14.

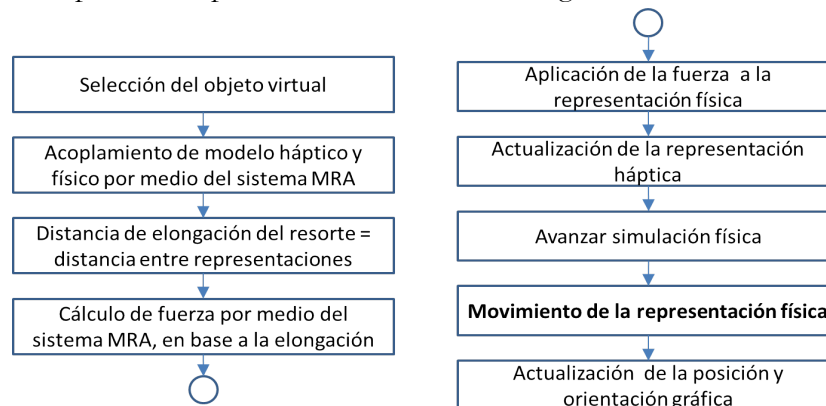


Figura 2.14. Proceso de manipulación de objetos virtuales.

### 2.5.2 Simulación háptica

Para poder sentir la forma de los objetos, su peso, inercia y colisiones a través del dispositivo háptico es necesario contar con un ciclo de simulación háptica. La función principal de este ciclo es leer la posición y orientación del dispositivo háptico en cada paso de la simulación y así calcular y renderizar las fuerzas que existan en el ambiente virtual. El cálculo de fuerzas debe realizarse de manera continua para que el usuario perciba uniformidad en la simulación.

Los datos de posición y orientación que se leen del dispositivo háptico son utilizados de manera diferente dependiendo de la etapa de simulación. En las primeras dos etapas, identificación del espacio y reconocimiento, estos datos son utilizados para dar la posición y orientación al cursor háptico. Al haber una colisión entre el cursor y la superficie de la representación háptica se crea una fuerza de repulsión que impide que el cursor traspase la superficie, esta fuerza es enviada al usuario por medio del dispositivo háptico para habilitar el sentido del tacto. En la fase de manipulación, la posición y orientación del dispositivo háptico se utilizan para controlar el modelo seleccionado, así al mover el dispositivo háptico se moverá la representación háptica y, por medio del modelo masa resorte amortiguador, se moverá la representación física la cual actualiza la representación gráfica de manera inmediata. En la Figura 2.15 se muestra el ciclo de renderizado háptico.

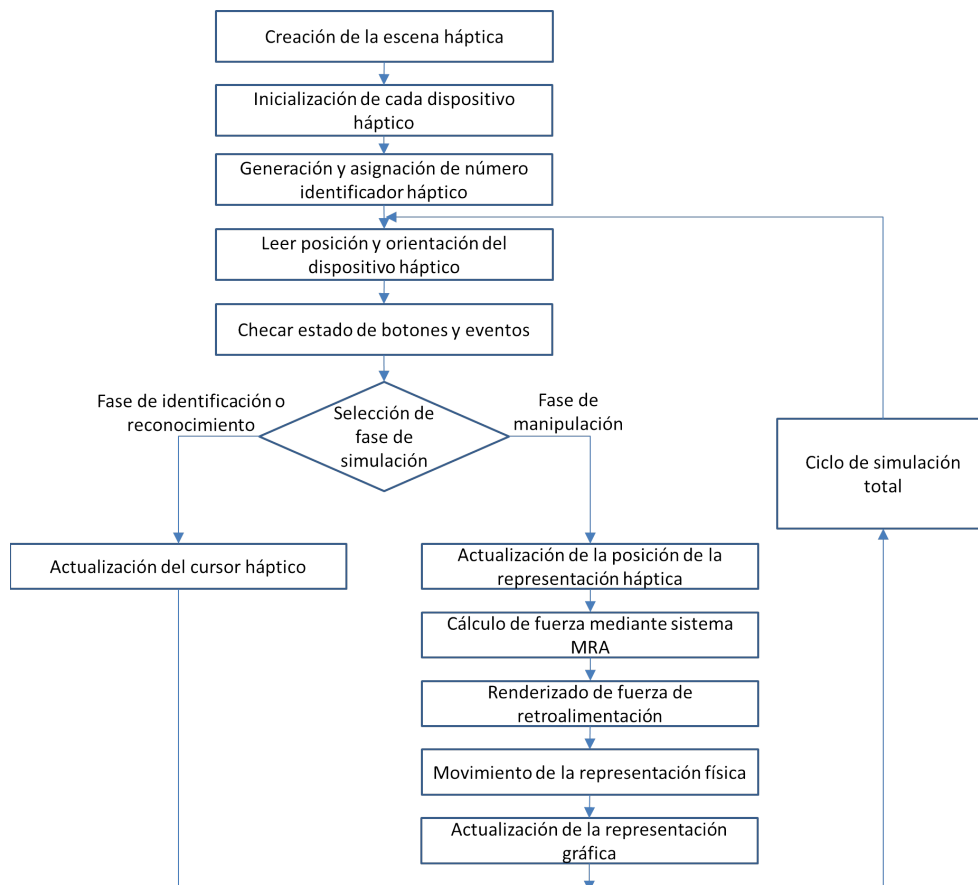


Figura 2.15. Ciclo de renderizado háptico.

### 2.5.3 Transformación de coordenadas

Los sistemas coordenados de cada módulo de simulación -háptico, físico y gráfico- tienen diferentes escalas y diferente forma de especificación y manejo de los datos. Por lo tanto, para lograr consistencia en la posición, orientación y escala de cada una de las representaciones de los objetos virtuales, es necesario realizar una transformación de coordenadas cada vez que se requiera intercambiar datos entre los módulos. Con este propósito se han definido dos transformaciones básicas:

*Transformación de coordenadas del espacio gráfico a coordenadas del espacio físico y háptico.* Cuando se carga un modelo en el sistema de ensamblajes se especifica su posición por medio de coordenadas del espacio gráfico, por lo tanto, es necesario aplicar una transformación de coordenadas para que las representaciones física y háptica tengan una posición y orientación consistentes con el ambiente gráfico. Además, esta transformación puede ser utilizada para controlar los objetos virtuales mediante el puntero del mouse de la computadora. El sistema HAMS incluye la posibilidad de controlar directamente los objetos virtuales mediante la representación gráfica y el mouse. El proceso de transformación de coordenadas gráficas se muestra en la Figura 2.16.

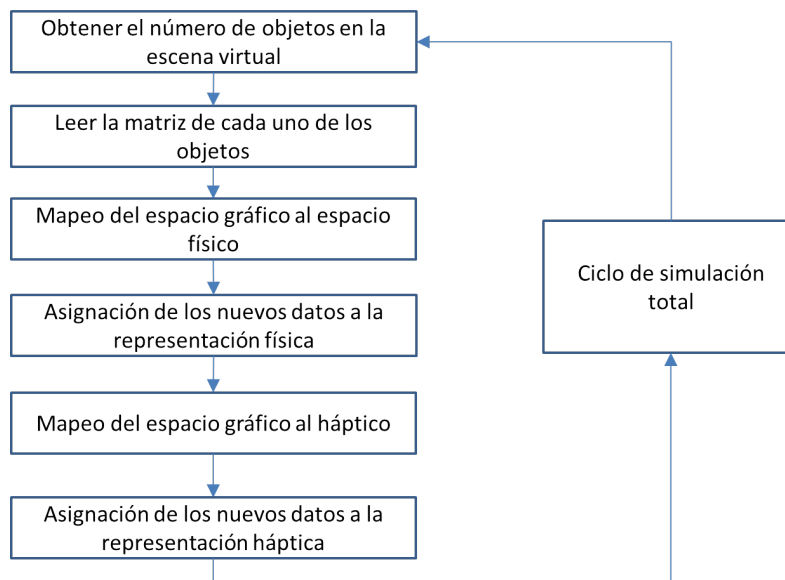


Figura 2.16. Proceso de transformación de coordenadas del espacio gráfico a coordenadas de los espacios físico y háptico.

*Transformación de coordenadas del espacio físico a coordenadas del espacio gráfico.* Durante cada paso de la simulación los resultados de la simulación física deben ser enviados a la representación gráfica para que el usuario pueda observar el movimiento de los objetos virtuales. Para transmitir los resultados de la simulación física a la escena gráfica se utiliza la transformación de coordenadas, la cual se ejecuta de manera continua en cada

ciclo de simulación durante la ejecución del programa. El proceso de transformación de coordenadas físicas a coordenadas gráficas se describe en la Figura 2.17.

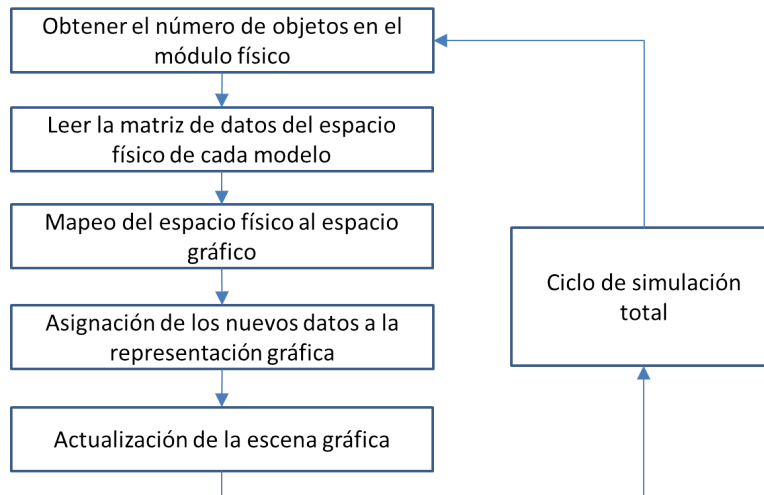


Figura 2.17. Proceso de transformación de coordenadas del espacio físico al espacio gráfico.

## 2.6 Funcionalidad del sistema HAMS

Además de las funciones básicas que integran el sistema de ensamblajes virtuales y que permiten cargar modelos y realizar la manipulación e interacción de los objetos virtuales, el sistema HAMS posee funciones que permiten incrementar su funcionalidad, realismo, flexibilidad y desempeño. Tales funciones son descritas en los siguientes párrafos.

### 2.6.1 Cálculo de la masa del objeto virtual

En la mayoría de los simuladores de física las ecuaciones que describen el movimiento de los objetos se integran a partir de la aceleración, la cual se calcula en base a la fuerza neta ejercida sobre el objeto y a su masa utilizando la segunda ley de Newton  $F = ma$ . Así, los objetos sin masa no tienen movimiento. Por lo tanto, para lograr un comportamiento dinámico real de los objetos virtuales se debe definir su masa de manera adecuada. Además, al definir la masa del objeto virtual el usuario podrá sentir la fuerza de gravedad actuando sobre dicho objeto, es decir, podrá sentir el peso del objeto manipulado por medio del dispositivo háptico.

El primer paso para definir la masa de un objeto es seleccionar el material del cual está compuesto. Con el objetivo de representar diversos materiales en el sistema de ensamblajes virtuales se han añadido cuatro opciones: acero, aluminio, plomo y plástico. Estos materiales se han seleccionado debido a la diferencia en densidad que tienen entre sí, Tabla 2.1.

Los valores de densidad están predefinidos en la aplicación de acuerdo a cada material. Cuando el usuario carga una nueva parte en el sistema HAMS, existe la opción para seleccionar el tipo de material de la pieza, Figura 2.18.

Tabla 2.1. Densidad de materiales utilizados.

| Material                  | Densidad (kg/m <sup>3</sup> ) |
|---------------------------|-------------------------------|
| Acero                     | 7850                          |
| Aluminio                  | 2700                          |
| Plomo                     | 11340                         |
| Plástico( Baja densidad ) | 900                           |

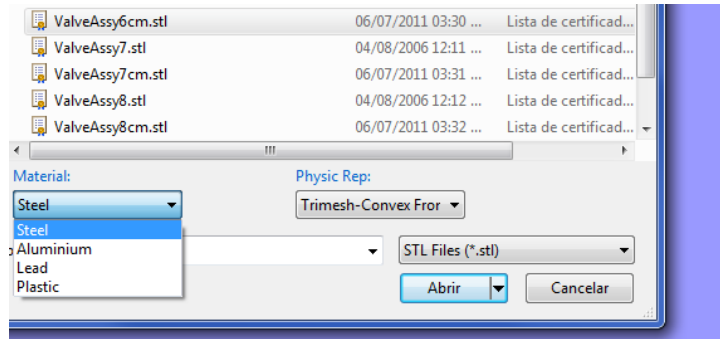


Figura 2.18. Selección de material del objeto virtual.

Para calcular la masa ( $m$ ) de un objeto virtual es necesario obtener el volumen de dicho objeto ( $V$ ) y la densidad del material seleccionado ( $\rho$ ).

$$m = \rho * V \tag{2.2}$$

El simulador gráfico *VTK* ofrece funciones para calcular el volumen de los objetos representados por una malla triangular. Una vez calculado el volumen es posible calcular la masa del objeto virtual. La masa calculada es finalmente asignada a la representación física del objeto, la cual mediante el modelo MRA permite al usuario sentir el peso de los objetos virtuales, este proceso se describe en la Figura 2.19.

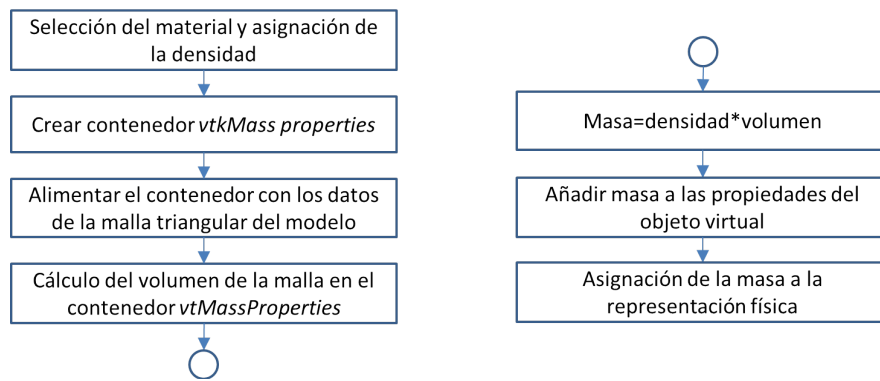


Figura 2.19. Proceso de cálculo y asignación de la masa.



Mediante este método es posible calcular la masa real de los objetos virtuales, sin embargo, los dispositivos hápticos tienen un límite de fuerza a ejercer; por ejemplo, el dispositivo Phantom Desktop permite una fuerza máxima de 7.5 N y una fuerza continua máxima de 1.75 N, mientras que el dispositivo Phantom Omni permite una fuerza máxima de 3.3 N y una fuerza continua máxima de 0.88 N. Por lo tanto, resulta impráctico y riesgoso para el dispositivo utilizar la masa real de los objetos virtuales. Debido a esto, se han escalado los valores de la densidad para que la masa del objeto virtual sea menor a la real. Se ha observado también que una masa menor a una unidad produce que el objeto manipulado sea inestable y muy difícil de controlar. Además, una masa mayor de diez unidades ocasiona que el objeto manipulado se sienta muy pesado y por lo tanto exista la posibilidad de dañar el dispositivo. Por esta razón el valor de masa se ha restringido en el rango de 1 a 10 unidades.

### 2.6.2 Rotaciones del objeto respecto al punto de selección

Cuando un objeto es seleccionado para su manipulación por medio del dispositivo háptico, las translaciones se generan mediante el sistema MRA. Sin embargo, en el caso de las rotaciones, existen dos opciones:

- Rotación respecto al centro. El objeto gira alrededor de su centro geométrico sin importar la ubicación del cursor háptico. En este método solo se especifica la rotación mediante ángulos de Euler y no existe translación del centro geométrico, Figura 2.20a.
- Rotación respecto al punto de selección. El objeto gira alrededor del cursor háptico, es decir, alrededor del punto donde el usuario selecciona al objeto y no del centro geométrico. Para realizar esta rotación se deben calcular los ángulos de giro y la translación del centro geométrico del objeto, Figura 2.20b. Esta última opción es la que ofrece un mayor nivel de realismo y una mejor manipulación, sin embargo, puede afectar el desempeño de la detección de colisiones.

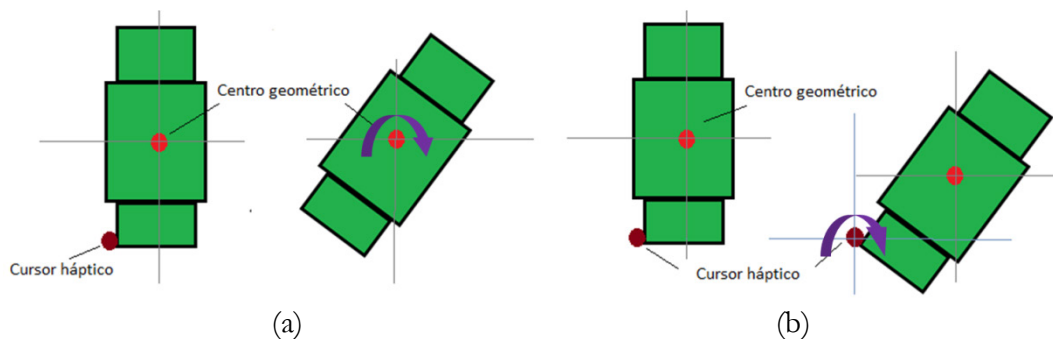


Figura 2.20. Rotación del objeto virtual: (a) rotación respecto al centro geométrico, (b) rotación respecto al punto de selección.

El sistema de ensamblajes virtuales desarrollado en este trabajo soporta los dos modos de rotación, los cuales pueden ser elegidos por el usuario en tiempo de ejecución. El procedimiento para realizar las rotaciones de los objetos virtuales manipulados se muestra en la Figura 2.21. De manera general, el proceso de rotación consiste en calcular un *offset* inicial entre

la posición del dispositivo háptico y la posición del centro geométrico del objeto, así como también calcular un *offset* inicial para las orientaciones de ambos elementos. Este cálculo se realiza solamente una vez cuando se selecciona el objeto. A continuación, en cada ciclo de simulación se lee la posición y orientación actuales del dispositivo háptico y del objeto virtual, con los cuales se calcula un nuevo *offset* para cada parámetro en cada paso de la simulación. La diferencia entre el *offset* actual y el *offset* inicial es introducida en una matriz de rotación. Los resultados de esta rotación son utilizados para especificar la nueva posición y la nueva orientación del modelo gráfico. De este modo el objeto virtual gira con respecto al punto de selección.

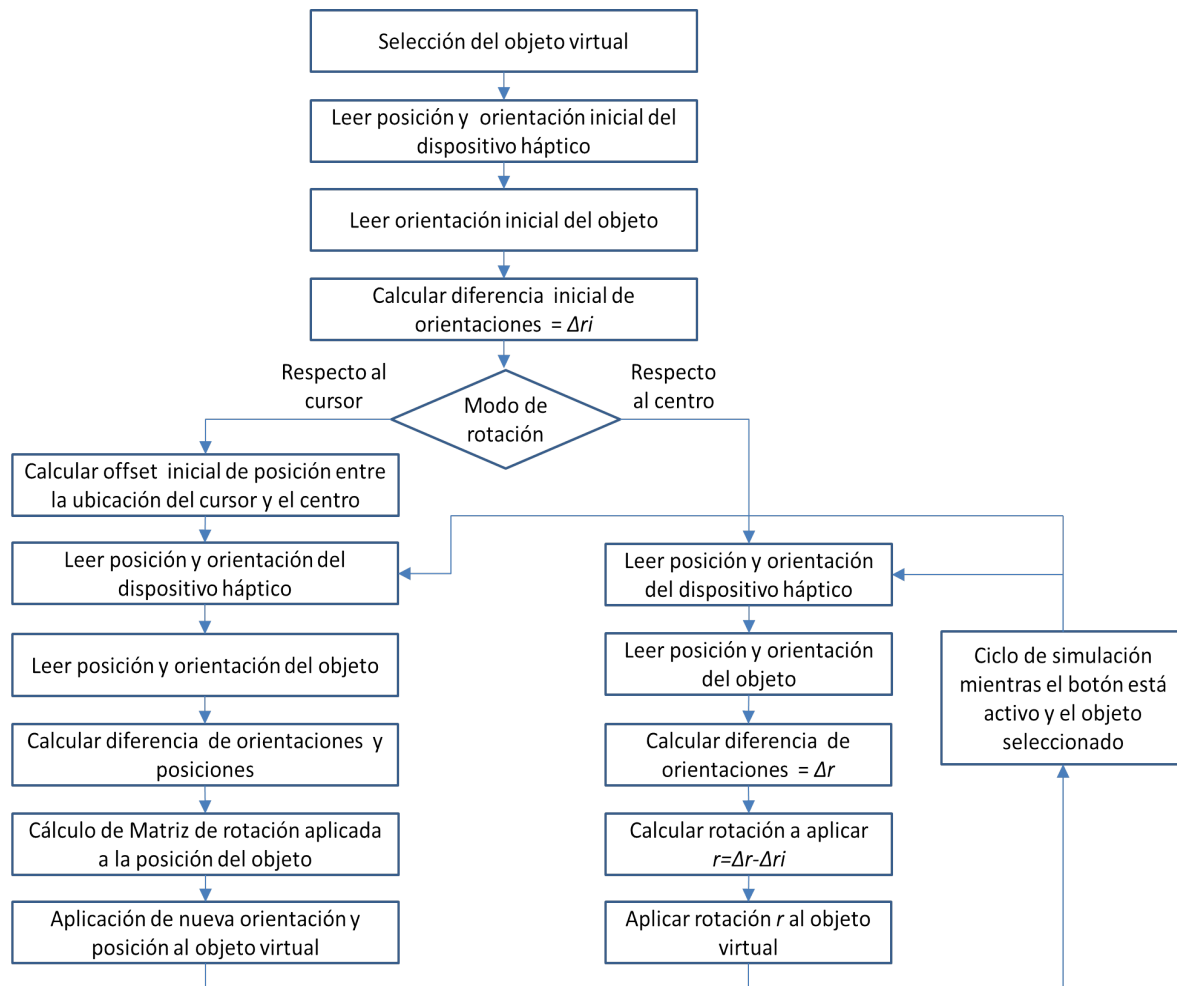


Figura 2.21. Proceso para realizar la rotación del objeto virtual.

Al aplicar la rotación sobre la representación gráfica se actualiza también la representación física, la cual toma la misma posición y orientación del modelo gráfico. Sin embargo, esta rotación no se aplica directamente sobre el modelo háptico, lo cual crea un *offset* entre la representación física y la representación háptica. Este *offset* crea una elongación del resorte en el

sistema MRA que une ambas representaciones, creando así una fuerza de retroalimentación en el dispositivo háptico sin que existan fuerzas externas actuando sobre el objeto virtual manipulado. Por tanto es necesario corregir dicho *offset* aplicando la rotación a la representación háptica; este proceso de corrección se muestra en la Figura 2.22. Mediante este procedimiento se asegura un comportamiento estable al realizar rotaciones del objeto virtual manipulado alrededor de la posición del cursor háptico o punto de selección, Figura 2.23.

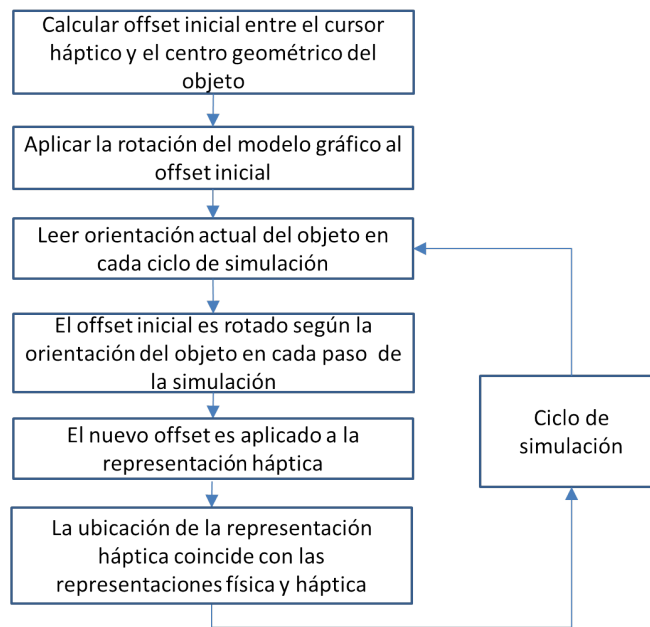


Figura 2.22. Proceso para corregir rotación de la representación háptica.

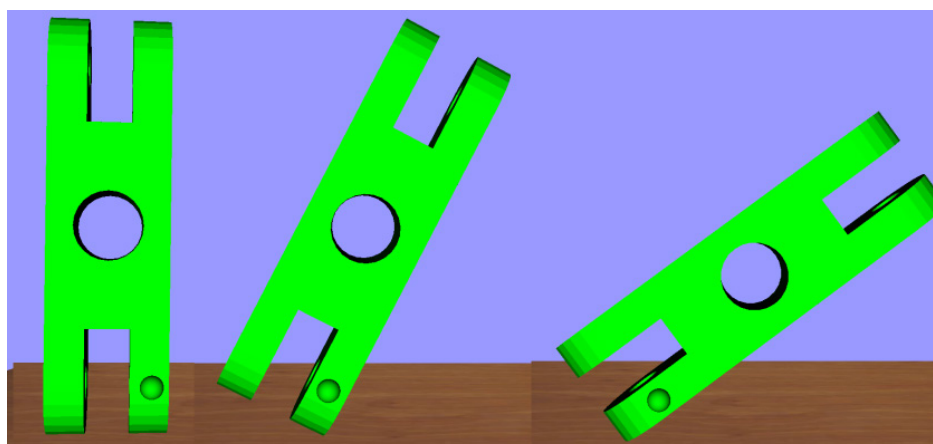


Figura 2.23. Rotaciones alrededor del punto de selección.

### 2.6.3 Modo háptico dual

En el ensamblaje de partes mecánicas es común el uso de las dos manos. Por esta razón, el sistema de ensamblés HAMS incluye el modo háptico dual, el cual se refiere al uso de dos dispositivos hápticos con el fin de habilitar el uso de las dos manos al realizar un ensamblaje, Figura 2.2. La habilitación del modo háptico dual requiere crear un segundo ciclo de simulación háptica (ver secciones 2.3.3 y 2.4 de este capítulo). El sistema HAMS es capaz de identificar automáticamente el número de dispositivos hápticos conectados a la computadora. En caso de detectar un solo dispositivo el sistema activa este dispositivo y el programa trabaja con un solo ciclo de simulación háptica. Por otro lado, en caso de encontrar dos dispositivos el programa habilitará cada uno y realizará un ciclo de renderizado háptico para cada dispositivo. En caso de no encontrar ningún dispositivo se envía un mensaje de advertencia pero el sistema sigue funcionando.

### 2.6.4 Manipulación de cámara

Con el fin de ofrecer mayor libertad de movimiento al usuario, el sistema HAMS permite la manipulación total de la cámara gráfica, es decir, el usuario puede manipular la cámara para seleccionar el punto de vista que le facilite realizar la tarea de ensamblaje. Esta manipulación incluye translaciones en los 3 ejes coordenados, rotación de 360° en cualquier dirección y aumento o alejamiento de la escena (*zoom*). Los modos de manipulación de la cámara pueden ser seleccionados por medio de botones en la barra de menú del sistema. La manipulación de cámara se ejecuta mediante el mouse de la PC o por medio del dispositivo háptico. También es posible seleccionar una vista predefinida de la escena mediante botones en la barra de menú de la aplicación, por ejemplo: vista frontal, trasera, lateral, superior, inferior o isométrica, Figura 2.24.

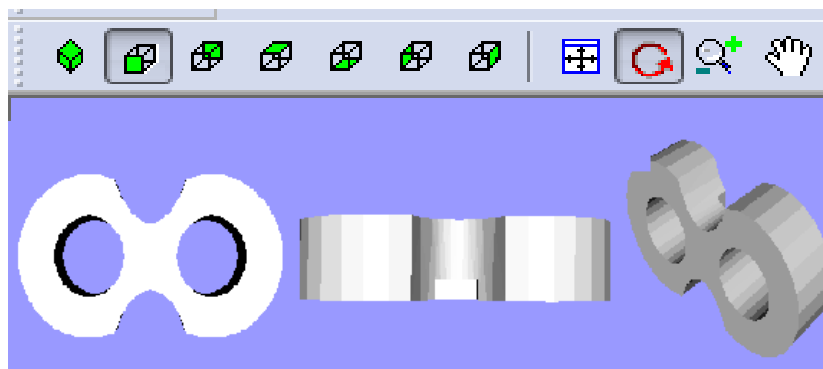


Figura 2.24. Botones de vistas predefinidas y modos de manipulación de cámara.

Uno de los mayores problemas al realizar la manipulación de cámara en ambientes virtuales hápticos es que el mapeo háptico no se actualiza directamente, es decir, al girar la cámara los movimientos del cursor háptico siguen obedeciendo al sistema coordenado original. Por ejemplo, en la posición original de la cámara el eje  $X$  positivo se dirige hacia la derecha, al mover el dispositivo háptico hacia la derecha el cursor también se mueve a la derecha. Sin

embargo, al rotar la cámara 180° sobre el eje Y, el eje X positivo quedará hacia la izquierda, por lo tanto al mover el dispositivo háptico a la derecha, el cursor se moverá hacia la izquierda, siguiendo el eje X positivo original. Este problema afecta el desempeño del proceso de ensamble virtual, pues al girar la cámara los movimientos del cursor gráfico no corresponden con los movimientos del dispositivo háptico. Por lo tanto, se debe actualizar el mapeo háptico cada vez que se realiza un movimiento de la cámara, de tal modo que sin importar los giros o desplazamientos que tenga la cámara, el cursor háptico siempre siga el movimiento del dispositivo háptico de forma consistente.

Para comprender mejor el problema se considera que existen dos cámaras: la cámara gráfica y la cámara háptica. La primera genera el punto de vista desde donde se ven los objetos, mientras que la segunda representa el espacio de trabajo háptico. Cuando no se ha realizado ningún movimiento de la cámara gráfica, las dos cámaras se encuentran en la misma posición, las coordenadas del espacio gráfico y del espacio háptico son consistentes, Figura 2.25a. Sin embargo, al girar la cámara gráfica, la cámara háptica permanece en su posición original, así al mover el cursor a la derecha la cámara gráfica capta su movimiento a la izquierda, Figura 2.25b. Por lo tanto, la solución al problema es trasladar la cámara háptica junto a la cámara gráfica y así lograr la consistencia entre ambos espacios de trabajo, Figura 2.25c.

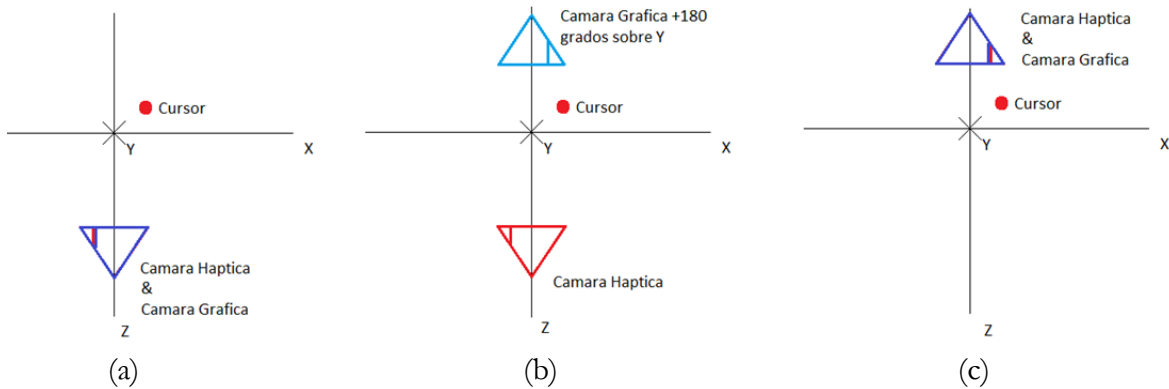


Figura 2.25. Movimiento de cámara gráfica y cámara háptica, (a) posición original, (b) movimiento de cámara gráfica, (c) movimiento de ambas cámaras.

Para realizar el movimiento de la cámara háptica se debe considerar el mapeo háptico, el cual consiste en realizar una serie de transformaciones de coordenadas para pasar del espacio de trabajo gráfico al espacio de trabajo real del dispositivo háptico, Figura 2.26.

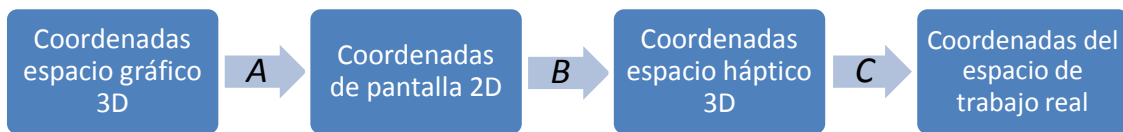


Figura 2.26. Mapeo del mundo virtual al espacio real de trabajo.

donde:

- $A$  Representa la matriz de transformación del espacio virtual 3D al espacio de cámara 2D.
- $B$  Define la rotación y translación del espacio de trabajo háptico relativo a las coordenadas del espacio de cámara (por lo general es la matriz identidad).
- $C$  Es la matriz de transformación para trasladar la escala, posiciones y orientaciones del espacio háptico al espacio de trabajo real y viceversa.

La matriz de transformación  $C$  es responsable de transformar los movimientos reales del dispositivo háptico a movimientos en el espacio háptico. Las coordenadas del espacio háptico pasan entonces a coordenadas de cámara por medio de la transformación  $B$ , para finalmente ser transformadas mediante la matriz  $A$  en coordenadas del espacio gráfico virtual. Estas matrices de transformación son manejadas por las librerías HLAPI, y no se están disponibles para su modificación directa. Sin embargo, HLAPI incluye funciones para manipular cualquiera de las matrices de transformación, por ejemplo,  $hlRotated()$  aplicar una rotación a la matriz y  $hlTranslated()$  aplica una translación. Al utilizar estas funciones para modificar la matriz  $C$  es posible manipular el espacio de trabajo háptico o cámara háptica y así lograr la consistencia entre coordenadas gráficas y hápticas. El procedimiento para la rotación y translación de la cámara háptica con base a los movimientos de la cámara gráfica se describe en la Figura 2.27.

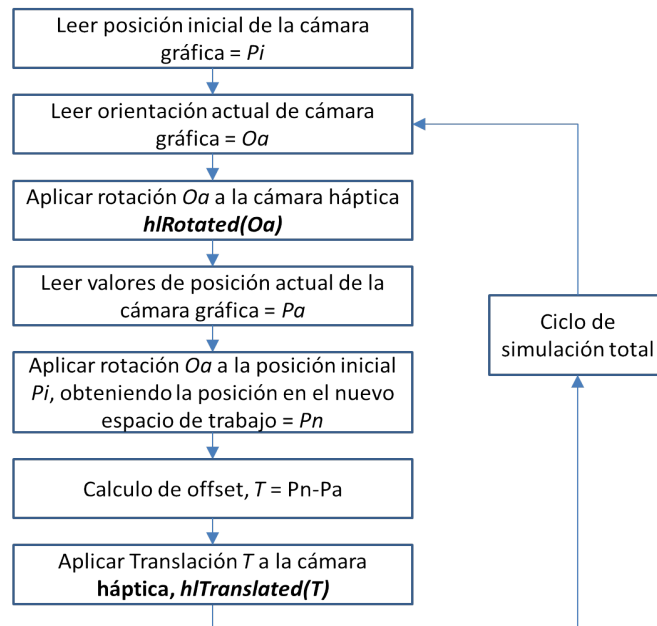


Figura 2.27. Rotación del espacio de trabajo háptico.

### 2.6.5 Registro de la información durante el ensamble

La información generada durante el proceso de ensamble virtual, como por ejemplo las trayectorias de ensamble, posiciones, tiempos de manipulación, secuencia de ensamble, fuerzas, entre otros, son parte fundamental para analizar, planear, evaluar y optimizar el proceso de

ensamble, así como para la generación de un plan de manufactura y/o ensamble. La información más relevante del proceso de ensamble se puede encontrar en los movimientos realizados por el usuario en el ambiente virtual a través del dispositivo háptico, para reconocer la escena, reconocer los objetos virtuales o manipularlos con el fin de llevar a cabo el ensamble. En el sistema HAMS los movimientos realizados por el usuario son identificados y guardados en un archivo de texto de acuerdo a lo siguiente:

- Fase de identificación de la escena, llamada “*wandering*”, en la cual el usuario mueve el cursor háptico en el espacio virtual con el fin de reconocer la escena o posicionar el cursor sobre un objeto. En esta fase se guardan los datos de la posición y orientación del cursor háptico y el tiempo en formato del reloj de la computadora (horas: minutos: segundos. milésimas).
- Fase de reconocimiento del objeto virtual, llamada “*touching*”, en esta fase el cursor háptico está en contacto con un objeto virtual, el usuario se encuentra reconociendo el objeto a través del sentido del tacto. Se registran los valores de posición, orientación, velocidad del cursor, tiempo, y el nombre del objeto tocado.
- Fase de manipulación, llamada “*controlling*”, en la cual el usuario selecciona un objeto virtual y lo manipula con el fin de llevarlo a su posición de ensamble final. En esta fase se registran los datos de posición y orientación del cursor háptico, ángulos del dispositivo háptico, tiempo, posición, orientación y nombre del objeto manipulado.

Estos valores son almacenados en cada ciclo de simulación en un archivo de texto, en el cual se puede identificar fácilmente cada una de las fases anteriormente descritas así como los valores registrados. En la Figura 2.28 se puede observar un ejemplo de dicho archivo.

```

btcontcd1.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
Gimbal Angles : X[-1.118467], Y[0.464900], Z[-0.272186]
Control time : 12:31:15:781
Prop : Position X[0.000125], Y[24.161352], Z[-0.000071]
Prop : Orientation X[-0.000210], Y[-0.000022], Z[-0.000263]
/----- End controlling -----/
/----- Start wandering SD -----/

wander time SD : 12:31:15:781
/----- End wandering SD-----/
/----- Controlling -----/
velocity : 9.302407
Proxy Pos: X[0.883497], Y[41.354522], Z[24.009858]
Proxy Rot: X[0.610800], Y[-0.628960], Z[-0.431868]
Gimbal Angles : X[-1.118467], Y[0.464900], Z[-0.272186]
Control time : 12:31:15:796
Prop : Position X[0.000125], Y[24.161352], Z[-0.000071]
Prop : Orientation X[-0.000210], Y[-0.000022], Z[-0.000263]
/----- End Controlling -----/
/----- Start wandering SD -----/

wander time SD : 12:31:15:796
/----- End wandering SD-----/
/----- Controlling -----/
velocity : 9.302407
Proxy Pos: X[0.883497], Y[41.354522], Z[24.009858]
Proxy Rot: X[0.610800], Y[-0.628960], Z[-0.431868]
Gimbal Angles : X[-1.118467], Y[0.464900], Z[-0.272186]
Control time : 12:31:15:812
Prop : Position X[0.000125], Y[24.161352], Z[-0.000071]
Prop : Orientation X[-0.000210], Y[-0.000022], Z[-0.000263]
/----- End Controlling -----/
/----- Start wandering SD -----/

```





Figura 2.28. Archivo de registro de datos.

Si los datos han sido almacenados, al terminar el proceso de ensamble es posible leer el archivo que contiene esos datos y representarlos de manera gráfica en la misma escena virtual mediante crono-ciclos, Figura 2.29. Los crono-ciclos son una representación gráfica de los movimientos del usuario realizados durante el ensamble, y son una manera intuitiva de observar y analizar la información del proceso de ensamble virtual. Para representar los movimientos del usuario en el ambiente virtual, los crono-ciclos utilizan esferas de diferente color, Tabla 2.2, que simbolizan de manera independiente cada una de las fases - *wandering*, *touching* o *controlling* -.



Figura 2.29. Representación gráfica de los movimientos del usuario.

Tabla 2.2. Representación gráfica de los movimientos del usuario.

| Representación grafica  | Significado  |
|---|--|
|  Esfera verde                | Fase “ <i>Wandering</i> ”, movimientos en el espacio         |
|  Esfera azul                 | Fase “ <i>Touching</i> ”, reconocimiento del objeto virtual  |
|  Esfera roja                 | Fase “ <i>Controlling</i> ”, manipulación del objeto virtual |
|  Espaciamiento entre esferas | Velocidad del movimiento                                     |

Además de los movimientos realizados por el usuario también se pueden guardar datos referentes al desempeño de la aplicación, específicamente tiempos de simulación de cada uno de los tres módulos - físico, gráfico y háptico -. También existe la opción para guardar la fuerza y el torque ejercidos por el dispositivo háptico durante el ensamble. En el caso del dispositivo háptico Omni, solo se utilizan los datos de fuerza pues el dispositivo no cuenta con retroalimentación de torque en los grados de libertad correspondientes a las rotaciones.



### 2.6.6 Manipulación de variables de simulación

Uno de los propósitos principales del sistema HAMS desarrollado es analizar y evaluar parámetros de simulación que afecten el proceso de ensamble. Por tanto, es importante poder modificar estos parámetros durante la ejecución del programa. Con este fin, se han parametrizado las variables principales de cada uno de los módulos de simulación, y se han integrado paneles donde el usuario puede manipular y ajustar los valores de estas variables. Se tiene un panel de propiedades hápticas, un panel de propiedades físicas y un panel de propiedades de la escena virtual, correspondientes a cada uno de los módulos de simulación.

En el panel de propiedades hápticas, Figura 2.30, se incluyen variables referentes a la simulación háptica, por ejemplo botones que permite activar o desactivar el dispositivo háptico, seleccionar el método de renderizado gráfico – *depth buffer* o *feedback buffer* –, controles para ajustar el tamaño del cursor, modificar propiedades como la rigidez, el amortiguamiento y la fricción que el usuario experimenta al tocar un objeto virtual. En este panel es posible modificar también la escala de la fuerza que se aplica a los objetos virtuales, y la rigidez y amortiguamiento del modelo MRA utilizado. Incluye también controles del tacto para activar la función *snap constraint guidance*, la cual permite al cursor pegarse a los objetos cuando se encuentra cerca de ellos. Finalmente también incluye una sección para crear y ajustar las restricciones de ensamble, las cuales son descritas en el capítulo III.

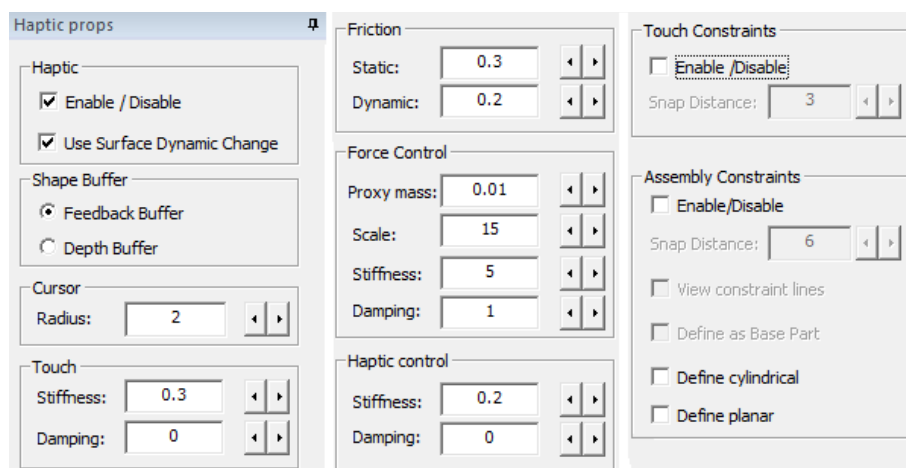


Figura 2.30. Panel de manipulación de propiedades hápticas.

El panel de propiedades físicas, Figura 2.31, permite seleccionar en tiempo de ejecución el simulador de física a utilizar durante la realización del ensamble – Bullet, PhysX v2.8 o PhysX v3.1 –. También permite modificar las propiedades físicas de los objetos virtuales como masa, tolerancia para la detección de colisiones, restitución, gravedad, fricción y ajuste del valor de gravedad. En este panel es posible modificar el paso de la simulación física, es decir el tiempo que utiliza el simulador para actualizar las posiciones de los objetos en cada ciclo de simulación.

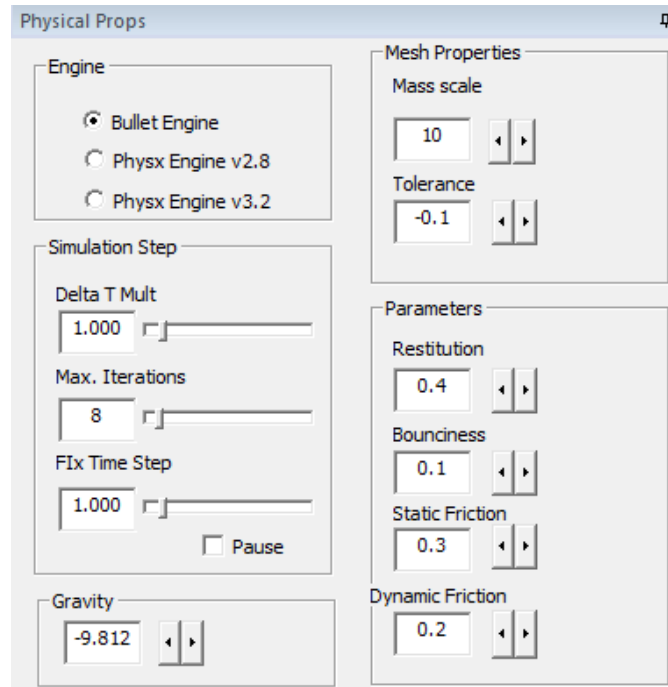


Figura 2.31. Panel de manipulación de propiedades físicas.

Por último el panel de propiedades de la escena virtual, Figura 2.32, contiene información acerca de los objetos en el ambiente virtual, como su nombre, su estado cinemático –dinámico o estático–, su algoritmo de representación y su material. Este panel permite cambiar el modo de visualización de cada uno de los objetos de manera individual, por ejemplo, aplicar una transparencia, hacerlo invisible, renderizarlo como un sólido, como una malla o como un conjunto de puntos, Figura 2.33. También es posible aplicar una textura al objeto para representarlo de una manera más realista. En este panel también se pueden eliminar e insertar objetos de la escena virtual.

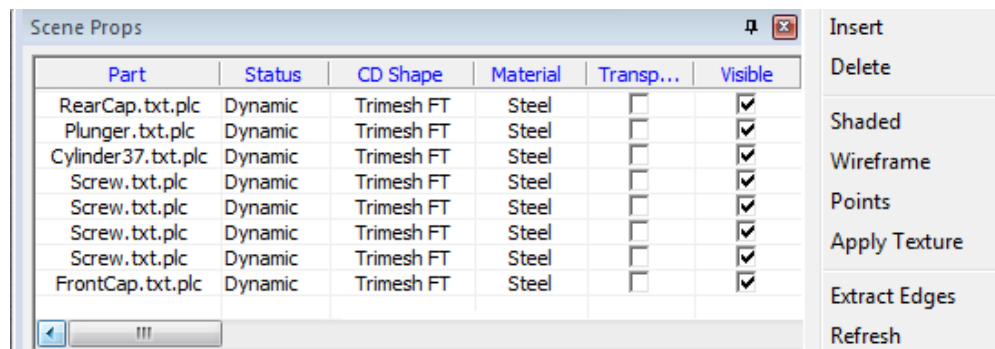


Figura 2.32. Panel de propiedades de la escena virtual.

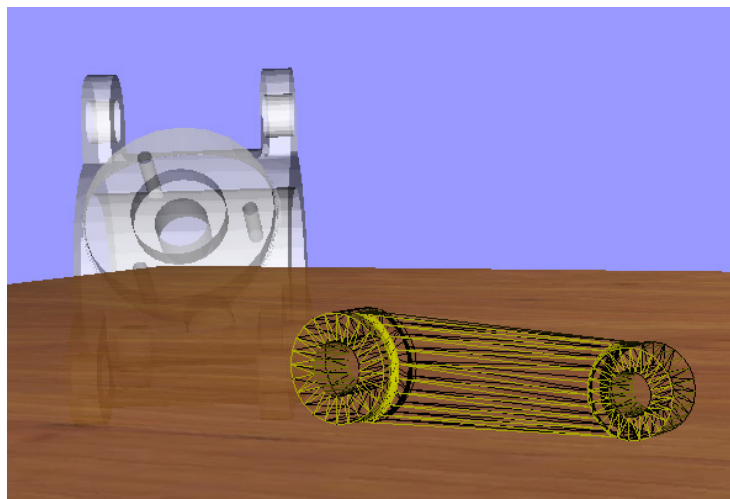


Figura 2.33. Representación de objetos transparentes y por malla.

Cada uno de los paneles es colapsable, ocultándose automáticamente cuando el cursor del mouse no está sobre ellos; esto con el fin de lograr una escena virtual más limpia, la cual permite al usuario enfocarse en el proceso ensamble con la mínima distracción. La Figura 2.34 muestra una escena del sistema HAMS con paneles activos.

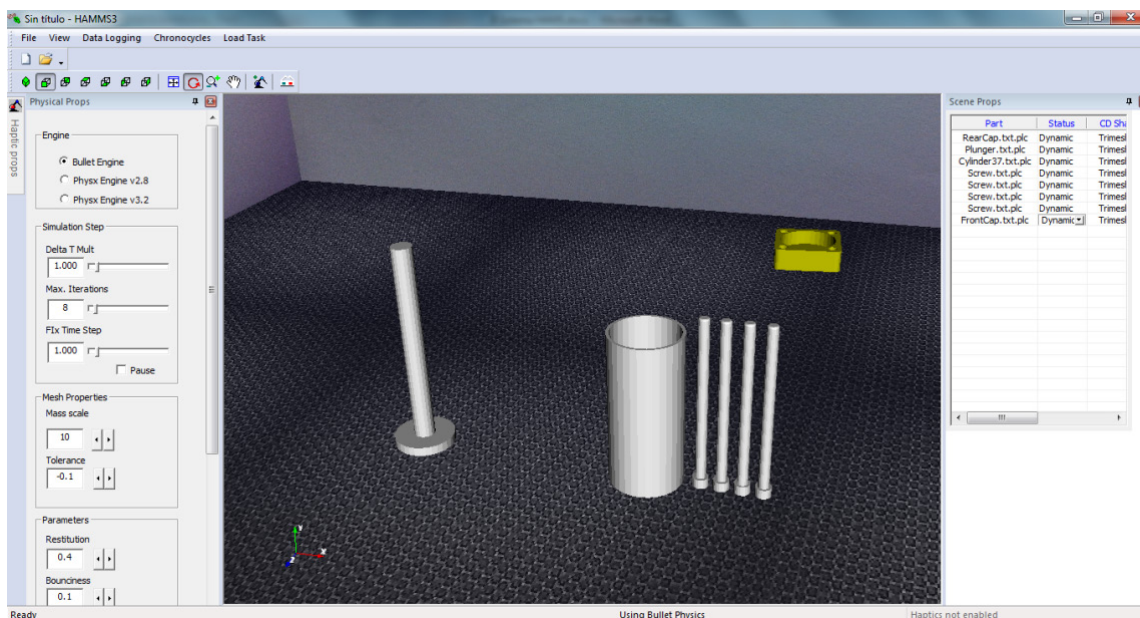


Figura 2.34. Escena principal del sistema HAMS con paneles activos.

### 2.6.7 Widgets para manipulación de propiedades

Los *widgets* son elementos dentro de la escena virtual con los que el usuario puede interactuar de manera directa por medio del mouse o del dispositivo háptico. A estos elementos se les

pueden agregar funciones para realizar acciones dentro del ambiente virtual. Su uso permite obtener una interfaz limpia, amigable y fácil de controlar. En el sistema HAMS los *widgets* forman parte del módulo de simulación gráfica y son utilizados como controles para manipular las variables relacionadas a las restricciones de ensamble, tales como modificación de la distancia de reconocimiento de la restricción, visualización de líneas guía donde existe una restricción, definición de la pieza base, o el uso de restricciones cinemáticas para crear sub-ensamblajes, Figura 2.35. La descripción y el uso de restricciones de ensamble se muestran en el capítulo III. Durante el proceso de ensamble los *widgets* permanecen ocultos y solo se muestran cuando las restricciones de ensamble están activas y el cursor se acerca a la esquina superior izquierda de la pantalla.

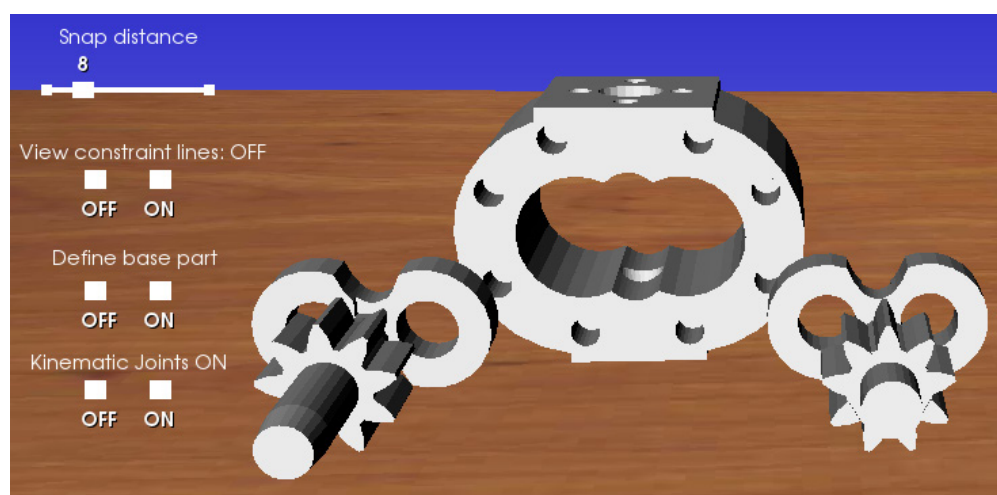


Figura 2.35. *Widgets* para controlar variables de las restricciones de ensamble.

### 2.6.8 Mouse háptico

La función de mouse háptico es una utilidad de las librerías OpenHaptics, la cual permite utilizar el dispositivo háptico como un mouse convencional, es decir, cuando el cursor háptico sale del espacio de trabajo tridimensional se convierte automáticamente en el puntero del mouse, permitiendo así acceder a los botones y menús del marco de la aplicación sin necesidad de utilizar el mouse, como por ejemplo, para cargar un nuevo modelo, cerrar, maximizar o minimizar la aplicación, etc. Por otro lado, al transformar el cursor háptico en el puntero del mouse dentro de la escena virtual es posible controlar la cámara de la escena con el mismo dispositivo háptico, y de este modo girar la cámara, hacer acercamientos, alejamientos o translaciones para obtener el punto de vista deseado. Mediante la función de mouse háptico el sistema HAMS puede ser totalmente controlado utilizando solamente el dispositivo háptico.

### 2.6.9 Visualización inmersiva

Una característica importante que deben tener las aplicaciones de realidad virtual es su capacidad de poder ser ejecutadas en monitores o dispositivos que permitan generar un

ambiente más inmersivo mediante el uso de estereovisión (visión tridimensional), dando la impresión al usuario de que los objetos son reales. El sistema HAMS permite ser ejecutado en este tipo de dispositivos y generar un renderizado tridimensional de la escena virtual mediante funciones específicas del simulador de gráficos VTK. En la Figura 2.36 se muestra el sistema HAMS en un muro de proyección tridimensional.

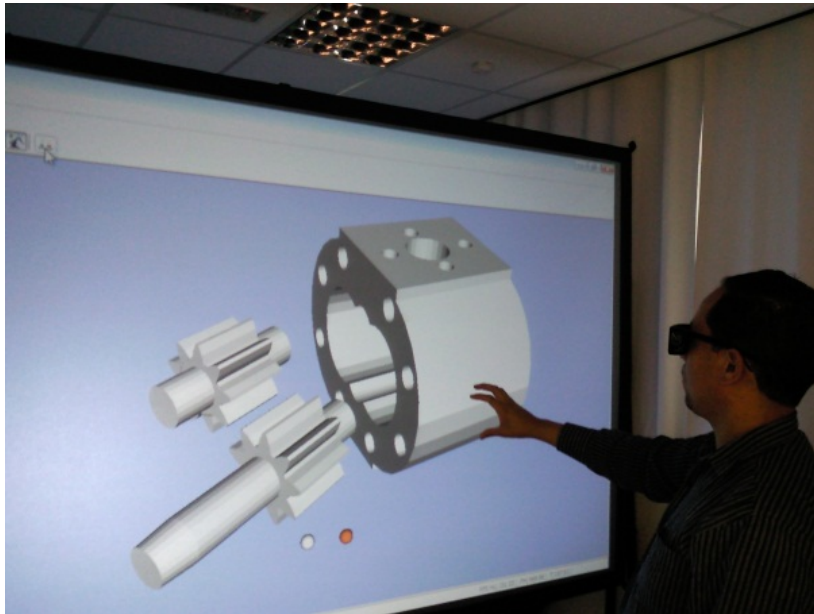


Figura 2.36. Ejecución de HAMS en un muro de proyección tridimensional.

Finalmente se puede decir que con la integración de todas estas funcionalidades y capacidades al sistema HAMS, se ha logrado obtener un sistema o plataforma de simulación al nivel, o incluso por encima, de las plataformas existentes reportadas en la bibliografía. Dichas funcionalidades o capacidades del sistema HAMS permiten llevar a cabo la evaluación del sistema y el análisis de diversos factores de influencia en el desempeño de las tareas de ensamble virtual.

### 3.1 Introducción

En un sistema de ensambles virtuales los objetos deben tener la capacidad de interactuar entre ellos como si fueran objetos reales. Asimismo, en un ambiente asistido por dispositivos hápticos el usuario debe sentir estos contactos y ser guiado por ellos. Este tipo de interacción permite que se pueda llevar a cabo el proceso de ensamble tal y como se realiza en el mundo real. De acuerdo a Seth, *et al.*, [48] (2011) y Xia, *et al.*, (2013) [13] existen dos metodologías con las cuales se puede lograr este tipo de interacción: (i) el modelado basado en física (*Physics Based Modeling*, PBM) y (ii) el modelado basado en restricciones (*Constraint Based Modeling*, CBM). Cada metodología tiene sus ventajas y desventajas, sin embargo, algunos investigadores han propuesto el modelado híbrido el cual considera el uso simultáneo de las dos metodologías para aprovechar las ventajas de cada una, minimizando sus inconvenientes.

El sistema HAMS originalmente estaba diseñado bajo el modelado basado en física, pero también se ha incluido el uso de restricciones de ensamble para ayudar al usuario en la fase final del empuje de los objetos a ensamblar, por lo tanto, se le puede considerar como un sistema que utiliza el enfoque híbrido como metodología de ensamble.

### 3.2 Modelado basado en física

El modelado basado en física (PBM) simula propiedades físicas reales como el efecto de la gravedad, la fricción, la restitución y las fuerzas de contacto para lograr el comportamiento dinámico de los objetos virtuales. La interacción entre los objetos virtuales es creada por medio de la detección y respuesta a colisiones entre ellos, generando una fuerza de reacción que obedece las leyes de la mecánica Newtoniana y que previene que los objetos se traslapen. Esta característica habilita el proceso de ensamble de componentes de una manera muy similar al proceso real.

Para utilizar el modelado basado en física es necesario un simulador de la física, el cual es el encargado de realizar todos los cálculos referentes al comportamiento dinámico de los objetos incluyendo la detección y respuesta a colisiones. Como se describió en el capítulo anterior, el sistema HAMS cuenta con tres simuladores de física –Bullet, v2.81, PhysX v2.8.4 y PhysX v3.1– los cuales pueden ser seleccionados por el usuario durante la ejecución del programa de acuerdo a las características de la simulación. Cualquier simulador de física requiere de al menos tres elementos principales: la escena virtual física, el temporizado de la simulación, y la representación física de cada objeto en el mundo virtual.

### 3.2.1 Escena virtual física

La escena virtual física se refiere al espacio virtual donde se realiza la simulación física, es decir, el espacio en el cual los objetos virtuales están sujetos a un campo gravitatorio y tienen comportamiento dinámico basado en la física real. La escena virtual física es invisible al usuario y por lo general es de mayores dimensiones que el espacio gráfico, incluso puede ser considerada de tamaño infinito. En el sistema HAMS la escena virtual física está delimitada por seis planos que forman un cubo. Las dimensiones de dicho cubo se establecen en base al tamaño de los objetos que se agregan al sistema; así, cuando se inserta un nuevo objeto se extraen sus medidas en los tres ejes coordenados, la mayor se duplica y este valor se utiliza para definir las dimensiones de cada lado del cubo. Al agregar un objeto nuevo se recalculan las dimensiones sumando el tamaño máximo de cada objeto. El centro del plano inferior del cubo se ubica en el centro del espacio de trabajo, es decir en la coordenada  $x = 0$ ,  $y = 0$ , y  $z = 0$ , y en base a este plano se crean los cinco planos restantes. La escena virtual física tiene un valor de gravedad de 10 unidades, que son equivalentes al valor de gravedad de  $9.81 \text{ m/s}^2$  de acuerdo a las dimensiones de los modelos utilizados.

### 3.2.2 Ciclo de simulación

La simulación física de una escena virtual consiste en la variación de la posición, velocidad y aceleración de los objetos virtuales con respecto al tiempo. Esta simulación es hecha por pasos, en cada uno de los cuales la simulación avanza de acuerdo a un intervalo de tiempo definido, que puede ser fijo o variable, resolviendo en cada paso el estado de movimiento de los objetos virtuales. Para una simulación en tiempo real la aplicación debe realizar varios de estos pasos de simulación para sincronizar el comportamiento físico con el renderizado gráfico.

En el sistema HAMS cada paso de la simulación física es realizado dentro del ciclo de simulación total de la aplicación, el cual se ejecuta de manera continua durante el funcionamiento del programa, cómo se detalló en la sección 2.3. El proceso para calcular el tamaño del paso o intervalo de tiempo de la simulación física se muestra en la Figura 3.1. En el caso de seleccionar un tamaño de paso fijo se utiliza siempre este tiempo para avanzar la simulación. En el caso de utilizar un tamaño de paso variable se lee el tiempo actual del procesador, al cual se le resta el tiempo del procesador del ciclo anterior y el resultado es utilizado para adelantar la simulación. La magnitud del paso de simulación, ya sea fija o variable, puede ser escalada por medio de los controles en el panel de propiedades físicas descrito en el capítulo II. Lo anterior con el propósito de analizar el efecto del paso de simulación en el desempeño de la plataforma y también poder ajustar la simulación a las condiciones particulares de cada ensamble.

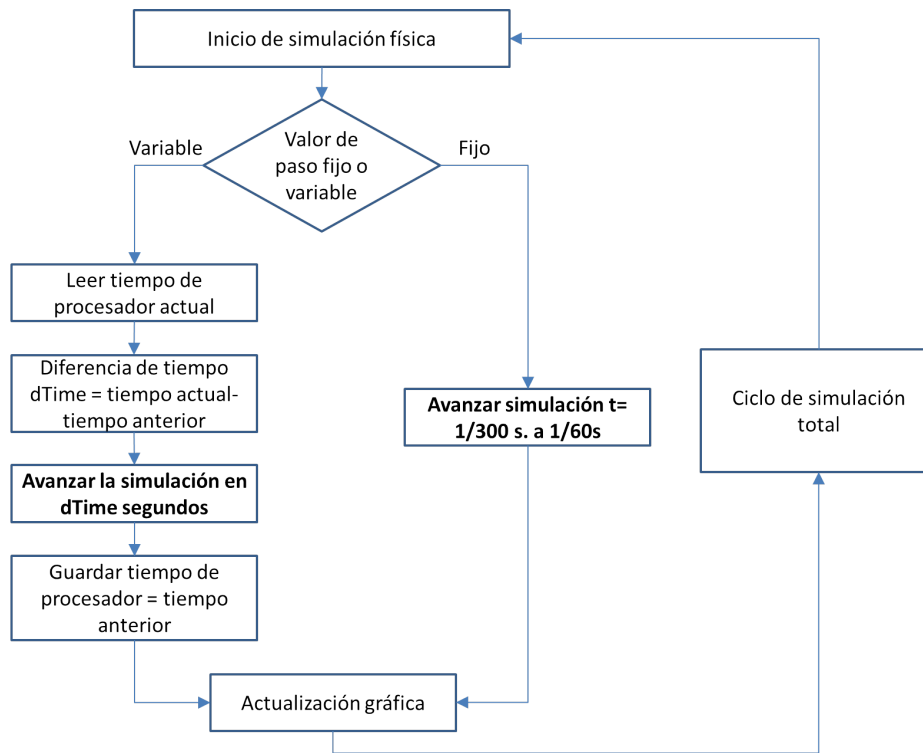


Figura 3.1. Proceso de asignación del tiempo de paso y avance de la simulación.

### 3.2.3 Representación física

La representación física se refiere a la representación de los objetos virtuales en la escena física, la cual es invisible al usuario y puede ser completamente diferente al objeto virtual. La representación física es la encargada de dar el comportamiento dinámico basado en la física real a cada uno de los objetos virtuales. Su principal función es la detección de colisiones entre los objetos del ambiente virtual. Por tanto, el nivel de realismo y buen desempeño de una aplicación pueden depender en gran medida de esta representación. En aplicaciones donde no se requiere de una detección de colisiones precisa, por ejemplo animaciones o videojuegos, la geometría de la representación física tiene poca influencia en el desempeño del sistema. Por ejemplo, la representación física de un engrane puede ser simplemente un cilindro cuando no importa la geometría de los dientes del engrane, Figura 3.2.

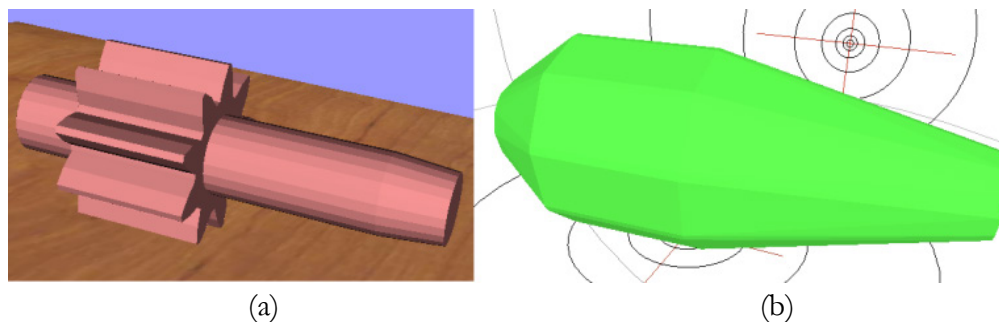


Figura 3.2. Representación de un engrane: (a) representación gráfica, (b) representación física.



Para aplicaciones que requieren de mayor precisión en la detección de colisiones, como es el caso de los sistemas de ensamble virtual, la representación física debe ser lo más precisa posible respecto a la geometría del objeto original (González, *et al.*, 2012 [147]), pues de lo contrario podrá existir traslape entre objetos, o incluso, la imposibilidad de ensamblar objetos que sí pueden ser ensamblados en la realidad.

Los simuladores de física utilizan diferentes métodos o algoritmos para crear la representación física dentro de la escena, siendo los más comunes:

1. **Representación por primitivos**, en la cual el objeto es representado como un cubo, un cilindro, una cápsula, una esfera. etc., sin importar la geometría del modelo original.
2. **Representación convexa**, en la cual el modelo descrito por una malla triangular está cubierto por una envoltura, por lo tanto no posee huecos o concavidades, es decir, el objeto siempre será convexo aunque la malla triangular no lo sea. En la Figura 3.3a se observa el modelo original de una carcasa, mientras que su representación convexa se observa en la Figura 3.3b.
3. **Representación por descomposición en elementos convexos**, en la cual un objeto descrito por una malla triangular es descompuesto en múltiples elementos convexos, los cuales al final son unidos para formar un objeto no convexo, es decir, un objeto o representación con huecos y con cavidades, Figura 3.3c.
4. **Representación de formas arbitrarias por medio de mallas triangulares**, en la cual se utiliza la misma malla triangular que describe la superficie de un objeto para crear la representación física con la mayor precisión posible, Figura 3.3d.

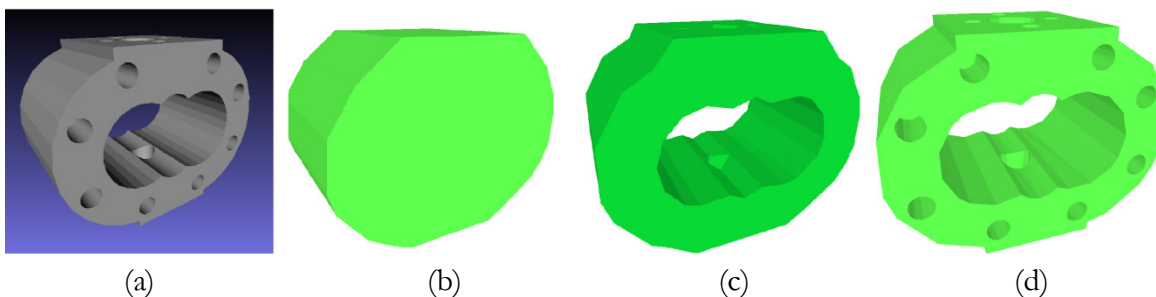


Figura 3.3. Carcasa de bomba de aceite: (a) modelo original, (b) representación convexa, (c) representación por unión de elementos convexos, (d) representación por malla triangular.

Los simuladores de física integrados en el sistema HAMS utilizan algunos de estos métodos para crear sus representaciones físicas. En la Tabla 3.1 se describen los métodos de representación física que emplea cada simulador.

Tabla 3.1. Métodos de representación de modelos en los simuladores físicos.

| Simulador         | Método                        | Descripción  |
|-------------------|-------------------------------|--|
| Bullet v2.81      | Primitivos                    | Pueden ser esferas, cubos, cilindros, capsulas o conos.  |
|                   | Rep. Convexa                  | El modelo es representado como un objeto convexo (el objeto está encerrado por una envoltura).                   |
|                   | HACD (Mamou, 2009 [148])      | <i>Hierarchical Approximate Convex Decomposition</i> , el objeto es descompuesto en pequeños elementos convexos. |
|                   | GIMPACT (Gimpact, 2011 [149]) | Utiliza la malla triangular para la detección de colisiones.   |
| PhysX v2.8 y v3.1 | Primitivos                    | Pueden ser esferas, cubos o capsulas.  |
|                   | Rep. Convexa                  | El modelo es representado como un objeto convexo (el objeto está encerrado por una envoltura).                   |
|                   | ACD (Ratcliff, 2011 [150])    | <i>Approximate convex decomposition</i> , el objeto es descompuesto en pequeños elementos convexos.              |
|                   | HACD (Mamou, 2009 [148])      | <i>Hierarchical Approximate Convex Decomposition</i> , el objeto es descompuesto en pequeños elementos convexos. |

Los procesos de ensamble involucran piezas con geometría muy variada, desde elementos muy sencillos, como pernos, hasta formas muy complejas con geometrías especiales. Sin embargo, de manera general se puede decir que un proceso de ensamble involucra tanto objetos convexos como objetos cóncavos. Debido a esto, la representación por primitivos y la representación convexa no son apropiadas para crear todos los modelos físicos en un sistema de ensambles virtuales, pues la respuesta a colisiones imposibilitaría ensamblar representaciones primitivas o convexas de objetos que originalmente son cóncavos. Por ejemplo, sería imposible insertar una pieza dentro de la carcasa mostrada en el Figura 3.3b. Por tanto, la representación por descomposición en elementos convexos y la representación por mallas triangulares son los métodos que ofrecen una detección de colisiones más realista entre los objetos virtuales.

El método de representación por mallas triangulares representa una mejor precisión geométrica, sin embargo, no todos los simuladores físicos soportan este tipo de representación, pues la detección de colisiones entre mallas triangulares con formas arbitrarias es de alta complejidad y alto costo computacional. En la Tabla 3.1 se puede observar que PhysX en cualquiera de sus dos versiones más recientes no soporta la detección de colisiones entre mallas triangulares, por lo que se deben utilizar los métodos de descomposición en elementos convexos. Dos algoritmos de descomposición en elementos convexos, - ACD (*Approximate Convex Decomposition*) y HACD (*Hierarchical Approximate Convex Decomposition*) -, han sido implementados en los simuladores de física utilizados en el sistema HAMS. En la Figura 3.4 se muestra el procedimiento para la creación de objetos por medio de estos dos algoritmos de descomposición.

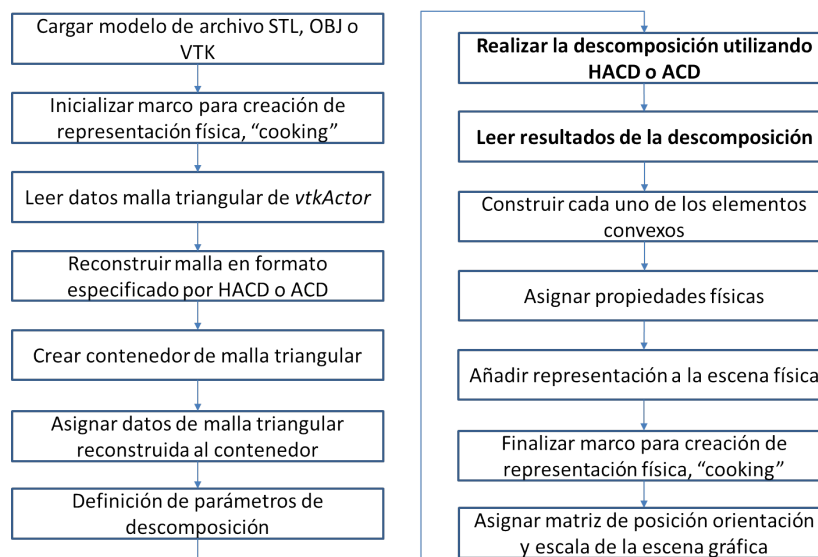


Figura 3.4. Proceso de creación del modelo por descomposición en elementos convexos.

Mediante la descomposición en elementos convexos es posible crear una representación física con geometría muy aproximada, pero no exacta, del modelo original, Figura 3.5. Cada algoritmo, ACD (Figura 3.5b) o HACD (Figura 3.5c) produce resultados diferentes, por lo tanto es necesario realizar una evaluación más detallada de este tipo de representaciones y su uso en las tareas de ensamble virtual.

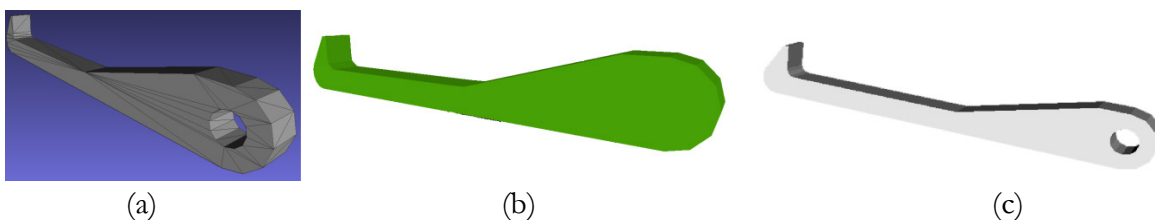


Figura 3.5. Resultado de la descomposición en elementos convexos: (a) modelo original, (b) algoritmo ACD, (c) algoritmo HACD.

### 3.2.4 Representación por elementos convexos creados desde la malla triangular

Con el fin de habilitar la respuesta a colisiones entre objetos representados por mallas triangulares en simuladores físicos que no soportan este tipo de contactos, como PhysX, se propone un método nuevo que consiste en crear un elemento convexo a partir de cada polígono existente en la malla triangular que describe al objeto. Este nuevo método, llamado ConvexFT (*Convex From Triangle*), pretende lograr una representación física más cercana al modelo original, pues cada triángulo es transformado en un elemento convexo utilizando las mismas dimensiones. La unión de todos estos elementos convexos conduce a la forma del modelo original. El procedimiento para transformar los triángulos en elementos convexos se describe en la Figura 3.6.

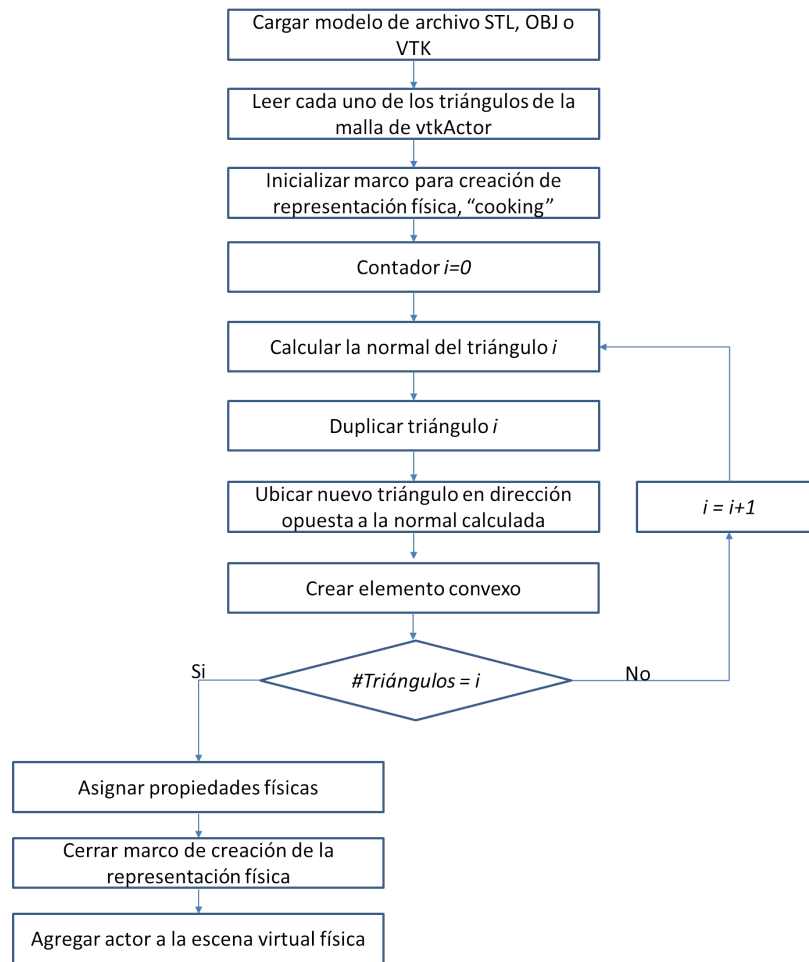


Figura 3.6. Proceso de conversión de cada elemento de la malla triangular en un elemento convexo.

Para crear un elemento convexo se requiere definir al menos 4 vértices que no se encuentren en el mismo plano. En este método cada nuevo elemento convexo está definido por seis vértices, tres del triángulo original y tres del triángulo duplicado, el cual es colocado hacia adentro de la superficie del modelo, a una distancia de 0.1 unidades en dirección opuesta a la normal del triángulo original. En la Figura 3.7 se puede observar de manera gráfica el proceso de creación de esta representación.

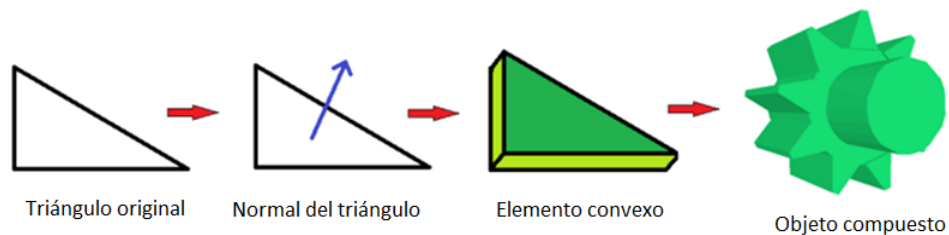


Figura 3.7. Proceso de creación de la representación física utilizando el método ConvexFT

Los resultados del algoritmo ConvexFT para el caso de la carcasa de la bomba se muestran en la Figura 3.8, donde se puede observar que la representación física utilizando el método ConvexFT, Figura 3.8c, tiene una mayor precisión geométrica comparado contra la representación por descomposición en elementos convexos (ACD), Figura 3.8b. De pruebas preliminares sin utilizar la manipulación háptica, se observó que la respuesta a colisiones entre objetos representados por el método ConvexFT es aceptable. Sin embargo, debido a la alta cantidad de elementos convexos creados, la tasa de actualización de la simulación física disminuye, por lo que es necesario realizar más pruebas para determinar el efecto de este nuevo método en el desempeño de aplicaciones de ensamble virtual.

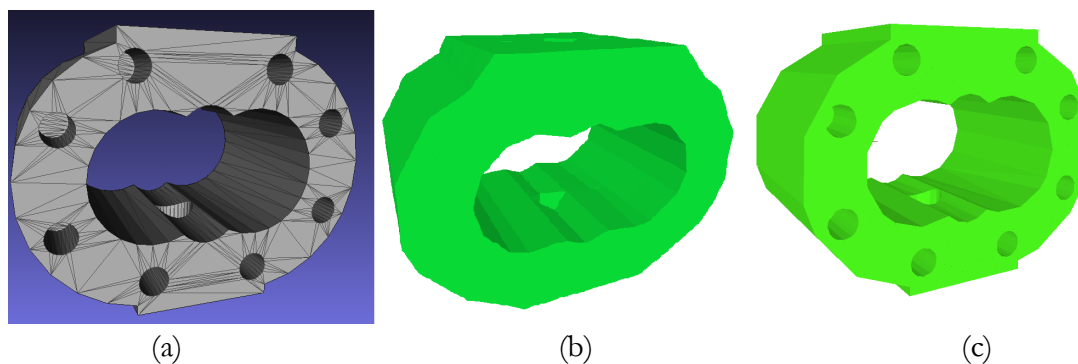


Figura 3.8. Carcasa de bomba de aceite: (a) modelo original, (b) representación por descomposición en elementos convexos (ACD), (c) representación por método ConvexFT.

Es importante mencionar que la detección de colisiones para objetos de pared delgada, como es el caso de los elementos convexos creados por este método, puede presentar algunas fallas. La unión entre cada uno de los elementos convexos puede representar problemas de detección de colisiones, pues dos elementos contiguos que representan una superficie cóncava, Figura 3.9a, tendrán un espacio vacío en su unión y no habrá detección de colisiones. Por otro lado, dos elementos contiguos que forman una superficie convexa, Figura 3.9b, tendrán un traslape en el cual la detección y respuesta a las colisiones no es precisa, además la interferencia entre los dos elementos disminuye el desempeño de la simulación.

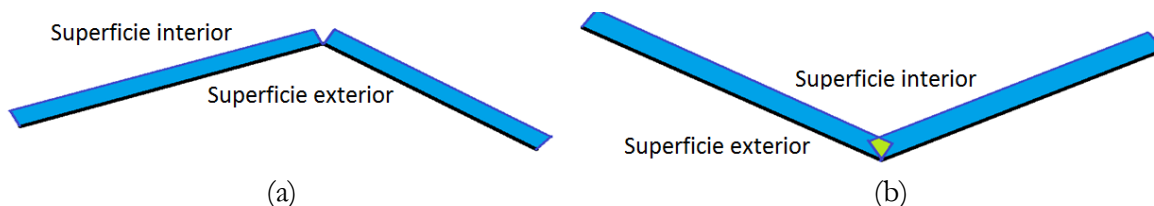


Figura 3.9. Elementos convexos adyacentes: (a) vacío en superficies cóncavas, (b) traslape en superficies convexas.

Se han realizado pruebas para determinar la influencia que tiene el modo de representación física en el desempeño de las tareas de ensamble, la descripción de las pruebas realizadas y los resultados obtenidos son mostrados en los capítulos IV y V de esta tesis.

### 3.3 Modelado basado en restricciones

El modelado basado en restricciones utiliza restricciones geométricas previamente definidas, las cuales reducen los grados de libertad de la pieza manipulada de acuerdo a la posición final de ensamble, resultando en un movimiento restringido del objeto que permite ubicar con mayor facilidad los objetos en su posición y orientación final de ensamble. Las restricciones de ensamble son creadas de una manera similar a la forma en que se definen las restricciones en un sistema CAD, en donde el movimiento del objeto a ensamblar puede estar restringido a ser concéntrico, paralelo, perpendicular, angular, etc., con respecto a una pieza base.

La Figura 3.10 muestra un ejemplo del ensamble basado en restricciones de un perno sobre un cubo. En este ejemplo el perno es manipulado y el cubo es la pieza base, ambos objetos tienen definidas restricciones de concenricidad para alinear ambas formas cilíndricas al realizar el ensamble. En la Figura 3.10a las restricciones son mostradas como líneas en el centro de las formas cilíndricas de los objetos. En la Figura 3.10b se observa que una vez que las restricciones coinciden, el objeto manipulado se alinea y solo se podrá trasladar o rotar sobre el eje de restricción.

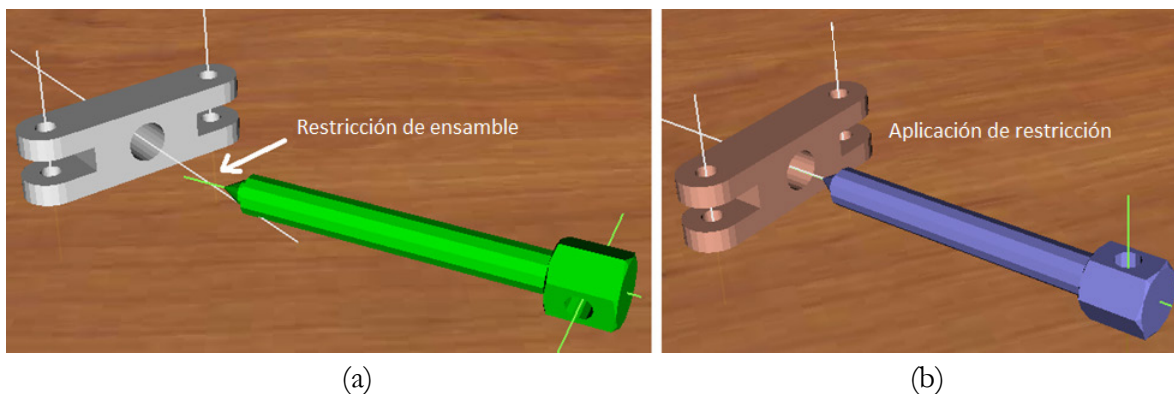


Figura 3.10. Restricciones de ensamble: (a) visualización de las restricciones, (b) aplicación de las restricciones.

El modelado basado en restricciones tiene la ventaja de bajo costo computacional debido a que se eliminan algunos grados de libertad de la pieza manipulada, simplificando así su movimiento. Además, las restricciones de movimiento reducen el traslape entre los objetos, eliminando el proceso de detección y respuesta a colisiones, el cual consume una gran parte del tiempo de simulación. Sin embargo, debido a la ausencia de respuesta a las colisiones, la ausencia del efecto de la gravedad y los movimientos restringidos, el nivel de realismo y de

interactividad de la simulación se reducen dado que los objetos no se comportan como objetos reales.

Cuando el modelado basado en restricciones es integrado a un sistema basado en física es posible mejorar el desempeño del sistema y al mismo tiempo mantener el nivel de realismo y comportamiento intuitivo.

### 3.3.1 Restricciones de ensamble dinámicas (DACs)

Para mejorar el desempeño del sistema HAMS se ha desarrollado e implementado el uso de restricciones de ensamble dinámicas (*Dynamic Assembly Constraints*, DACs), las cuales permiten limitar el movimiento de la parte manipulada con respecto a la pieza base, manteniendo el comportamiento dinámico de los objetos virtuales. Las DACs ayudan a reducir la carga computacional de la detección de colisiones durante el proceso de ensamble; esto debido a que el sistema guía o restringe el movimiento del objeto manipulado por el usuario hasta alcanzar la posición y orientación de ensamble final.

La activación de las DACs durante el proceso de ensamble ocurre cuando la parte manipulada se encuentra cerca de su posición de ensamble, dentro de un rango definido (por el usuario) respecto a la pieza base. Al activarse la restricción, la pieza manipulada es alineada con respecto a la orientación de la pieza base, y su movimiento se restringe para que pueda ser llevada con facilidad a su posición final de ensamble. Una característica clave de las DACs es que el comportamiento dinámico y la respuesta a colisiones de los objetos virtuales se mantienen durante todo el proceso. Por lo tanto, el sistema conserva el nivel de realismo e interactividad, y al mismo tiempo se reduce la carga de la detección de colisiones del objeto manipulado, pues al alinearlo y restringir su movimiento las colisiones se minimizan.

En el sistema HAMS se han desarrollado e implementado dos tipos de restricciones de ensamble:

1. *Restricciones cilíndricas*, las cuales son aplicadas a objetos con formas cilíndricas; por ejemplo, pernos o flechas que se ensamblan en un cubo o viceversa. Este tipo de restricción se define mediante un eje que coincide con el centro de la forma cilíndrica, y sobre el cual el objeto manipulado únicamente puede rotar y trasladarse cuando está activa la restricción, Figura 3.10.
2. *Restricciones planares*, las cuales son aplicadas a superficies planas de un objeto. Este tipo de restricción está definida por un plano que coincide con la superficie a restringir del modelo. Cuando esta restricción está activa el objeto manipulado puede trasladarse y rotar solo sobre el plano de restricción, Figura 3.11.

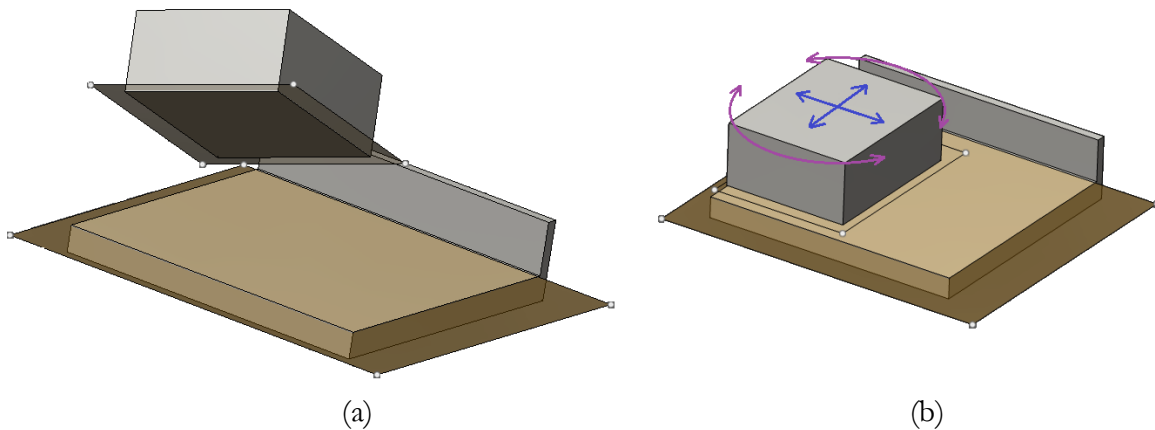


Figura 3.11. Restricciones de ensamble planares: (a) restricciones en cada plano, (b) aplicación de la restricción.

Para utilizar las restricciones de ensamble, ya sean cilíndricas o planas, se requiere tener una pieza base sobre la cual se ensamblan los objetos manipulados. Además, tanto la pieza base como la pieza manipulada deben contar con restricciones del mismo tipo para que puedan ser aplicadas durante el ensamble. Las restricciones de ensamble de cada una de las piezas deben ser definidas de manera previa a la simulación del ensamble, preferentemente al cargar el modelo en el sistema. En la Figura 3.12 se muestra el proceso utilizado en el sistema HAMS para la aplicación de las DACs durante la simulación.

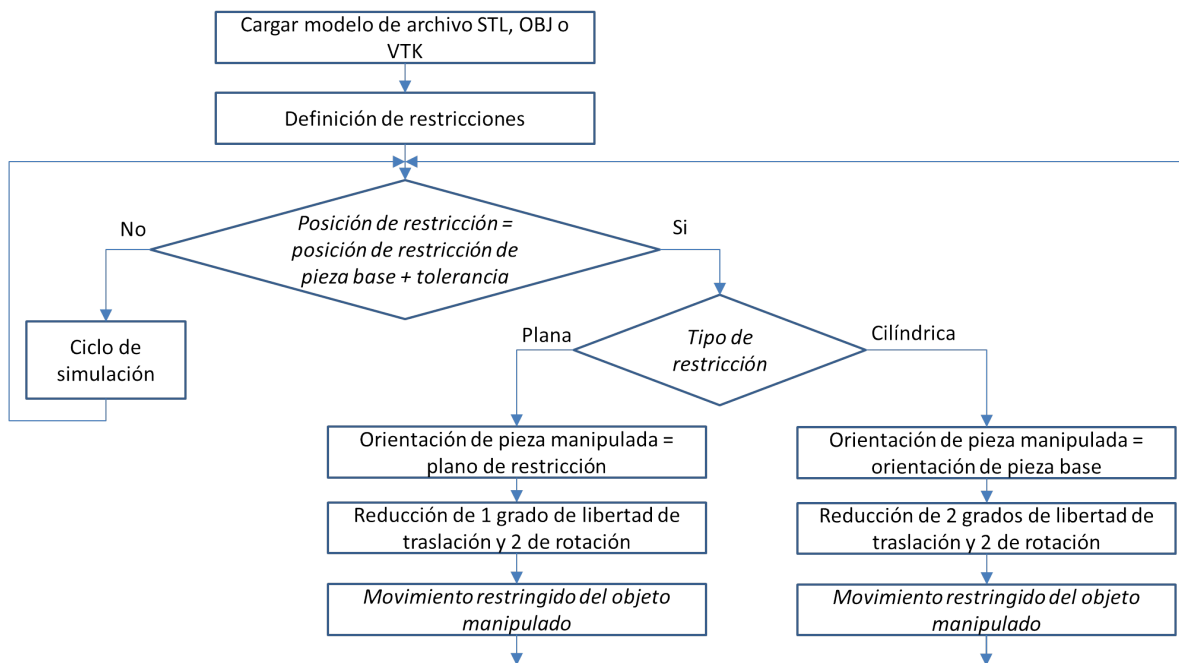


Figura 3.12. Aplicación de las restricciones de ensamble.



### 3.3.2 Definición de restricciones de ensamble

Las restricciones de ensamble pueden ser utilizadas siempre y cuando estén previamente definidas. En el sistema HAMS las restricciones pueden ser definidas de dos maneras: manual mediante el dispositivo háptico, y automáticamente sin la intervención del usuario.

#### 3.3.2.1. Definición manual de restricciones de ensamble

El modo manual para la definición de restricciones de ensamble, permite la intervención del usuario para definir cada una de las restricciones de ensamble a las que estará sujeto el modelo virtual durante el proceso de ensamble. Cuando un modelo ha sido cargado en el sistema, es posible iniciar el modo manual para la definición de restricciones al seleccionar la función correspondiente en el menú de propiedades hápticas, Figura 3.13. En este menú también es posible definir la pieza base, activar o desactivar el uso de restricciones durante el ensamble, seleccionar la distancia o rango de activación de las DACs, y activar la visualización de las líneas de restricción.

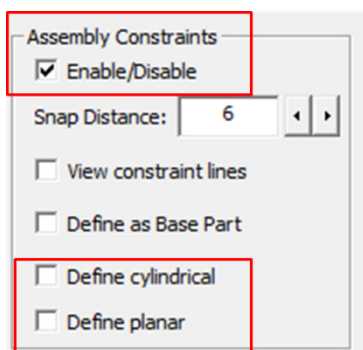


Figura 3.13. Menú para la definición manual de las restricciones de ensamble.

Al activar la casilla de definición de restricciones, plana o cilíndrica, el sistema entra en un modo especial que permite la definición de restricciones utilizando el dispositivo háptico. En este modo el cursor háptico cambia de color y cuando se encuentra cerca de un objeto se pega a él como si la superficie del modelo fuera un imán que atrae al cursor. Cuando el cursor háptico está en contacto con un objeto se deben seleccionar cuatro puntos en la superficie del modelo, los cuales son utilizados para definir la restricción, ya sea cilíndrica o plana. Una vez que una restricción ha sido definida, ésta se guarda como un archivo de texto que contiene su información y que puede posteriormente ser utilizado cada vez que la pieza es cargada en la plataforma, evitando definir nuevamente la misma restricción.

#### 3.3.2.2. Definición de restricción planar

Una restricción planar se define al seleccionar cuatro puntos en la superficie plana de un modelo. Estos cuatro puntos definen un plano de cuatro vértices sobre el cual se aplicará la restricción, Figura 3.14. Las coordenadas de cada uno de estos puntos son almacenadas en el contenedor de restricciones planares de la variable *vtkActor*, y son utilizados en cada ciclo de

simulación para verificar si esta restricción debe ser activada. Estas coordenadas están expresadas en términos de la posición relativa de cada punto con respecto a la posición del objeto.

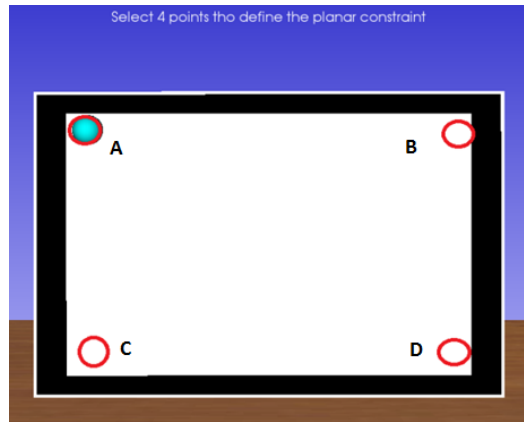


Figura 3.14. Selección de puntos para definición de una restricción planar.

### 3.3.2.3. Definición de restricción cilíndrica

Para definir una restricción cilíndrica manualmente en el sistema HAMS, es necesario seleccionar cuatro puntos del objeto en el cual se creará la restricción. Los tres primeros puntos,  $A$ ,  $B$  y  $C$  son seleccionados en el borde de la forma cilíndrica de la pieza, Figura 3.15. Mediante estos tres puntos es posible calcular el centro del arco que describen, el cual coincide con la ubicación del eje de la forma cilíndrica y que por lo tanto será la ubicación de la restricción respecto al centro geométrico de la pieza.

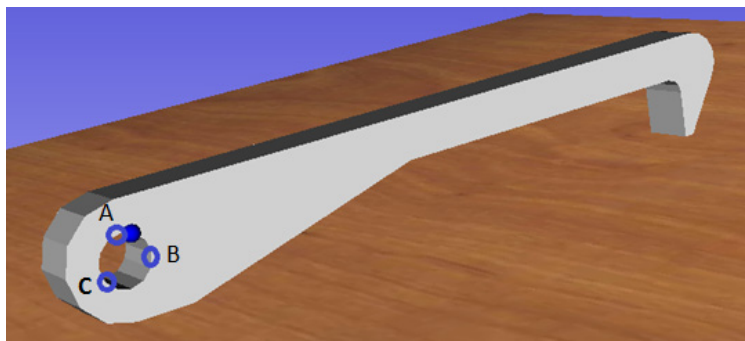


Figura 3.15. Selección de los primeros tres puntos para crear una restricción cilíndrica.

El cuarto punto,  $E$ , se selecciona en el borde opuesto a la forma cilíndrica donde se seleccionaron los primeros tres puntos. Este cuarto punto genera información acerca de la dirección y profundidad que tiene la forma cilíndrica que será restringida, Figura 3.16.

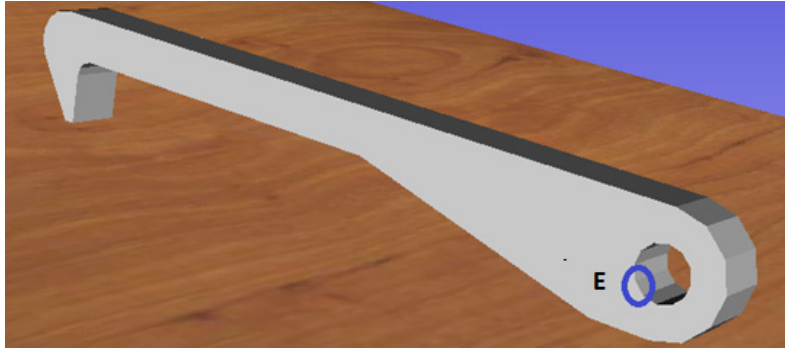


Figura 3.16. Selección del cuarto punto para la definición de la restricción cilíndrica.

Una vez definidos los cuatro puntos de la restricción, es posible determinar el centro del arco que forman dichos puntos mediante la siguiente expresión:

$$\begin{aligned} X_c &= X_a + (E_x * AB_i) + (E_y * CD_i) \\ Y_c &= Y_a + (E_x * AB_j) + (E_y * CD_j) \\ Z_c &= Z_a + (E_x * AB_k) + (E_y * CD_k) \end{aligned} \quad (3.1)$$

donde:

|              |  |
|--------------|--|
| $X_c$        | Posición del centro del arco sobre eje $X$   |
| $Y_c$        | Posición del centro del arco sobre eje $Y$   |
| $Z_c$        | Posición del centro del arco sobre eje $Z$   |
| $X_a$        | Posición del centro geométrico del objeto en $X$                                   |
| $Y_a$        | Posición del centro geométrico del objeto en $Y$                                   |
| $Z_a$        | Posición del centro geométrico del objeto en $Z$                                   |
| $E_x$        | Coordenada $X$ del centro del círculo proyectado sobre plano $XY$                  |
| $E_y$        | Coordenada $Y$ del centro del círculo proyectado sobre plano $XY$ sobre el eje $Y$ |
| $AB_{i,j,k}$ | Cosenos directores del vector $AB$   |
| $CD_{i,j,k}$ | Cosenos directores del vector $CD$   |

El punto  $D$  es la proyección normal del punto  $C$  sobre el vector  $AB$

Una vez encontrado el centro de la forma circular, se calcula la orientación y la profundidad de la forma cilíndrica. Para encontrar la orientación se analizan las distancias entre los puntos  $A$  y  $B$  en cada eje coordenado. Si la diferencia entre alguna de las coordenadas tiende a ser cero entonces la orientación de la forma cilíndrica está sobre ese eje. A continuación se asigna el valor de la profundidad como la diferencia entre la coordenada del punto  $E$  en dicho eje menos la coordenada del punto  $C$  sobre ese mismo eje. En la Figura 3.17 se describe este proceso.

Finalmente, los datos obtenidos son agregados a la variable *vtkActor* para definir la restricción y tenerlos disponibles en cada ciclo de simulación. Estos datos también son guardados en un archivo de texto para su uso posterior cuando el mismo modelo vuelva a ser cargado en el sistema HAMS.

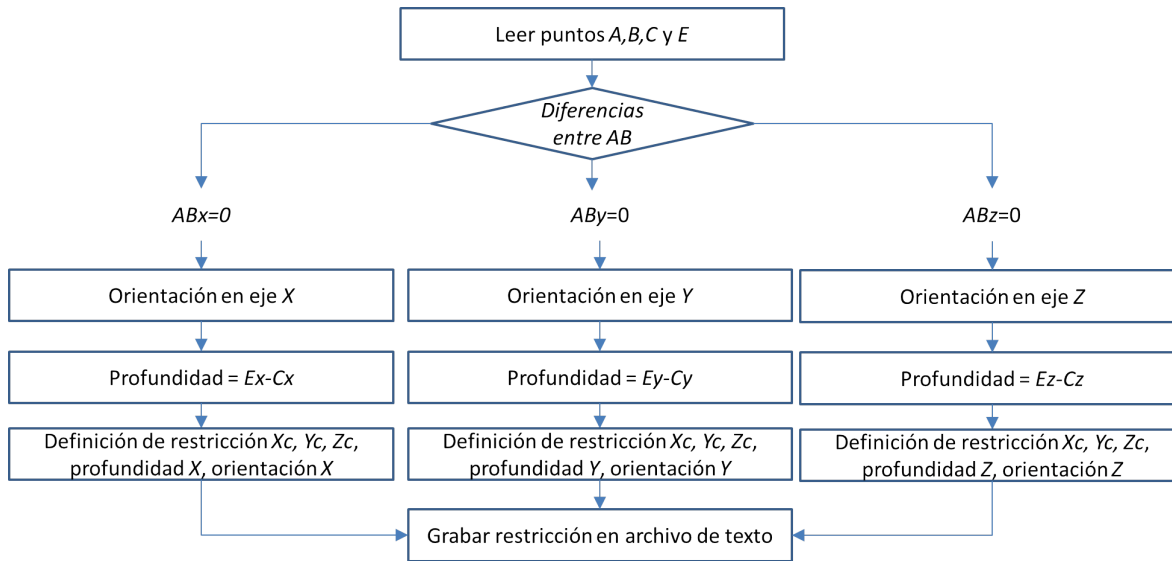


Figura 3.17. Determinación de la orientación y profundidad para la restricción cilíndrica.

#### 3.3.2.4. Creación automática de restricciones cilíndricas

Con el fin de optimizar el proceso de definición de restricciones de ensamble, se ha implementado en el sistema HAMS un algoritmo de reconocimiento automático de formas cilíndricas. El algoritmo se basa en la metodología de identificación de bordes circulares propuesto por Qu y Stucker, (2005) [151], el cual permite identificar las formas cilíndricas que pueden existir en un modelo descrito por una malla triangular. Las formas cilíndricas tiene dos características principales: forman un borde cerrado, y sus dimensiones respecto a dos ejes coordenados son iguales mientras que la dimensión restante tiende a ser cero. El algoritmo implementado en HAMS para identificar formas cilíndricas puede ser descrito de acuerdo a los siguientes pasos:

1. Extraer todas las líneas de borde de la pieza. VTK cuenta con funciones definidas que permiten extraer las líneas de borde dentro de un modelo definido por una malla triangular.
2. Leer los índices de cada línea de borde. Una línea de borde está compuesta por un índice inicial  $A$  y un índice final  $B$ . El borde de una superficie puede estar formado por varias líneas de borde.
3. Determinar si un borde es cerrado. Si el índice inicial de la primera línea de borde analizada es el mismo que el índice final de la última línea de borde analizada entonces esto indica que existe un borde cerrado. En la Figura 3.18 se aprecia que la primera línea contiene los índices 0 y 1 mientras que la última línea contiene los índices 11 y 0, lo que indica que es un borde cerrado, la primera característica de una forma cilíndrica.
4. Depurar los bordes. Un borde cerrado que está compuesto por menos de cinco líneas no se considera un círculo y es descartado del análisis pues corresponde a una forma cuadrada o triangular.

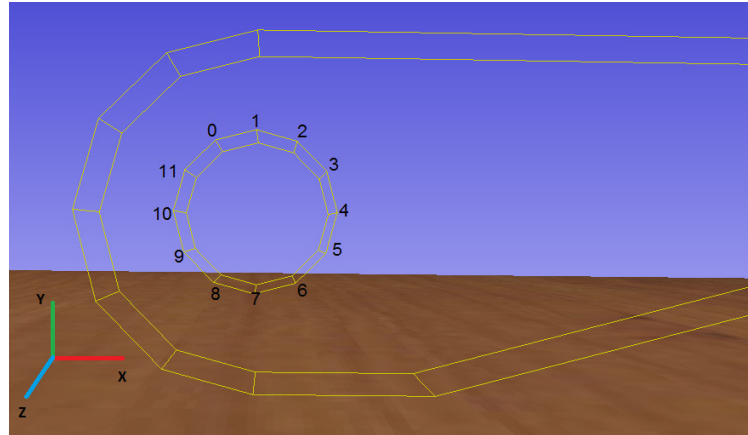


Figura 3.18. Índices de las líneas de borde de forma circular.

5. Analizar las dimensiones de cada índice. En este proceso se identifican las coordenadas máximas y mínimas del borde en cada eje coordenado. Por ejemplo, en la Figura 3.18 el índice 4 tiene la coordenada máxima en el eje X mientras que el índice 10 tiene la coordenada mínima en ese mismo eje. De la misma manera, el índice 7 tiene la coordenada mínima sobre el eje Y mientras que el índice 1 tiene la coordenada máxima. Para este ejemplo las coordenadas sobre el eje Z de todos los puntos son iguales.
6. Con las coordenadas máximas y mínimas se obtiene una dimensión para cada eje coordenado  $D_x$ ,  $D_y$  y  $D_z$ :

$$\begin{aligned}
 D_x &= X_{m\acute{a}xima} - X_{m\acute{i}nima} \\
 D_y &= Y_{m\acute{a}xima} - Y_{m\acute{i}nima} \\
 D_z &= Z_{m\acute{a}xima} - Z_{m\acute{i}nima}
 \end{aligned}
 \tag{3.2}$$

7. Si las dimensiones en dos de los ejes coordenados son iguales, es decir,  $D_x = D_y$ ,  $D_x = D_z$  ó  $D_y = D_z$ , dentro de cierta tolerancia, y la dimensión restante tiende a ser cero, entonces se concluye que los bordes analizados pertenecen a una forma circular con diámetro igual a las dimensiones calculadas. Para el caso del ejemplo de la Figura 3.18  $D_x = D_y$  y la dimensión  $D_z$  es cero.
8. Determinación de la ubicación. Con las dimensiones obtenidas es posible calcular el diámetro  $\Phi$  y las coordenadas del centro  $C$  de la forma circular:

$$\begin{aligned}
 C_x &= X_{m\acute{i}nima} + \frac{D_x}{2} \\
 C_y &= Y_{m\acute{i}nima} + \frac{D_y}{2} \\
 C_z &= Z_{m\acute{i}nima} + \frac{D_z}{2}
 \end{aligned}
 \tag{3.3}$$

9. Determinación de la orientación. La orientación de la forma circular está en función de las dimensiones calculadas: si  $D_x=0$ , entonces la forma se encuentra en el plano  $YZ$ ; si  $D_y=0$  entonces la forma se encuentra en el plano  $XZ$ ; y finalmente si  $D_z=0$  entonces la forma se encuentra en el plano  $XY$ .

Este análisis se realiza para cada uno de los bordes encontrados en el modelo con el fin de encontrar la mayoría de las formas circulares. Sin embargo, es necesario completar el análisis para encontrar todos los datos de las formas cilíndricas, como por ejemplo su profundidad. Para lograr esto, se lleva a cabo un segundo análisis, donde se comparan cada una de las formas circulares identificadas en la etapa anterior con el fin de encontrar si dos círculos tienen la misma posición, orientación y tamaño. Si esto sucede, entonces se ha encontrado una forma cilíndrica, y su profundidad está dada por la distancia entre las dos formas circulares analizadas.

Finalmente, la posición, orientación y profundidad de la forma cilíndrica encontrada son utilizadas para definir una restricción de ensamblaje cilíndrica en el modelo analizado. Este proceso se repite para encontrar todas las formas cilíndricas que contiene el modelo y así definir todas las restricciones cilíndricas de la pieza. Los datos de las restricciones son enviados a la variable *vtk.Actor* para su uso durante el ciclo de simulación.

Cabe hacer mención que éste algoritmo tiene ciertas limitaciones. Una de ellas es la incapacidad para identificar formas cilíndricas que no tengan un borde cerrado o que no estén alineadas con alguno de los ejes coordenados. Sin embargo, este algoritmo ha permitido identificar la mayoría de las formas cilíndricas en diversos modelos sometidos a este método, Figura 3.19. Los resultados de las formas cilíndricas detectadas se muestran en la Tabla 3.2.

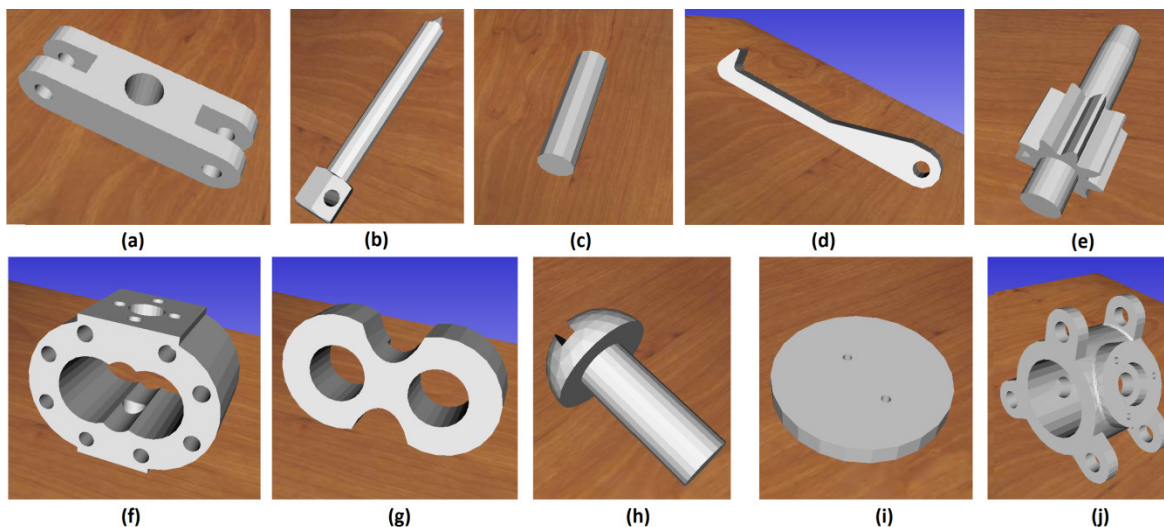


Figura 3.19. Modelos utilizados en el algoritmo de reconocimiento de formas cilíndricas: (a) base de un extractor de poleas, (b) tornillo del extractor, (c) perno del extractor, (d) brazo del extractor, (e) engrane de bomba, (f) carcasa de bomba, (g) cojinete, (h) t

Tabla 3.2. Resultados del algoritmo de identificación automática de formas cilíndricas.

| Parte                        | Originales | Detectadas |
|------------------------------|------------|------------|
| Base del extractor de poleas | 5          | 5          |
| Tornillo del extractor       | 2          | 2          |
| Perno del extractor          | 1          | 1          |
| Brazo del extractor          | 1          | 1          |
| Engrane de bomba             | 1          | 0          |
| Carcasa de bomba             | 26         | 24         |
| Cojinetes                    | 2          | 0          |
| Tornillo                     | 1          | 1          |
| Mariposa válvula             | 3          | 3          |
| Carcasa de válvula           | 8          | 5          |

### 3.3.3 Aplicación de restricciones de ensamble

Cuando las restricciones de ensamble cilíndricas o planares han sido definidas para cada uno de los objetos del ambiente virtual, ya sea manual o automáticamente, y además se ha definido una pieza base sobre la cual se ensamblarán el resto de los objetos, es posible utilizar estas restricciones para limitar el movimiento de la pieza manipulada y así facilitar el proceso de ensamble.

Para activar una restricción de ensamble, el sistema verifica en cada ciclo de simulación si alguna de las restricciones de la pieza manipulada está cerca de alguna de las restricciones de la pieza base dentro de una tolerancia predefinida. En caso de que no exista cercanía entre ninguna restricción, la pieza manipulada puede moverse en cualquiera de sus seis grados de libertad obedeciendo únicamente el comportamiento basado en la física real y la detección de colisiones. Por el contrario, si alguna restricción de la pieza manipulada se encuentra lo suficientemente cerca de alguna restricción del mismo tipo en la pieza base entonces la restricción de ensamble se activa, el objeto manipulado es orientando de acuerdo a la orientación de la pieza base y se reducen sus grados de libertad dependiendo del tipo de restricción. Cada tipo de restricción tiene métodos de aplicación y operación diferentes.

#### 3.3.3.1. Aplicación de restricciones cilíndricas

Las restricciones cilíndricas se aplican a objetos o formas cilíndricas, las cuales tienen un eje central o eje de rotación, Figura 3.20. La restricción cilíndrica coincide con este eje de rotación, además, un objeto puede tener tantas restricciones como formas cilíndricas tenga; por ejemplo, la base del extractor mostrada en la Figura 3.19a contiene cinco formas cilíndricas, por lo tanto podrá tener hasta cinco restricciones de este tipo.

La aplicación de la restricción cilíndrica consiste en limitar la movilidad de la pieza manipulada en el plano perpendicular al eje de rotación, o eje de la restricción, con el fin de solo permitir su movimiento a lo largo de este eje. Además, los grados de libertad de rotación se reducen para solo permitir el giro con respecto al eje de restricción.

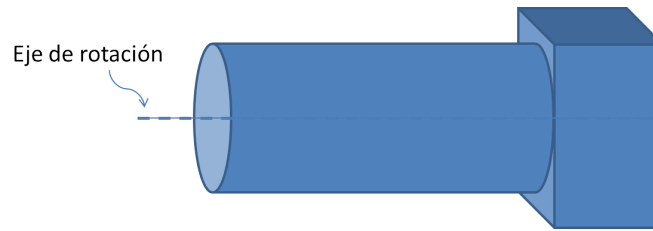


Figura 3.20. Eje de rotación de una forma cilíndrica.

El procedimiento para aplicar este tipo de restricción consiste en tres pasos principales:

1. Verificar que la restricción cilíndrica de la pieza manipulada y la pieza base coincidan, lo cual indica que el objeto manipulado está cerca de su posición de ensamble.
2. Cuando las restricciones coinciden, el objeto manipulado es orientado con respecto a la orientación de la forma cilíndrica de la pieza base, la cual es similar a la orientación que deberá tener la pieza una vez ensamblada.
3. Una vez que la pieza manipulada se ha orientado, los desplazamientos en las direcciones ortogonales al eje del cilindro son restringidos dejando únicamente la posibilidad de mover la pieza en la misma dirección del eje de rotación, Figura 3.21. Este movimiento está caracterizado por la ecuación paramétrica de la recta:

$$X = Ax + ABx - t \quad (3.4)$$

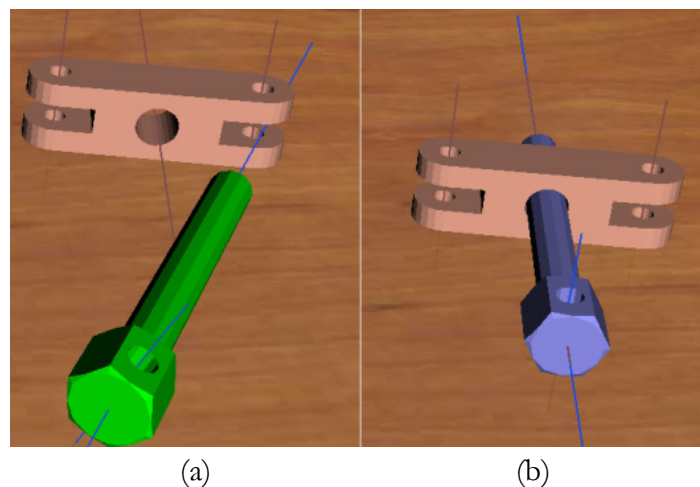


Figura 3.21. Ensamble de partes utilizando restricción cilíndrica: a) restricciones sin aplicar, b) movimiento restringido.

En la Figura 3.22 se describe de manera general el proceso para activar las restricciones de ensamble cilíndricas.



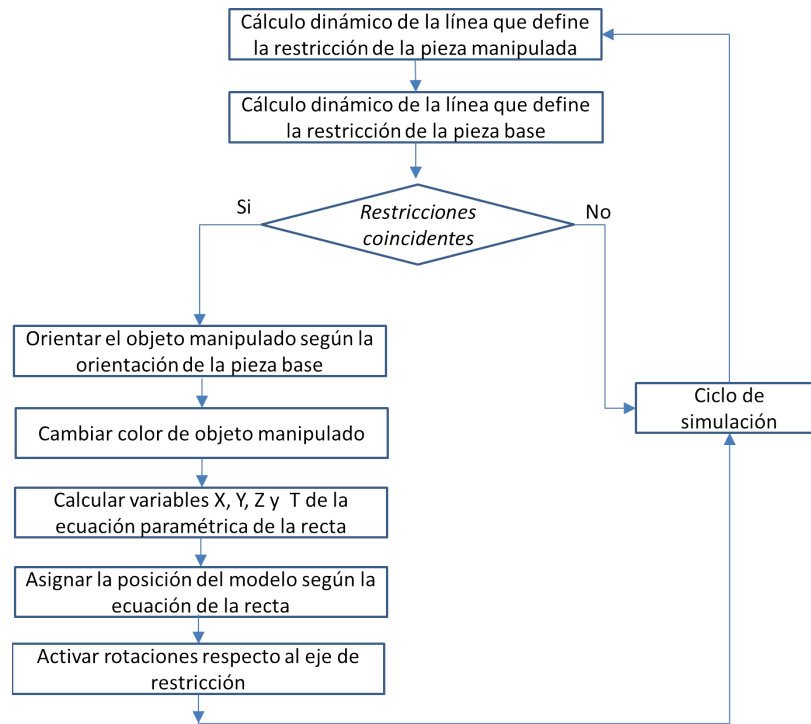


Figura 3.22. Aplicación de restricción de ensamble cilíndrica.

3.3.3.2. Aplicación de restricciones planares

Cuando la pieza base y la pieza manipulada tienen definidas restricciones planares, el programa verifica en cada ciclo de simulación si alguna restricción de la pieza manipulada coincide con alguna restricción de la pieza base de la misma manera que lo hace con las restricciones cilíndricas, es decir, verifica si una superficie restringida de la pieza manipulada está cerca del plano restringido de la pieza base. Cuando un par de planos de restricción se encuentran lo suficientemente cercanos entonces la pieza manipulada toma la orientación del plano de restricción de la pieza base. Los grados de libertad correspondientes a la traslación en los ejes que forma el plano permanecen activos, mientras que el grado de libertad de traslación restante se elimina, permitiendo que la pieza solo se deslice sobre el plano de restricción pero que no penetre o se aleje de dicho plano, Figura 3.23. La rotación respecto al eje normal al plano también es permitida, mientras que las otras rotaciones respecto a los otros dos ejes se eliminan.

En el sistema HAMS las restricciones planares son creadas por medio de la definición de una junta cinemática plana entre la pieza base y la pieza manipulada, la cual cumple con las condiciones de movimiento establecidas por este tipo de restricción. La distancia que puede recorrer la pieza manipulada una vez que está restringida es limitada por las dimensiones de la restricción planar de la pieza base dadas al momento de crear dicha restricción.

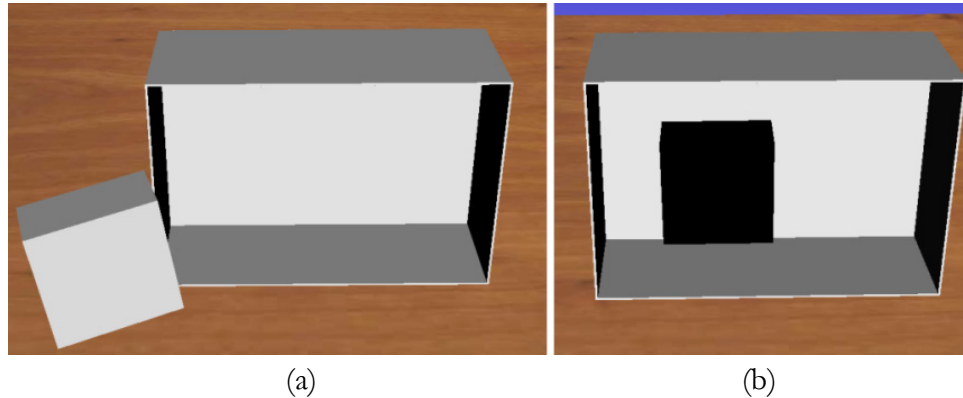


Figura 3.23. Aplicación de restricciones planares: a) restricciones sin aplicar, b) movimiento restringido.

### 3.3.4 Creación de sub-ensambles

Cuando se ha realizado el ensamble de diversas piezas para formar un componente, debe existir alguna manera de sujetar los componentes entre ellos para evitar que el ensamble se destruya al manipularlo. En el caso de ensambles reales es común el uso de tornillos, remaches, soldadura, etc., para realizar esta unión.

En el caso de ensambles virtuales, y especialmente cuando se utiliza el modelado basado en física, es muy difícil la simulación de elementos como tornillos o remaches que puedan sujetar las diversas piezas de un ensamble, pues requiere de una alta demanda de capacidad de cómputo para manejar de manera correcta todas las colisiones entre todos los objetos. Sin embargo, una alternativa que permite mantener unidos las piezas de un componente es el uso de juntas cinemáticas. Las juntas cinemáticas están definidas dentro del simulador físico y se pueden crear para unir un par de objetos, por ejemplo la pieza base y la pieza manipulada.

Las juntas cinemáticas permiten reducir los grados de libertad de un objeto virtual para producir un movimiento deseado respecto al objeto al que se une, incluso se pueden reducir todos los grados de libertad con el fin de eliminar completamente el movimiento relativo entre los objetos unidos. De este modo, las juntas cinemáticas son utilizadas para fijar el movimiento de la pieza ensamblada respecto a la pieza base, es decir, cuando una pieza ha sido ensamblada en la pieza base se crea una junta cinemática entre las dos para eliminar su movimiento relativo, de esta manera cuando la pieza base sea manipulada, todas las piezas que están ligadas a la pieza base tendrán el mismo movimiento, dando así la impresión de que todas las piezas están unidas, Figura 3.24. En el sistema HAMS las juntas cinemáticas se crean de manera automática cuando la pieza manipulada es ensamblada en su posición final. Debido a que la junta cinemática une las representaciones físicas de los modelos, las características físicas del ensamble son actualizadas también de manera automática, por lo que no es necesario definir nuevamente la masa o la matriz de inercia para el ensamble.

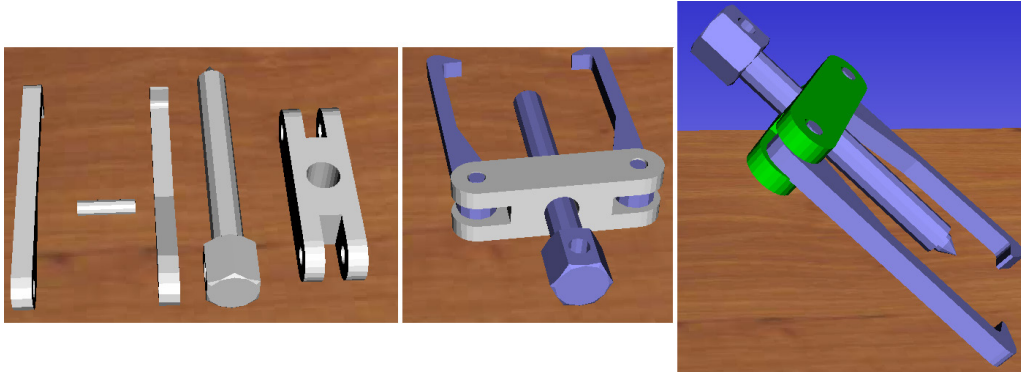


Figura 3.24. Manipulación de sub-ensamble por medio de juntas cinemáticas: a) partes a ensamblar, b) componente ensamblado, c) manipulación de todo el objeto ensamblado.

### 4.1 Introducción

El objetivo del sistema HAMS es ser utilizado como una plataforma para llevar a cabo tareas de ensamble virtual con fines de planeación, evaluación, entrenamiento y generación automática de planes de ensamble dentro de un ambiente intuitivo y con un desempeño satisfactorio. Para lograr lo anterior, es necesario identificar y evaluar la influencia que tienen los diversos componentes del sistema y parámetros de la simulación sobre el desempeño del proceso de ensamble. Al cuantificar la influencia de dichos parámetros se obtendrá una mejor comprensión del funcionamiento de los sistemas de ensamble virtual, permitiendo así mejorar su desempeño. De la misma manera, dichos parámetros podrán ser ajustados de acuerdo a las características específicas de cada caso particular de ensamble, logrando así reducir las limitaciones del sistema.

Los componentes que tienen una mayor influencia en el proceso de ensamble son aquellos que involucran la percepción del usuario respecto a los objetos virtuales y la forma en que se lleva a cabo la interacción entre estos objetos. Por tanto, el simulador físico, la representación física del modelo, la percepción del peso y el uso de restricciones de ensamble son algunos de los elementos más representativos en la simulación del proceso de ensamble. Estos elementos requieren ser analizados para evaluar su influencia en el proceso al llevar a cabo diversos casos de ensamble con diferentes características.

### 4.2 Evaluación de simuladores físicos

La evaluación de los simuladores físicos tiene como objetivo identificar sus fortalezas y debilidades cuando son utilizados en tareas de ensamble virtual que incluyen el uso de dispositivos hápticos. Existen diversos trabajos donde se proponen metodologías para evaluar el desempeño de los simuladores de física, sin embargo, estas metodologías se enfocan en la evaluación general del funcionamiento del simulador y no en aplicaciones específicas, como por ejemplo su uso en sistemas de ensamble virtual. En este trabajo se propone una metodología para evaluar los simuladores de física cuando son utilizados en sistemas de ensamble virtual apoyados por dispositivos hápticos.

La metodología propuesta consiste en siete pruebas de ensamble, cada una con niveles de complejidad variable y diferente propósito. En la Tabla 4.1 se muestra un resumen de estas pruebas junto con sus objetivos y las propiedades a ser evaluadas. Estas pruebas han sido definidas para evaluar de una manera global el desempeño de los simuladores de física en un ambiente de ensamble virtual. Las pruebas uno a la cinco evalúan propiedades individuales de los simuladores de física, como la respuesta a colisiones y la estabilidad bajo condiciones y características de ensamble variables. Las pruebas siete y ocho involucran el ensamble virtual

de modelos complejos que representan partes reales. El propósito de estas últimas pruebas es evaluar el desempeño de los simuladores de física en casos generales de tareas de ensamble.

Tabla 4.1. Pruebas para evaluar el desempeño de simuladores físicos en el sistema HAMS.

| Prueba                             | Objetivo   | Propiedades a evaluar   | Parámetro de evaluación   |
|------------------------------------|--|---|---|
| 1. Caída libre                     | Evaluar el método integrador de cada simulador   | Rapidez de la simulación y estabilidad, influencia de la complejidad del modelo, influencia del ciclo háptico | Tiempo de caída libre   |
| 2. Balanza                         | Evaluar la precisión de la respuesta a colisiones  | Precisión de la respuesta a colisiones, influencia del método de representación                               | Ángulo de inclinación de la balanza   |
| 3. Pila de cajas                   | Evaluar el desempeño de contactos acumulativos en superficies planas                                 | Comportamiento de contactos acumulativos, respuesta a colisiones y estabilidad                                | Tiempo de finalización de la tarea (TFT) y número de cajas apiladas             |
| 4. Empaquetado                     | Evaluar la respuesta a colisiones y la estabilidad de contactos múltiples en diferentes direcciones  | Comportamiento de contacto multidireccional, respuesta a colisiones y estabilidad                             | Tiempo de finalización de la tarea (TFT) y estabilidad                          |
| 5. Percepción del peso             | Evaluar la influencia del peso de los objetos virtuales  | Facilidad de manipulación de los objetos virtuales, influencia del peso                                       | Tiempo de finalización de la tarea (TFT) y facilidad de manipulación de objetos |
| 6. Tamaño de la malla              | Evaluar el desempeño del simulador al utilizar diferentes tamaños de la malla u objeto               | Influencia del tamaño de la malla triangular  | Tiempo de finalización de la tarea (TFT) y tiempo de simulación física          |
| 7. Ensamble de extractor de poleas | Evaluar el desempeño del simulador al llevar a cabo ensambles de objetos reales con formas regulares | Desempeño en aplicaciones reales  | Tiempo de finalización de la tarea (TFT), Fuerzas y tiempo de simulación física |
| 8. Ensamble de prensa de banco     | Evaluar el desempeño del simulador al llevar a cabo ensambles de objetos reales con formas complejas | Desempeño en aplicaciones reales  | Tiempo de finalización de la tarea (TFT), Fuerzas y tiempo de simulación física |

#### 4.2.1 Prueba de caída libre

El objetivo de esta prueba es evaluar el método integrador utilizado por cada simulador físico, el cual involucra el método numérico utilizado para calcular la nueva posición del objeto en cada paso de la simulación. Su desempeño está afectado por diversos factores como son el tiempo del paso de la simulación, la complejidad del modelo virtual, la complejidad de la escena virtual, el número de objetos en la escena, etc. El desempeño del integrador afecta la percepción del usuario sobre el comportamiento dinámico de los objetos en el sistema de ensambles virtuales, un mal desempeño puede crear diferentes efectos como la apariencia de baja gravedad (efecto de gravedad lunar), el traslape entre objetos virtuales, la inestabilidad de los componentes ensamblados e incluso la discontinuidad de la simulación. Un desempeño adecuado del método integrador resultará en una simulación rápida, continua y precisa.

La prueba de caída libre consiste en dejar caer los objetos virtuales desde una altura de 500 unidades y medir el tiempo que tardan en llegar al suelo, Figura 4.1. El tiempo de caída libre refleja qué tan rápido el método integrador resuelve la posición del objeto virtual en cada ciclo de simulación. En esta prueba se evalúan dos condiciones: la influencia de la complejidad de la forma (número de triángulos), y la influencia del ciclo de renderizado háptico en el desempeño del integrador. Cuatro modelos virtuales con diferente complejidad fueron seleccionados para llevar a cabo esta prueba: una caja, un perno, un engrane y la carcasa de la bomba de aceite. Se realizaron cinco repeticiones para cada condición y para cada modelo.

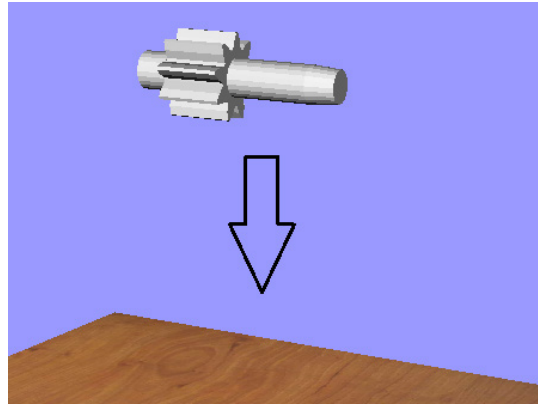


Figura 4.1. Pruebas de caída libre.

#### 4.2.2 Prueba de balanza

El objetivo de esta prueba es evaluar la precisión de la respuesta a colisiones. Esta prueba consiste en dos esferas del mismo tamaño y de la misma masa y una balanza que incluye una caja en cada extremo, Figura 4.2. Al inicio de la prueba la balanza está completamente horizontal y permanece estática. Enseguida, cada esfera es colocada en cada una de las cajas de la balanza, cada caja mantiene sujeta a la esfera para restringir completamente su movimiento. Una vez que las dos esferas son colocadas en las cajas, el estado cinemático de la balanza se cambia a dinámico.

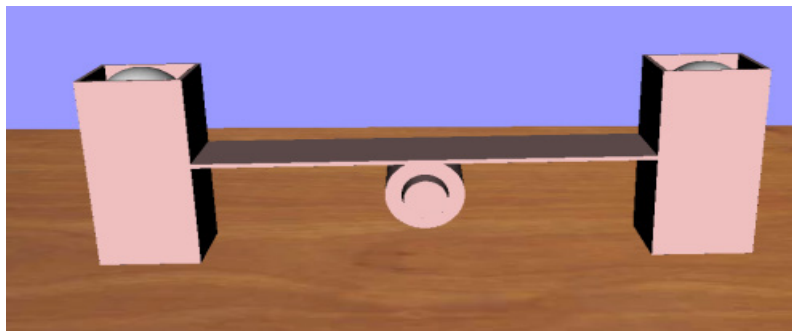


Figura 4.2. Balanza para la evaluación de la precisión de colisiones.

En teoría la balanza deberá permanecer en equilibrio pues las dos esferas son idénticas. Sin embargo, en caso de que la respuesta a colisiones de alguna de las dos esferas sea diferente, la balanza se inclinará hacia uno de los lados, esto significa que la respuesta a colisiones no es precisa. Durante esta prueba se mide el ángulo de inclinación de la balanza a lo largo de 60 segundos; si el ángulo es menor a un grado entonces el sistema se considera balanceado; si el ángulo es mayor a  $1^\circ$  pero menor a  $5^\circ$  entonces las fuerzas de reacción tienden a ser similares pero existe una diferencia mínima, en este caso el resultado es expresado mediante la dirección hacia la cual se inclinó la balanza: “derecha” o “izquierda”. Finalmente, si el ángulo es mayor a  $5^\circ$  el sistema se considera completamente desbalanceado y el resultado es expresado como “derecha+” o “izquierda+” dependiendo de la inclinación de la balanza. Respecto al ensamble virtual, una respuesta de colisiones inadecuada puede resultar en el traslape de objetos virtuales o fuerzas de retroalimentación poco realistas, lo cual afecta el desempeño y resultados de la simulación del ensamble.

Para esta prueba la balanza es creada utilizando una malla triangular y las dos esferas son creadas utilizando los diferentes métodos de representación descritos en el capítulo III (primitivos o por medio de los algoritmos GIMPACT, HACD, o ConvexFT). Cada uno de los diferentes métodos de representación disponibles en cada simulador físico es evaluado mediante esta prueba.

### 4.2.3 Pila de cajas

Esta prueba incluye 15 cajas planas que deben ser apiladas utilizando el dispositivo háptico, Figura 4.3. Durante esta prueba el simulador físico debe resolver la respuesta a colisiones para contactos acumulativos de superficies planas. Un bajo desempeño del simulador de física resultará en una estabilidad pobre de la pila ensamblada, y por lo tanto una mayor dificultad para realizar el ensamble y un mayor tiempo de finalización de la tarea (TFT). Mediante esta prueba se pretende evaluar la habilidad del simulador físico para manejar contactos acumulativos entre superficies planas, lo cual es una condición común en tareas de ensamble.

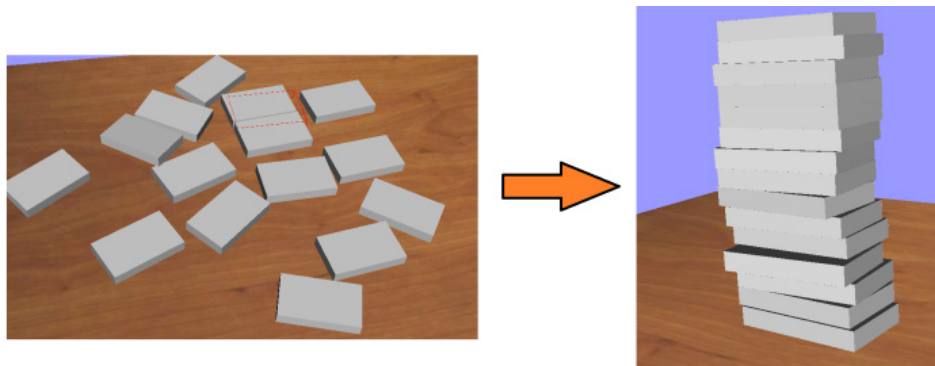


Figura 4.3. Prueba de apilado de cajas.

#### 4.2.4 Prueba de empaquetado

El objetivo de esta prueba es evaluar la respuesta a colisiones y estabilidad del simulador de física cuando ocurren contactos múltiples en diferentes direcciones. Esta prueba consiste en empaquetar ocho cajas en un contenedor utilizando el dispositivo háptico, Figura 4.4. La primera caja debe ser ensamblada en la esquina inferior izquierda del contenedor, la segunda en la esquina inferior derecha, la tercera en la esquina superior izquierda y la cuarta en la esquina superior derecha del contenedor. Enseguida, un segundo nivel de cajas debe ser ensamblado sobre la primera capa siguiendo el mismo orden. Una vez que las ocho cajas han sido ensambladas dentro del contenedor el ensamble se considera como completado.

En esta tarea el simulador físico debe calcular la respuesta a colisiones en diferentes direcciones al mismo tiempo; por ejemplo, las colisiones entre la caja manipulada, el piso del contenedor, el muro izquierdo y el muro frontal. Un bajo desempeño del simulador causará vibración de la parte manipulada, además de traslape entre los objetos virtuales, resultando en una mayor dificultad para realizar el ensamble y como consecuencia un mayor tiempo de finalización de la tarea.

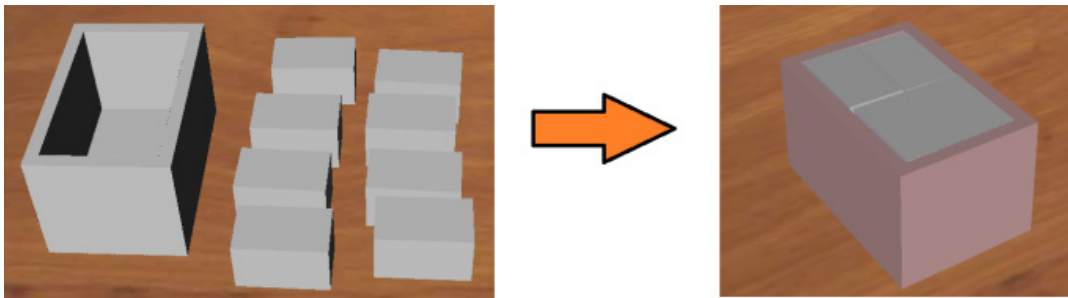


Figura 4.4. Prueba de empaquetado.

#### 4.2.5 Prueba de tamaño de la malla

Los sistemas de ensamble virtual son particularmente útiles en la planeación del ensamble de componentes pequeños o microcomponentes. Sin embargo, los simuladores físicos tienen un tamaño de objeto mínimo para el cual se pueden detectar las colisiones de manera precisa; objetos muy pequeños pueden producir una respuesta a colisiones poco realista, incrementando la dificultad para llevar a cabo el proceso de ensamble virtual. Por tanto, el objetivo de esta prueba es evaluar el desempeño de los simuladores físicos en tareas de ensamble virtuales-hápticos de componentes de tamaño pequeño y muy pequeño. Lo anterior con el fin de identificar el tamaño de objetos mínimo que el simulador puede manejar de manera correcta. En esta prueba se utiliza el caso de ensamble perno-agujero, sobre el cual se definen siete niveles de escala conforme a la Tabla 4.2. Una comparativa gráfica de las primeras cuatro escalas puede ser observada en la Figura 4.5. En esta prueba los parámetros de evaluación son el tiempo de finalización de la tarea y el tiempo de simulación física.



Tabla 4.2. Niveles de escala para la evaluación del tamaño de los objetos.

| Escala       | Dimensiones de la caja con agujero (mm) | Volumen (mm <sup>3</sup> ) | Triángulos |
|--------------|---|----------------------------|------------|
| 1x (regular) | 65 x 65 x 40                            | 169000                     | 160        |
| 0.1x         | 6.5 x 6.5 x 4.0                         | 169                        | 160        |
| 0.01x        | 0.65 x 0.65 x 0.4                       | 0.169                      | 160        |
| 0.001x       | 0.065 x 0.065 x 0.04                    | 1.69 x 10 <sup>-4</sup>    | 160        |
| 0.0001x      | 0.0065 x 0.0065 x 0.004                 | 1.69 x 10 <sup>-7</sup>    | 160        |
| 0.00001x     | 0.00065 x 0.00065 x 0.0004              | 1.69 x 10 <sup>-10</sup>   | 160        |
| 0.000001x    | 0.000065 x 0.000065 x 0.00004           | 1.69 x 10 <sup>-13</sup>   | 160        |

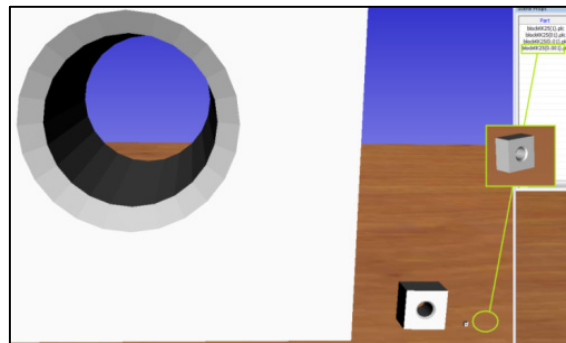


Figura 4.5. Escala de ensamble perno-agujero: 1x, 0.1x, 0.01x y 0.001x.

#### 4.2.6 Ensamble de extractor de poleas

El objetivo de esta prueba es evaluar el desempeño general del simulador físico cuando se lleva a cabo un ensamble virtual de modelos que representan objetos reales y utilizando el dispositivo háptico. Para esta prueba se ha seleccionado el ensamble de un extractor de poleas, el cual consiste de seis partes con formas cilíndricas: la base del extractor, el tornillo, dos brazos y dos pernos pequeños, Figura 4.6a.

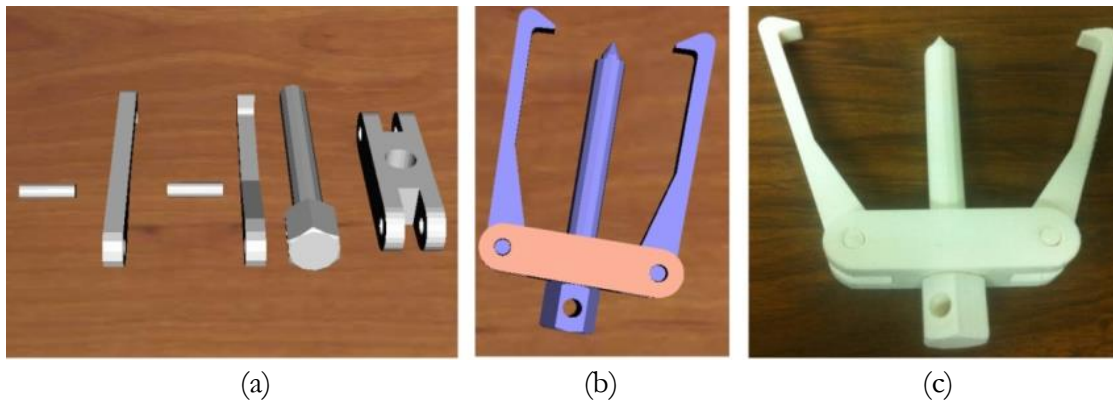


Figura 4.6. Ensamble del extractor de poleas: (a) piezas sin ensamblar; (b) ensamble virtual; (c) ensamble real.

El proceso de ensamble consiste en ubicar la base del extractor en una posición donde se puedan ensamblar el resto de los objetos sobre ella. Una vez posicionada, esta pieza permanece estática y el siguiente paso es ensamblar el tornillo sobre la base, seguido del brazo derecho y a continuación el perno que sujeta el brazo a la base. El procedimiento se repite para el brazo izquierdo y su perno correspondiente, Figura 4.6b. La dificultad de este ensamble consiste en que el tornillo representa un objeto con longitud mucho mayor que su sección transversal. Este tipo de objetos son difíciles de manejar por el sistema de detección de colisiones debido a la alta aceleración que se presenta en sus extremos. Por otro lado, al ensamblar los pernos que sujetan a los brazos, se debe buscar que los agujeros de la base y del brazo estén alineados para poder realizar el ensamble, tal y como sucede en el mundo real. Estas condiciones ponen a prueba el desempeño del sistema de detección de colisiones dentro de cada simulador de física. Para obtener datos cuantitativos que permitan evaluar el desempeño del sistema, en la simulación se mide la fuerza de retroalimentación, el tiempo de simulación física y el tiempo de finalización de la tarea.

Con fines comparativos también se han fabricado representaciones reales del modelo del extractor de poleas. Estas representaciones han sido creadas mediante técnicas de prototipado rápido utilizando el mismo archivo STL del modelo virtual, garantizando condiciones similares a las del ambiente virtual, Figura 4.6c.

#### **4.2.7 Ensamble de la prensa de banco**

El objetivo de esta prueba es evaluar el desempeño de cada simulador físico cuando se llevan a cabo ensambles virtuales de objetos que representan partes reales con formas especiales o complejas. El modelo seleccionado corresponde a una prensa de banco que consta de cuatro partes: una mordaza corta, una mordaza larga, un tornillo y un pasador. Figura 4.7a. Para llevar a cabo este ensamble la mordaza larga es definida como la pieza base y permanece estática durante todo el proceso. A continuación la mordaza corta se desliza sobre el perfil I de la mordaza larga para ensamblar estas dos piezas. Una vez ensambladas, el tornillo debe ser insertado en la cavidad que poseen las dos mordazas. Finalmente, el pasador es ensamblado en el agujero del tornillo, con lo cual se completa el proceso de ensamble, Figura 4.7b. Como se puede observar, la parte inferior de la mordaza larga posee una corredera en forma de I, sobre la cual se debe deslizar la mordaza corta para realizar el ensamble. Esta característica impone condiciones de simulación y ensamble diferentes al ensamble de piezas con forma cilíndrica, pues en este caso se puede presentar contactos múltiples en varias direcciones a la vez. En caso de que el simulador físico no sea capaz de manejar de manera efectiva este tipo de contactos la respuesta a colisiones será excesiva o con vibraciones, dificultando o incluso imposibilitando la realización del ensamble.

Al igual que en el caso de ensamble anterior, la fuerza de retroalimentación, el tiempo de simulación física y el tiempo de finalización de la tarea son los parámetros cuantitativos que se miden para evaluar el desempeño del simulador. Con el propósito de realizar un análisis comparativo entre el ensamble virtual y el ensamble real, se fabricó un prototipo real utilizando técnicas de prototipado rápido, Figura 4.7c.

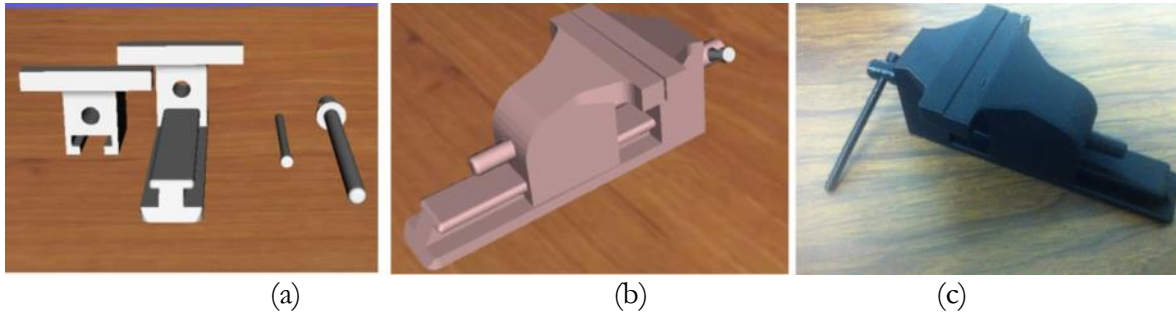


Figura 4.7. Ensamble de la prensa de banco: (a) piezas sin ensamblar, (b) ensamble virtual, (c) ensamble real.

### 4.3 Análisis y evaluación de modos de representación física

La detección de colisiones es un aspecto clave en las tareas de ensamble virtual, y su desempeño está directamente ligado a la forma en que los objetos virtuales son creados y manejados dentro del simulador físico. La representación física del objeto virtual tiene como función principal la detección y respuesta a colisiones además del comportamiento dinámico. Existen diversos métodos o algoritmos para crear la representación física. Los simuladores utilizados en el sistema HAMS – Bullet y PhysX – soportan los algoritmos de representación GIMPACT, ACD, HACD y ConvexFT, descritos en el Capítulo III. El desempeño de cada uno de estos métodos en tareas de ensamble háptico es diverso y por tanto debe ser evaluado.

Los modelos CAD utilizan información analítica y topológica para crear una representación de alta precisión del objeto. Sin embargo, esta información es difícil de manejar en aplicaciones que requieren detección de colisiones de objetos dinámicos en tiempo real. Por lo tanto, en ambientes virtuales es común el uso de representaciones teseladas de los modelos CAD, las cuales consisten en representar la superficie de un modelo sólido por medio de un conjunto de polígonos, conocido como malla triangular. Para casos de ensamble virtual, la malla triangular o representación física del objeto debe ser lo más precisa posible, pues una representación con baja precisión geométrica puede resultar en una simulación incorrecta del proceso de ensamble virtual.

Con el fin de identificar las fortalezas y debilidades de cada uno de los algoritmos de representación se han propuesto dos etapas de evaluación: 1) evaluación de precisión geométrica y 2) evaluación del desempeño. El algoritmo que no supere la primera etapa no será evaluado en la segunda.

#### 4.3.1 Evaluación de precisión geométrica

En esta primera etapa se evalúa de manera visual la precisión geométrica de la representación física obtenida por medio de cada uno de los métodos de representación. La precisión geométrica de la representación física de un objeto es evaluada mediante una inspección visual comparando el modelo original con el modelo obtenido al aplicar el método de representación física - GIMPACT, ACD, HACD o ConvexFT -. La inspección visual consiste en verificar si

todas las características geométricas del modelo original tales como agujeros, esquinas, bordes y concavidades son reproducidas de manera fiel.

Varios modelos con características geométricas diferentes fueron seleccionados para llevar a cabo esta evaluación, Figura 4.8. Cada uno de estos modelos fue cargado en el sistema de ensamblajes virtuales y se creó su representación física utilizando cada uno de los métodos de representación disponibles en cada simulador de física. La aplicación *PhysX Visual Debugger* (PVD) [152] permite observar gráficamente la escena del módulo físico, haciendo posible observar la geometría de la representación física creada en PhysX. Por otro lado, MeshLab [153] es la aplicación utilizada para visualizar la representación física creada en Bullet. De este modo es posible comparar la representación física con el modelo original, y así establecer cual algoritmo de representación produce los mejores resultados en términos de precisión geométrica.

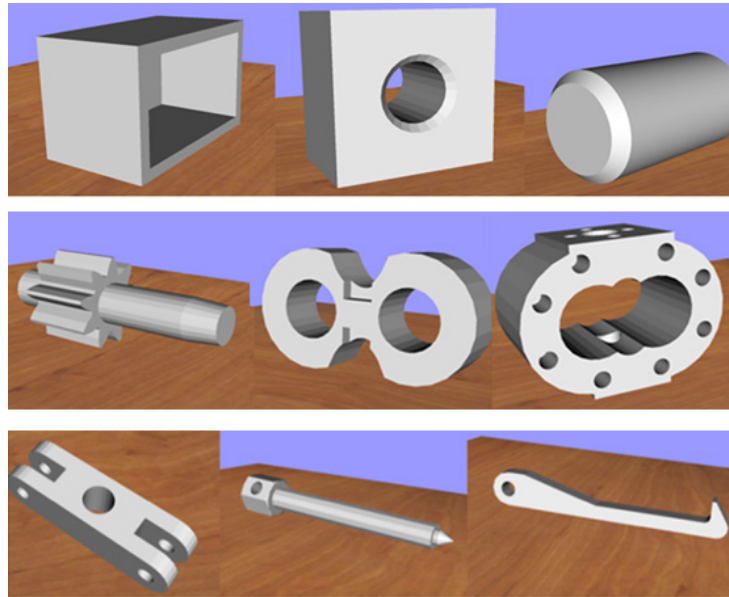


Figura 4.8. Modelos utilizados para realizar la evaluación de precisión geométrica.

### 4.3.2 Evaluación del desempeño

En esta etapa se analiza la influencia de cada algoritmo de representación en el desempeño del sistema y en el proceso de ensamble virtual. Esta evaluación se realiza mediante un conjunto de pruebas para las cuales se realiza una tarea de ensamble considerando los diferentes métodos de representación disponibles. Las pruebas consideradas se detallan a continuación.

#### 4.3.2.1 Influencia sobre el método integrador

Para evaluar la influencia que tiene el método de representación física sobre el método integrador, se realizan pruebas de caída libre en donde se deja caer un objeto desde una altura de 500 unidades y se mide el tiempo que tarda en alcanzar el suelo. En estas pruebas se utilizan tres objetos: una caja, un perno y un engrane. A diferencia de las pruebas realizadas en la

evaluación de los simuladores, ahora el método de representación física se considera como el factor a evaluar de la siguiente manera:

1. La caja es representada mediante primitivos y mediante los métodos de representación GIMPACT y ConvexFT.
2. El perno es representado por los métodos GIMPACT y ConvexFT.
3. El engrane es representado por los métodos GIMPACT, ConvexFT, HACD y ACD.

Para cada modelo y cada método de representación se realizaron cinco repeticiones y se registró el tiempo en alcanzar el suelo para cada una de ellas.

#### *4.3.2.2 Precisión de respuesta a colisiones*

Con el fin de evaluar la influencia del método de representación física en el cálculo de las fuerzas de reacción, se propone utilizar el experimento de la balanza presentado en la sección 4.2.2 de este documento, con la diferencia de que ahora se compararán los diferentes métodos de representación entre sí, es decir, cada lado de la balanza tendrá una esfera creada con un método de representación diferente. En este caso se realizan cinco repeticiones para cada combinación de métodos de representación física, midiendo el ángulo de inclinación de la balanza durante 60 segundos y expresando el resultado como “derecha” o “izquierda”. Con el fin de validar los resultados y el correcto funcionamiento de la balanza, la posición de las esferas es intercambiada de izquierda a derecha.

#### *4.3.2.3 Tolerancia y precisión de respuesta a colisiones*

El objetivo de la prueba es evaluar la capacidad en el manejo de tolerancias de cada uno de los métodos de representación. La prueba consiste en empaquetar doce cajas sólidas en dos niveles dentro de un contenedor de pared delgada, Figura 4.9. El contenedor tiene dimensiones internas de 200mm x 300mm x 100mm, de ancho, largo y altura respectivamente. Además, se utilizan tres tamaños diferentes de cajas: 99.5 x 99.5mm, 99.8 x 99.8mm y 100 x 100mm, todas con una altura de 50mm, lo cual lleva a obtener tolerancias de ensamble de 1.5mm, 1.2mm y 0mm, cuando tres cajas están alineadas dentro del contenedor. En esta prueba se mide el tiempo de finalización de la tarea y el tiempo de simulación física y háptica, para cada uno de los métodos de representación disponibles y para cada nivel de tolerancia.

#### *4.3.2.4 Desempeño en una tarea de ensamble háptico*

El objetivo de estas pruebas es evaluar el comportamiento de los diferentes métodos de representación del modelo físico al realizar tareas de ensamble virtual háptico. Para llevar a cabo esta evaluación se utilizan casos de ensamble que corresponden a objetos reales. El tiempo TFT es una medida del desempeño general de la simulación del ensamble. Un mal desempeño del sistema ocasionará dificultad para manipular y controlar los objetos virtuales. Por ejemplo, una mala representación geométrica o una detección de colisiones poco efectiva ocasionaran que los objetos se traslapen, se muevan con lentitud o se produzcan fuerzas de reacción excesiva, lo cual resultará en un tiempo de finalización de la tarea grande debido a las dificultades para llevar a cabo el ensamble.

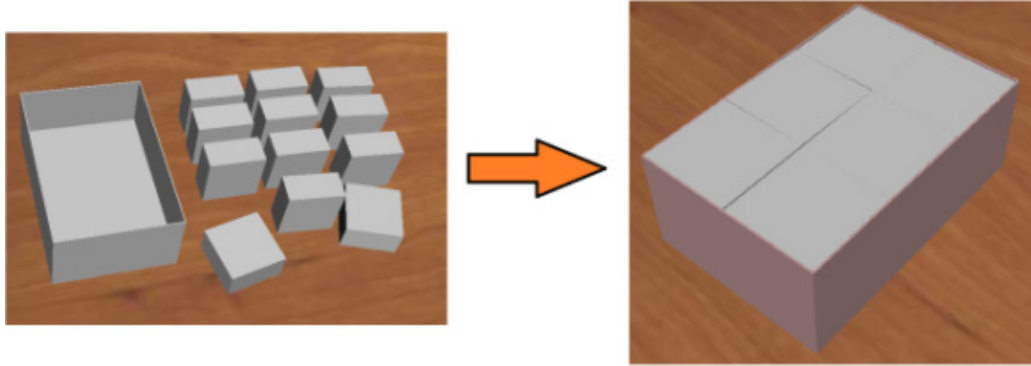


Figura 4.9. Empaquetado de cajas para la evaluación del método de representación.

También se mide la fuerza de retroalimentación háptica, la cual es la fuerza calculada por el simulador físico. La magnitud de la fuerza de retroalimentación puede ofrecer datos referentes a la estabilidad y realismo de la aplicación, pues fuerzas discontinuas, muy grandes o muy pequeñas indican un bajo desempeño de la simulación.

Finalmente se mide el tiempo de simulación física, el cual indica el tiempo que tarda el simulador en calcular las nuevas posiciones de los objetos virtuales en cada paso de la simulación. Un tiempo de simulación alto indicará un pobre desempeño del simulador físico, causando un bajo desempeño general del sistema.

Las pruebas se llevan a cabo utilizando cada uno de los modos de representación disponibles en los simuladores físicos integrados en el sistema HAMS, los cuales se resumen en la Tabla 4.3. Además se ha incluido el método de representación mixta, el cual se describe a continuación.

Tabla 4.3. Métodos de representación física disponibles en cada simulador de física.

| Representación       | Bullet         | PhysX v2.x                       | PhysX v3.x                       |
|----------------------|----------------|----------------------------------|----------------------------------|
| Por malla triangular | GIMPACT        | ConvexFT                         | ConvexFT                         |
| Por descomposición   | HACD           | HACD                             | HACD                             |
| Mixta                | GIMPACT & HACD | Malla triangular estática & HACD | Malla triangular estática & HACD |

*Método de representación mixta*

Con el fin de obtener un mejor desempeño del sistema y del proceso de ensamble virtual se propone un nuevo método para la representación física de los objetos virtuales, el cual consiste en mezclar diferentes modos de representación en un mismo escenario de ensamble para aprovechar lo mejor de cada método, a este planteamiento se le ha llamado método de representación mixta.

El método de representación mixta consiste en utilizar la representación por descomposición en elementos convexos (HACD) junto con la representación por mallas triangulares en el mismo objeto, donde cada representación es intercambiada dependiendo de su estado. Este método funciona de la siguiente manera: la representación física de los objetos que están siendo manipulados por el usuario es creada por medio de la descomposición en elementos convexos, obteniendo así una respuesta a colisiones rápida; para el caso del sistema HAMS se utiliza el algoritmo HACD. Para para la pieza base u objetos estáticos que ya han sido ensamblados y que requieren de una alta precisión geométrica se utiliza el modo de representación por mallas triangulares (GIMPACT o malla triangular estática). Durante el proceso de ensamble un objeto puede ser representado por cualquiera de los dos métodos dependiendo de su estado actual, es decir, si está siendo manipulado o si debe permanecer estático, Figura 4.10.

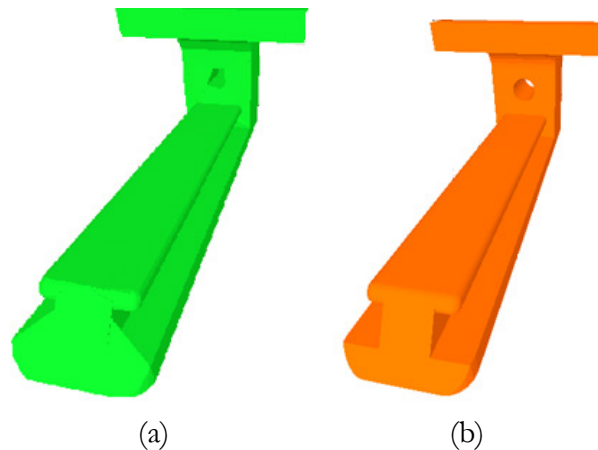


Figura 4.10. Método de representación mixta: (a) objeto dinámico, (b) objeto estático ensamblado.

Para realizar la evaluación del desempeño de los métodos de representación en el proceso de ensamble virtual háptico se seleccionaron cuatro casos de ensamble: extractor de poleas, prensa de banco, bomba de engranes, y cilindro neumático, los cuales utilizan modelos que representan objetos reales. Los ensambles del extractor de poleas y el ensamble de la prensa de banco corresponden a los descritos en las secciones anteriores 4.2.6 y 4.2.7 de este capítulo.

#### *Ensamble de bomba de engranes*

Este ensamble consiste de cinco elementos: una carcasa, un engrane largo, un engrane corto y dos cojinetes, Figura 4.11a. Para realizar esta tarea primero se debe posicionar y orientar la carcasa de modo que el resto de los elementos puedan ser ensamblados en ella desde arriba, la cual sería la forma natural de realizar el ensamble real. A continuación se inserta el cojinete inferior, el cual deberá ubicarse hasta el fondo de la carcasa, tocando la mesa o piso de la escena virtual. Enseguida se ensambla el engrane largo insertándolo en la carcasa y verificando

que la flecha del engrane embone en el orificio del cojinete. Una vez ensamblado este elemento se deberá ensamblar el engrane corto, para lo cual se deben alinear los dientes de ambos engranes verificando también que la flecha sea insertada en el cojinete, tal y como sucede en el ensamble real. Finalmente se ensambla el cojinete superior, el cual debe ser insertado en las flechas superiores de los dos engranes y en la carcasa. Figura 4.11b. Al igual que los demás casos de ensamble que representan objetos reales, cada una de las piezas fue manufacturada utilizando técnicas de prototipado rápido, Figura 4.11c, con fines de comparar el proceso real y el virtual.

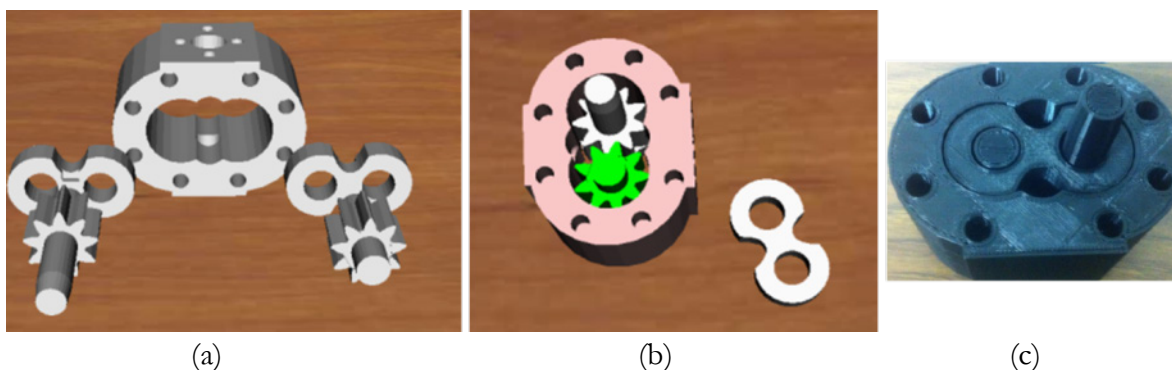


Figura 4.11. Caso de ensamble de bomba de aceite de engranes: (a) partes desensambladas, (b) proceso de ensamble, (c) prototipo real.

El ensamble de la bomba de engranes es representativo de tareas de ensamble reales que tienen características cilíndricas, es decir, se ensambla un perno o flecha sobre un agujero. Además presenta características que suponen cierta dificultad para el sistema de detección y respuesta a las colisiones, por ejemplo, al ensamblar el engrane corto se debe de calcular la respuesta a colisión entre tres elementos - el engrane corto con el engrane largo, el cojinete inferior y la carcasa, todos al mismo tiempo-. Una situación similar existe al ensamblar el cojinete superior, donde el sistema debe verificar colisiones para todos los elementos que componen el ensamble. Las tolerancias en este caso de ensamble son de 1.5mm entre la carcasa y los cojinetes, y entre la carcasa y los engranes, mientras que entre la flecha de los engranes y el orificio de los cojinetes existe una holgura de 2.5mm.

#### *Ensamble de cilindro neumático*

Este ensamble consta de ocho piezas: un cilindro hueco, un émbolo, una tapa trasera, una tapa delantera y cuatro tornillos largos. El procedimiento de ensamble consiste en primero ubicar la tapa inferior de modo que los orificios para los tornillos y el cilindro queden hacia arriba. Enseguida se ensambla el cilindro sobre la tapa trasera, en el cual se inserta el émbolo una vez ensamblado. Después se ensambla la tapa delantera sobre el eje del embolo hasta que se ajuste al cilindro, Figura 4.12a. Finalmente, cada uno de los tornillos es insertado en los orificios de las dos tapas, empezando por la tapa frontal, con lo cual el componente queda ensamblado,



Figura 4.12b. Los componentes de este ensamble también fueron manufacturados utilizando técnicas de prototipado rápido para realizar el proceso de ensamble real con fines comparativos, Figura 4.12c.

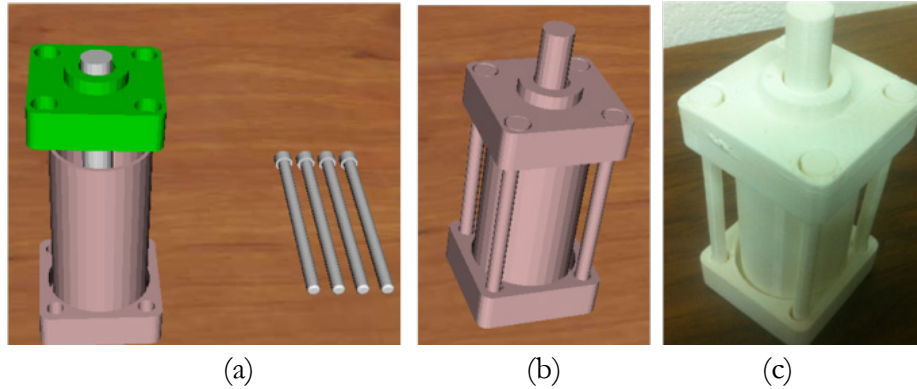


Figura 4.12. Caso de ensamble de cilindro neumático: (a) proceso de ensamble, (b) componente ensamblado, (c) prototipo rápido real.

Las características especiales de este caso de ensamble son los objetos cilíndricos largos que contiene, por ejemplo, los tornillos que deben ser insertados en las dos tapas separadas por el cilindro, lo cual representa condiciones de simulación difíciles para el sistema de detección y respuesta a colisiones. Asimismo, la pared delgada del cilindro representa ciertos problemas para la respuesta a colisiones, lo que puede ocasionar traslape y respuesta a colisiones no deseada durante la simulación.

Cada caso de ensamble descrito - extractor de poleas, prensa de banco, bomba de engranes y cilindro neumático - se llevó a cabo utilizando cada uno de los tres simuladores de física y cada uno de los métodos de representación disponibles mostrados en la Tabla 4.3. Se realizaron cinco repeticiones por cada factor a evaluar. Cada ensamble fue llevado a cabo por un experto con el objetivo de minimizar el efecto de la curva de aprendizaje. Los resultados se muestran y discuten en el capítulo V del presente documento.

#### 4.4 Análisis y evaluación del efecto del peso

El uso de dispositivos hápticos en aplicaciones de realidad virtual permite al usuario sentir el peso y la inercia de los objetos virtuales. La forma en que el usuario percibe el peso puede afectar la manipulación de dichos objetos. Por tanto, el objetivo de este análisis es evaluar la influencia del peso de los objetos virtuales en el desempeño del proceso de ensamble virtual.

El peso de un objeto es directamente proporcional a su masa. En el sistema HAMS la masa de los objetos es calculada por medio de la expresión  $Masa = volumen * densidad$ , donde el volumen se obtiene por medio de la función `getvolume()` incluida en el simulador gráfico VTK, la

cual permite obtener el volumen de un modelo descrito por una malla triangular. El sistema de ensambles desarrollado permite asignar uno de cuatro materiales diferentes a cada objeto: acero, plomo, aluminio o plástico, los cuales tienen diferente densidad; según el material seleccionado se asigna este valor de densidad al objeto.

Para manipular los objetos virtuales por medio del dispositivo háptico se utiliza un sistema masa-resorte-amortiguador (MRA) como se describió en el Capítulo II. El sistema MRA permite crear una fuerza entre la representación háptica y la representación física, la cual es utilizada para producir el movimiento de esta última representación y así actualizar la posición del objeto manipulado en cada ciclo de la simulación. Debido a que el sistema MRA está acoplado a la representación háptica, la reacción de esta fuerza es también renderizada al usuario por medio del dispositivo háptico. De este modo, cuando la representación física está sujeta al efecto de la gravedad, existe una fuerza que lo atrae hacia el suelo y que el usuario experimentará a través del dispositivo háptico, Figura 4.13. Esta fuerza gravitatoria es entonces proporcional a la masa del objeto y al valor de gravedad definido para la escena virtual física. De manera similar, el usuario puede sentir la inercia de los objetos virtuales.

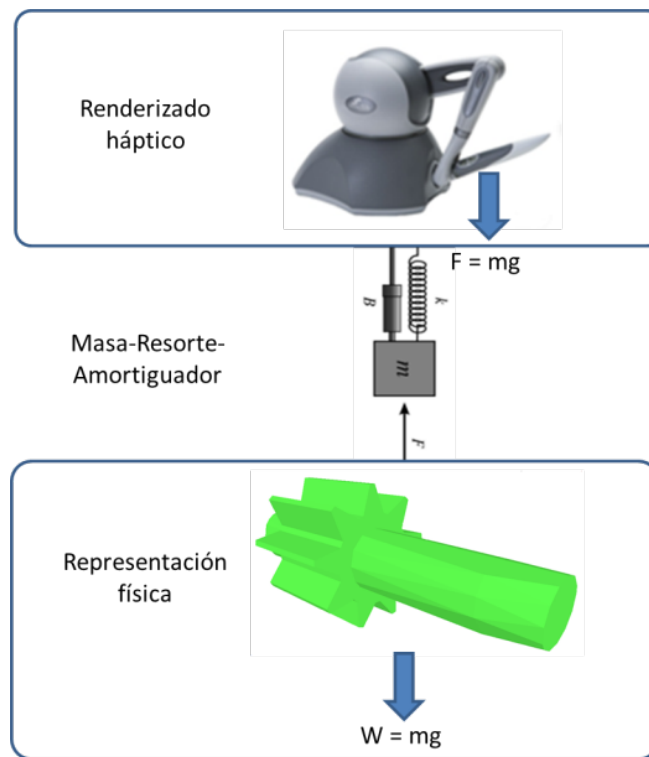


Figura 4.13. Percepción del peso por medio del dispositivo háptico.

Para realizar el análisis que permita determinar el efecto que tiene la percepción del peso sobre la tarea de ensamble se utiliza el caso de la bomba de engranes, Figura 4.11. Se selecciona este caso debido a que representa un caso de ensamble real, donde las partes tienen pesos

diferentes y las características del ensamblaje permiten generalizar el proceso, ya que se realizan ensamblajes de piezas cilíndricas y de formas irregulares. También se cuenta con las partes reales del componente, Figura 4.14, las cuales son utilizadas para realizar la comparativa del ensamblaje real contra el ensamblaje virtual.



Figura 4.14. Partes reales del caso de ensamblaje de bomba de engranes.

Para realizar la prueba se definen ocho niveles de peso para cada una de las partes de la bomba, donde el nivel uno es el peso mínimo que pueden tener los objetos para tener un comportamiento dinámico estable, mientras que el nivel ocho representa los pesos máximos, limitado por la fuerza máxima de 3.3 N que puede soportar el dispositivo háptico Phantom Omni y la cual se tomó en cuenta al asignar el peso del componente más pesado. Los pesos de cada pieza para cada uno de los niveles de peso se muestran en la Tabla 4.4. Estos pesos fueron generados al escalar los valores de densidad de los objetos virtuales. Únicamente se escaló el peso de los engranes y de los cojinetes, ya que la carcasa de la bomba se considera la pieza base y permanece estática durante la mayor parte del proceso de ensamblaje. Debido a las limitaciones del dispositivo háptico, los pesos de los objetos virtuales en el nivel de peso máximo son menores que los pesos de los objetos reales, por tanto, para este caso no es posible realizar el ensamblaje virtual utilizando los pesos reales.

Tabla 4.4. Niveles de peso para cada elemento de la bomba de engranes.

| Nivel de peso | Carcasa (N) | Engrane largo (N) | Engrane corto (N) | Cojinetes (N) |
|---------------|-------------|-------------------|-------------------|---------------|
| N1            | 0.02        | 0.02              | 0.02              | 0.02          |
| N2            | 1.3         | 0.17              | 0.13              | 0.1           |
| N3            | 3.3         | 0.41              | 0.34              | 0.29          |
| N4            | 3.3         | 0.82              | 0.66              | 0.51          |
| N5            | 3.3         | 1.11              | 0.9               | 0.69          |
| N6            | 3.3         | 1.64              | 1.31              | 1.01          |
| N7            | 3.3         | 2.23              | 1.81              | 1.34          |
| N8            | 3.3         | 3.24              | 2.71              | 1.47          |
| Real          | 16.7        | 6.7               | 5.2               | 1.6           |

También, como parte de este análisis se evalúa el efecto del peso cuando el ensamble es llevado a cabo con una sola mano o con dos manos. De manera similar, también se evalúa el efecto que tiene el simulador físico utilizado. Por tanto, se realizaron cuatro repeticiones del ensamble para cada nivel de peso por cada simulador físico y por cada modo de manipulación, es decir, una mano o dos manos. El ensamble virtual fue llevado a cabo por un usuario con experiencia en tareas de ensamble virtual háptico con el fin de minimizar los efectos de la curva de aprendizaje.

La tarea de ensamble real fue realizada por cinco participantes, donde cada uno realizó el ensamble con una mano y con dos manos. En ambas tareas, real y virtual, se midió el tiempo TFT como parámetro cuantitativo para evaluar el desempeño del proceso de ensamble. Adicionalmente, en el ensamble virtual se analizan las trayectorias de ensamble como un parámetro de evaluación cualitativo a través del uso de crono-ciclos.

#### 4.5 Análisis y evaluación de restricciones de ensamble dinámicas

Para analizar el efecto de las Restricciones de Ensamble Dinámicas (DACs) en el desempeño del ensamble virtual, se han seleccionado tres tareas de ensamble: extractor de poleas, bomba de engranes y cilindro neumático, cada una de las cuales ha sido descrita en las secciones anteriores y son representativas de casos de ensambles reales, Figura 4.15. En esta prueba se evalúan dos condiciones de simulación: 1) modelado basado en física y 2) modelado híbrido, es decir, utilizando las DACs. Se realizaron al menos cinco repeticiones para cada ensamble y para cada condición de simulación. Los ensambles virtuales fueron llevados a cabo por un usuario con experiencia en ensambles virtuales hápticos. Para realizar estas pruebas se utilizó el simulador de física Bullet y el método de representación por malla triangular GIMPACT, debido a que en pruebas preliminares mostraron el mejor desempeño. En este análisis se evalúa el desempeño del proceso de ensamble virtual mediante la medición del TFT. Asimismo también se compara el TFT con su correspondiente del ensamble real.

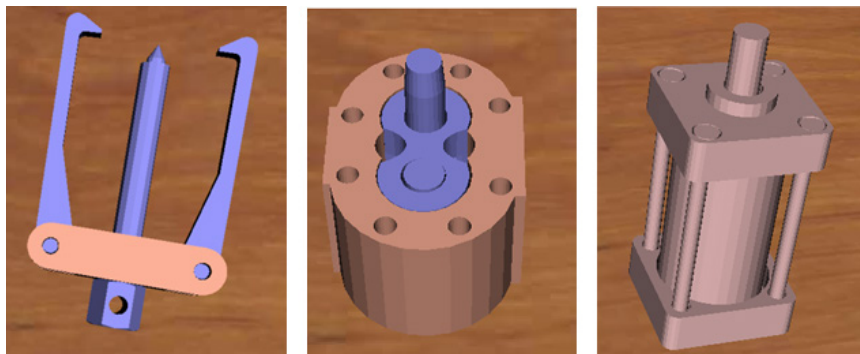


Figura 4.15. Casos de ensamble utilizados para la evaluación de DACs.

Por otro lado, también se evalúa el impacto que tiene el uso de las DACs sobre el desempeño del sistema. Esta evaluación se realiza mediante el análisis del tiempo que tarda cada módulo del sistema HAMS – háptico, gráfico y físico – en realizar un ciclo de simulación completo. De manera similar, también se observa el comportamiento de la fuerza de retroalimentación al utilizar las DACs. En la Tabla 4.5 se observa el diseño de experimentos para realizar esta prueba.

Tabla 4.5. Diseño de experimentos para evaluación de restricciones de ensamble.

| <b>Caso de ensamble</b> | <b>Método de ensamble</b> | <b>Parámetro a medir</b>                               |
|-------------------------|---------------------------|--|
| Extractor de poleas     | PBM, DACs y Real          | TFT, Tiempo de simulación, Fuerza de retroalimentación |
| Bomba de engranes       | PBM, DACs y Real          | TFT, Tiempo de simulación, Fuerza de retroalimentación |
| Cilindro neumático      | PBM, DACs y Real          | TFT, Tiempo de simulación, Fuerza de retroalimentación |

#### 4.6 Evaluación de la percepción del usuario sobre el sistema HAMS

Con el fin de validar la funcionalidad de la plataforma se pidió a varios participantes llevar a cabo un ensamble virtual utilizando el sistema HAMS y evaluar de manera cualitativa su experiencia en el proceso de ensamble virtual. En estas pruebas los usuarios realizaron el ensamble de la bomba de engranes utilizando tanto el modelado basado en física como las DACs.

Para realizar esta evaluación se crearon dos grupos de participantes. Con el fin de obtener una mayor confiabilidad de los datos obtenidos al reducir los efectos de la curva de aprendizaje, la prueba se dividió en tres etapas:

1. En la primera etapa se le permitió a cada participante utilizar el sistema de manera libre por 20 minutos con fines de familiarización con la interfaz y con el caso de ensamble.
2. En la segunda etapa se le pide a cada integrante del primer grupo realizar tres repeticiones del ensamble utilizando el modelado basado en física. Mientras que a cada integrante del segundo grupo se le pide realizar tres repeticiones del mismo caso pero utilizando las restricciones de ensamble dinámicas, es decir, empleando el enfoque híbrido.
3. En la tercera etapa se invierte el enfoque del ensamble para cada grupo, es decir, cada participante del primer grupo ahora realiza el ensamble utilizando el enfoque híbrido mientras que cada participante del segundo grupo realiza el mismo caso utilizando ahora el modelado basado en física.

Cada participante realiza las tres etapas en una misma sesión. Durante cada prueba se guardan todos los datos del proceso de ensamble como son nombre, posiciones y orientaciones del objeto manipulado, trayectorias del cursor háptico, fuerzas, tiempos de simulación y tiempo de finalización de la tarea.

Al final de cada una de las dos últimas etapas se le solicitó al usuario responder un cuestionario respecto a su experiencia en el proceso de ensamble y su percepción de la funcionalidad de la plataforma, Anexo A. En el cuestionario se evalúan cualitativamente los siguientes parámetros:

1. *Facilidad de controlar el dispositivo háptico*, desde 0 si el dispositivo es muy difícil de controlar, hasta 4 si es muy fácil y cómodo.
2. *Facilidad de realizar la tarea de ensamble*, desde 0 si el proceso de ensamble fue muy difícil, hasta 4 si el proceso fue muy fácil.
3. *Estabilidad de los objetos virtuales*, desde 0 si los objetos virtuales saltan en la escena virtual sin tocarlos, hasta 3 si los objetos son muy estables.
4. *Realismo del proceso*, desde 0 si el comportamiento de los objetos no es real, hasta 4 si los objetos se comportan exactamente como objetos reales.
5. *Retroalimentación de fuerza*, desde 0 si la retroalimentación de fuerza a través del dispositivo háptico no es perceptible, hasta 3 si todas las colisiones de los objetos manipulados son claramente perceptibles a través del dispositivo háptico.
6. *Percepción del tiempo*, desde 0 si el usuario percibe que el tiempo en completar el ensamble virtual es mucho mayor al real, hasta 3 si considera que el tiempo de ensamble virtual es menor al tiempo real.

Al finalizar todas las pruebas se pide al usuario contestar un segundo cuestionario referente a su percepción global del sistema de ensamblajes virtuales HAMS. Mediante dicho cuestionario se pretende obtener información que permite evaluar la funcionalidad y aplicación del sistema HAMS en términos generales. En este segundo cuestionario se evalúan los siguientes parámetros:

1. *Percepción de la experiencia de ensamble virtual háptico*. Se le pide al usuario seleccionar la palabra que mejor describa el proceso de ensamble: aburrido, agotador, estresante, interesante, útil o divertida.
2. *Precisión del sistema*. El usuario debe seleccionar una de cuatro opciones que describen la precisión del sistema: (i) sin precisión, los objetos se traslapan y la retroalimentación de fuerza no es perceptible; (ii) precisión baja, no hay traslape entre los objetos pero las reacciones a colisiones no son realistas; (iii) precisión media, los objetos se comportan de manera realista pero la retroalimentación háptica no es perceptible; (iv) precisión alta, los objetos se comportan tal y como objetos reales, además la fuerza de retroalimentación háptica es perceptible.
3. *Confianza al utilizar el sistema*. El usuario selecciona la opción que mejor describa su nivel de confianza al utilizar el sistema: (i) baja, no sabe lo que debe hacer ni cómo hacerlo,

- (ii) media, solo en ocasiones no sabe el siguiente paso durante un proceso; (iii) alta, el sistema es intuitivo y amigable, es decir, el usuario sabe cómo realizar el proceso.
4. *Reproducción de los movimientos reales.* El usuario evalúa cómo son los movimientos del objeto virtual manipulado respecto a los movimientos realizados por su mano: (i) baja, los objetos virtuales se mueven completamente diferente a la mano, (ii) media, las traslaciones son correctas pero no las rotaciones y; (iii) alta, los objetos virtuales se mueven tal y como la mano lo indica.
  5. *Factores de mayor relevancia.* El usuario enumera de mayor a menor una lista de factores que considera de importancia para que el sistema pueda ser utilizado en aplicaciones reales. La lista de factores a enumerar incluye el uso de objetos más complejos, escenas virtuales más complejas, mayor nivel de inmersión gráfica, mejor interacción háptica, mayor rapidez de la simulación y realismo en la representación gráfica de los objetos.
  6. *Consideración del tiempo de ensamble virtual.* El usuario responde si considera o no que el tiempo de ensamble virtual debe ser igual al tiempo de ensamble real para que el sistema pueda ser utilizado de manera práctica en casos reales.
  7. *Consideración del sistema HAMS como una herramienta de planeación.* El usuario responde si considera o no que el sistema puede ser utilizado en planeaciones o análisis de ensamble reales.
  8. *Ventajas y desventajas del sistema.* Se pide al usuario describir cuales son las ventajas y desventajas del sistema desde su punto de vista.
  9. *Áreas de aplicación del sistema.* Se pide al usuario describir, según su criterio, las aplicaciones prácticas que puede tener el sistema HAMS.
  10. *Propuesta de funcionalidad.* Se solicita al usuario describir qué funciones pueden ser agregadas la sistema para hacerlo más eficiente, intuitivo y práctico.

#### **4.7 Comparativa del sistema HAMS contra plataformas de ensamble similares**

A la fecha diversos autores han desarrollado plataformas para simular y planear actividades de manufactura, dentro de ellas los sistemas de ensamble virtual. Sin embargo, la gran mayoría de estos sistemas se encuentran aún en una fase de desarrollo, por lo tanto, no es posible encontrarlos de manera comercial o de manera libre para poder realizar un estudio comparativo práctico entre estas plataformas y el sistema HAMS.

Por otro lado, es posible llevar a cabo un estudio comparativo entre las características principales que reportan algunos de los sistemas de ensamble virtual más completos encontrados en la revisión bibliográfica. Del estudio bibliográfico realizado se han identificado las siguientes características como las más relevantes a considerar en un sistema de ensambles virtuales:

1. *El objetivo principal del sistema de ensambles.* Se refiere al propósito o aplicación que se le pretende dar al sistema desarrollado, el cual puede ser para entrenamiento, análisis de

- manufacturabilidad, generación de planes de ensamble, planeación de operaciones de mantenimiento, etc.
2. *La metodología utilizada para llevar a cabo el ensamble.* Se refiere a la forma en cómo se modela el comportamiento de los objetos y el proceso de ensamble, es decir, observar si el sistema desarrollado utiliza el modelado basado en física, el modelado basado en restricciones o el enfoque híbrido.
  3. *Registro de datos.* Se refiere al tipo de información que se considera relevante durante el proceso de ensamble virtual y la forma en que esta información es almacenada y presentada al usuario.
  4. *Tipo de interfaz.* Se refiere al tipo de dispositivos para la interacción usuario-computadora. Se analizan los dispositivos hápticos empleados, identificando sus grados de libertad, configuración y número de dispositivos. También se analiza el tipo de interfaz gráfica, la cual pueden ser un monitor convencional, pantalla montada a la cabeza (HMD), muros de proyección tridimensional, etc.
  5. *Método de importación de modelos.* Se refiere a la forma en que son cargados los objetos en el sistema de ensambles desarrollado. Se observa el formato y tipo de modelos que soporta como entrada y si existe intercambio de información con algún sistema CAD.
  6. *Método de representación de los modelos.* Se refiere al método de representación de los modelos dentro del sistema de detección de colisiones y al algoritmo utilizado para el manejo de respuesta a colisiones.
  7. *Ensamble colaborativo.* Se debe observar si el sistema presentado permite a múltiples usuarios llevar a cabo el mismo proceso de ensamble a la vez, indicando si esta colaboración es local, en el mismo espacio físico, o es remota, desde lugares geográficos distintos.
  8. *Materiales.* Se refiere al tipo de materiales que pueden asignarse a los objetos virtuales, con fines de percepción del peso, fricción, rigidez, etc.
  9. *Escena virtual.* Se observa el tipo de la escena virtual y la forma de interacción, es decir, se analizan los elementos de la escena virtual y la forma en que el usuario puede modificarla o adaptarla, por ejemplo, si la escena permite la manipulación de la cámara para obtener diferentes perspectivas del ensamble.
  10. *Manejo de sub-ensambles.* Se refiere a la capacidad del sistema de crear sub-ensambles una vez que se han ensamblado los objetos en su posición final. Se observa la forma en que se crean estos sub-ensambles, su comportamiento dinámico y la posibilidad de manipulación y desensamble.
  11. *Interfaz de entrenamiento.* Se refiere a la existencia de un módulo dedicado al entrenamiento del usuario en operaciones repetitivas, por ejemplo video de entrenamiento o guía háptica para realizar el ensamble.
  12. *Generación de planes de ensamble.* Se observa si el sistema propuesto tiene la capacidad de generar instrucciones de ensamble una vez que el proceso ha sido llevado a cabo en el ambiente virtual.



Los sistemas reportados en la literatura pueden contar con algunas otras funciones diferentes, e incluso, puede haber sistemas que sí cuenten con estas funciones pero no han sido declaradas en el reporte de presentación del sistema. Por otro lado, también pueden existir casos en donde se declare que el sistema cuenta con algunas características, pero que estas no sean completamente funcionales.

Para la evaluación comparativa se seleccionaron seis plataformas con características similares al sistema HAMS:

- SHARP (*System for Haptic Assembly and Realistic prototyping*), Figura 4.16a.
- HVAS (*Haptic-based Virtual Assembly System*), Figura 4.16b.
- HIVEx (*Haptically Enabled Interactive/Immersive Virtual Reality Experience*), Figura 4.16c.
- HIDRA (*Haptic Integrated Dis/Re-assembly Analysis*), Figura 4.19d.

Estos sistemas han sido reportados en la literatura y se encuentran entre los más actualizados y cuentan con un mayor nivel de funcionalidad.

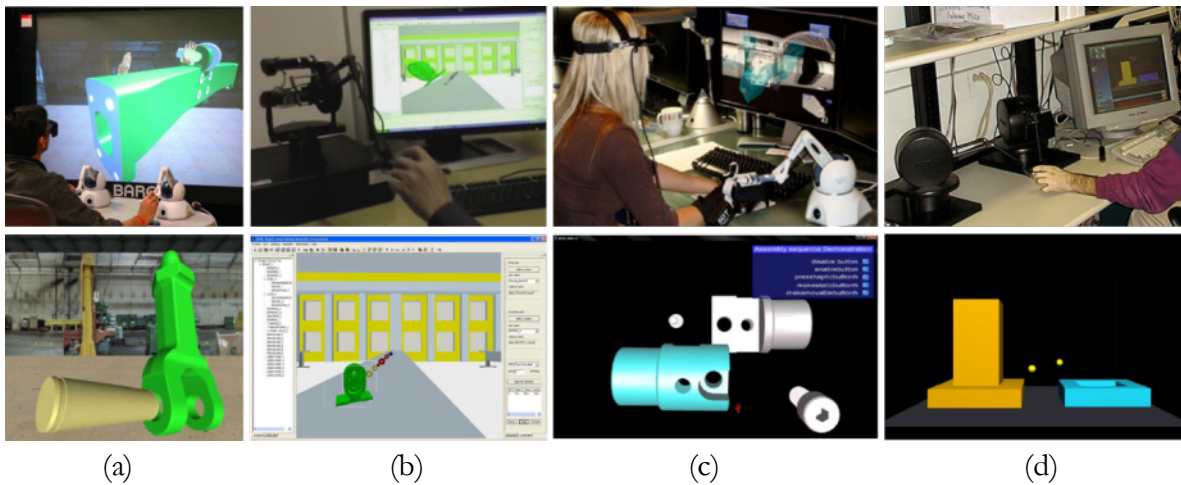


Figura 4.16. Sistemas de ensamble virtual con características similares a HAMS: (a) SHARP, (b) HVAS, (c) HIVEx, (d) HIDRA.

### 5.1 Introducción

Las pruebas presentadas en el capítulo anterior fueron propuestas y realizadas con el propósito de evaluar cuatro aspectos fundamentales que influyen en el desempeño del proceso de ensamble virtual: 1) el simulador físico, 2) el modo de representación de los modelos y su efecto en la detección de colisiones, 3) el uso de restricciones de ensamble, y 4) la percepción del peso a través del dispositivo háptico. Asimismo, también se evalúa el desempeño general del sistema desarrollado mediante dos actividades: 1) evaluación de la percepción del usuario sobre el sistema, y 2) una comparativa teórica contra sistemas con características y propósitos similares. En este capítulo se muestran los resultados obtenidos en cada una de las evaluaciones propuestas.

### 5.2 Evaluación de los simuladores físicos

El sistema HAMS permite realizar el proceso de ensamble virtual utilizando cualquiera de los tres simuladores de física disponibles: Bullet, PhysX v2.8.2 y PhysX v3.1.1. Para determinar las fortalezas y debilidades de cada simulador de física en diversas condiciones de ensamble se realizaron siete pruebas: prueba de caída libre, prueba de balanza, prueba de pila de cajas, prueba de empaquetado, prueba del tamaño de la malla, caso de ensamble del extractor de poleas y caso de ensamble del tornillo de banco. Los resultados de estas pruebas se presentan a continuación.

#### 5.2.1 Prueba de caída libre

En esta prueba se evaluó el método integrador de cada simulador mediante el análisis de dos factores: la complejidad del modelo utilizado y la influencia del ciclo de renderizado háptico.

##### *Complejidad del modelo*

En esta prueba se utilizaron cuatro objetos diferentes: una caja, un perno, un engrane y una carcasa, en donde el nivel de complejidad geométrica se ve reflejado en el número de triángulos que forman la malla triangular que describe cada objeto, 12, 44, 276 y 1934, respectivamente. Las pruebas fueron realizadas utilizando únicamente los módulos de simulación gráfico y físico. Los resultados respecto a la influencia de la complejidad de la forma se muestran en la Figura 5.1.

Se puede observar que cuando el número de triángulos del modelo es pequeño (en el caso de la caja, el perno y el engrane), el desempeño del método integrador de PhysX v2.8.2 se mantiene uniforme, dando como resultado que el tiempo de caída de cada objeto permanezca constante. Sin embargo, en el caso del simulador Bullet, se observa un incremento de hasta 43% en el

tiempo de caída libre del engrane con respecto al perno. En el caso de modelos complejos compuestos por alrededor de 2000 triángulos, el desempeño del integrador del simulador PhysX v2.8.2 se ve altamente afectado, pues se observa un incremento de hasta 100% en el tiempo de caída de la carcasa con respecto al engrane. Al utilizar el simulador Bullet se observa un incremento de alrededor de 13% para el objeto más complejo.

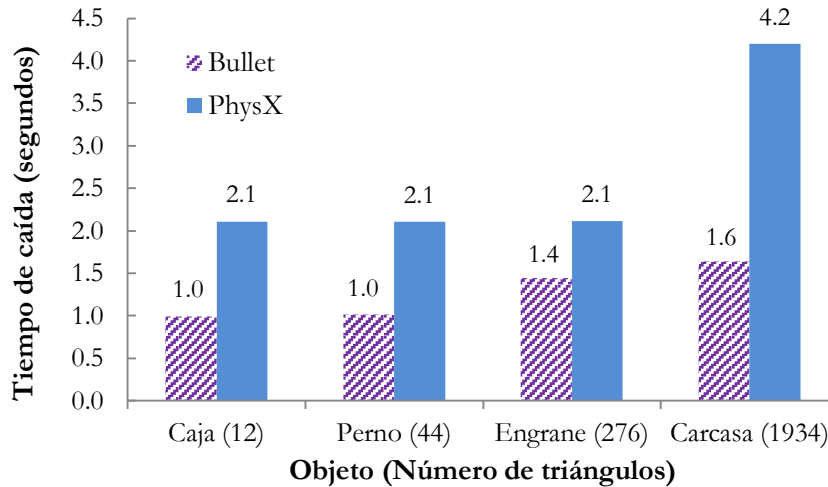


Figura 5.1. Tiempo de caída libre de objetos con diferente complejidad.

La desviación estándar máxima en las pruebas realizadas con el simulador PhysX es de 0.29s, que corresponde al caso de la carcasa. Mientras que la desviación estándar máxima en las repeticiones hechas utilizando Bullet es de 0.06s para el mismo objeto, lo que refleja la gran repetibilidad que tiene este simulador físico.

Los resultados anteriores indican que el método integrador utilizado en PhysX se ve altamente afectado por la complejidad del modelo. Además, al utilizar Bullet los tiempos de caída libre son más pequeños y cercanos al tiempo de caída teórico (0.32s) que los obtenidos al utilizar PhysX. Cabe hacer mención que aunque el tiempo de caída virtual es afectado por diversos parámetros como el tiempo de paso de la simulación, las características del CPU, la configuración del sistema, etc., las pruebas anteriores fueron realizadas en igualdad de condiciones para ambos simuladores, es decir, mismo CPU, mismo tiempo de paso de la simulación, etc.

#### *Influencia del ciclo de renderizado háptico*

Con el fin de cuantificar la influencia del ciclo de renderizado háptico sobre el desempeño del integrador, se llevó a cabo un segundo conjunto de pruebas de caída libre pero ahora incluyendo también el ciclo de renderizado háptico. En la Tabla 5.1 se muestra el porcentaje de incremento en el tiempo de caída libre de los objetos cuando el ciclo de renderizado háptico está activo respecto a la condición de simulación sin renderizado háptico.

Tabla 5.1. Porcentaje de incremento en el tiempo de caída libre.

| Objeto  | Bullet | PhysX |
|---------|--------|-------|
| Caja    | 3.6%   | 0.2%  |
| Perno   | 3.8%   | 0.1%  |
| Engrane | 50.4%  | 2.2%  |

Los resultados muestran que al utilizar Bullet el tiempo de caída libre incrementó un 50%, mientras que al utilizar el simulador PhysX se observa un incremento de alrededor 2%. Estos resultados sugieren que el método integrador utilizado en PhysX puede adaptarse mejor a aplicaciones que incluyen renderizado háptico.

### 5.2.2 Prueba de balanza

La prueba de la balanza pretende medir la precisión en la respuesta a colisiones de cada simulador físico. El resultado está expresado como derecha o izquierda, dependiendo de la inclinación de la balanza. En caso de una inclinación muy pronunciada el resultado se expresa como izquierda+ o derecha+. Para esta prueba se realizaron cinco repeticiones por cada simulador físico y por cada método de representación física -Primitivo, HACD, GIMPACT o ConvexFT-. La Tabla 5.2 muestra los resultados de cada una de las repeticiones llevadas a cabo en esta prueba. En la primera columna se observa el simulador físico utilizado, mientras que en la segunda columna se muestra el método utilizado para crear la representación física. El número indicado en cada una de las celdas de resultados muestra las veces que se repitió la inclinación (izquierda+, izquierda, equilibrio, derecha o derecha+) para el simulador físico y modo de representación indicados.

Tabla 5.2. Resultados de la prueba de balanza respecto a cada simulador de física.

| Simulador | Representación | Izquierda+ | Izquierda | Equilibrio | Derecha | Derecha+ |
|-----------|----------------|------------|-----------|------------|---------|----------|
| Bullet    | Primitivo      | 0          | 0         | 5          | 0       | 0        |
| PhysX     | Primitivo      | 0          | 0         | 5          | 0       | 0        |
| Bullet    | GIMPACT        | 2          | 0         | 1          | 1       | 1        |
| PhysX     | ConvexFT       | 4          | 0         | 0          | 0       | 1        |
| Bullet    | HACD           | 3          | 0         | 2          | 0       | 1        |
| PhysX     | HACD           | 3          | 2         | 0          | 0       | 0        |

Se puede observar que cuando se utiliza la representación por primitivos con cualquiera de los dos simuladores de física, Bullet o PhysX, la balanza permaneció en equilibrio para las cinco repeticiones. Esto sugiere que la respuesta a colisiones es estable y precisa cuando se utiliza este tipo de representación. Sin embargo, al utilizar cualquiera de los otros métodos de

representación disponibles -GIMPACT, ConvexFT o HACD-, la balanza se inclina hacia diferentes lados en cada una de las repeticiones, lo cual indica que la respuesta a colisiones para estos métodos de representación no es precisa. Esta imprecisión se debe a que en las representaciones basadas en malla triangular pueden existir diferentes puntos de contacto, ocasionando que la respuesta a colisiones sea diferente para cada esfera.

### 5.2.3 Pila de cajas

En esta prueba se evaluó el desempeño de los simuladores físicos utilizando el método de representación por malla triangular GIMPACT para el caso de Bullet, y el método de representación ConvexFT para el caso de PhysX. Además, también se analizó el comportamiento al representar los objetos por medio del algoritmo HACD en ambos simuladores. Se realizaron cinco repeticiones por cada método de representación y por cada simulador. Los resultados se observan en la Tabla 5.3, donde se muestra el promedio del tiempo de finalización de la tarea y su desviación estándar (DE) en segundos, para cada factor evaluado.

Tabla 5.3. Tiempo de finalización de la tarea para la prueba de apilado de cajas.

| Representación   | Bullet (s)    | PhysX (s)        |
|------------------|---------------|------------------|
| HACD             | 167s (35s DE) | 117s (0.3s DE)   |
| GIMPACT/ConvexFT | No posible    | 114s (13.6s DE). |

Al utilizar el método de representación HACD en Bullet las cajas mostraron baja estabilidad en la respuesta a colisiones, provocando que se movieran ligeramente al estar ensambladas. Debido a esta situación solo se pudieron apilar hasta un máximo de 10 cajas, después de las cuales la pila se derrumbaba. Por otro lado, no fue posible llevar a cabo el apilado de las cajas al utilizar el algoritmo GIMPACT en Bullet, esto debido a que la respuesta a colisiones no permitió poner una caja encima de la otra, pues inmediatamente que se colocaba la caja, esta saltaba fuera de la posición deseada.

En el caso del simulador PhysX, se observó un comportamiento más estable, pues mediante esta configuración se pudieron ensamblar las 15 cajas utilizando cualquiera de los dos algoritmos de representación, HACD o ConvexFT. Los tiempos de finalización de la tarea para ambos algoritmos resultaron muy similares, y por debajo de los obtenidos con Bullet.

### 5.2.4 Prueba de empaquetado

En esta prueba se evalúa el desempeño de cada simulador de física cuando existen contactos acumulativos en múltiples direcciones. Para llevar a cabo esta prueba los objetos fueron representados mediante el algoritmo HACD en ambos simuladores físicos. Se realizaron cinco repeticiones para cada simulador. Se midió el tiempo de finalización de la tarea desde que se empieza a manipular la primera caja hasta el momento en el que se suelta la última. La

respuesta a colisiones y la estabilidad del ensamble fueron evaluadas cualitativamente de acuerdo a los valores mostrados en la Tabla 5.4.

Tabla 5.4. Valores cualitativos para evaluación de la estabilidad y respuesta a colisiones.

| Valor | Respuesta a colisiones  | Estabilidad del ensamble  |
|-------|---|---|
| 3     | La respuesta es similar a la de objetos reales                        | Los objetos permanecen estáticos en su posición de ensamble final |
| 2     | Traslape o fuerzas irreales solo ocasionalmente                       | Los objetos tiemblan en su posición de ensamble final             |
| 1     | Respuesta a colisiones excesiva, los objetos responden exageradamente | Los objetos saltan de su posición de ensamble                     |
| 0     | No hay respuesta a colisiones, traslape evidente                      | El ensamble se destruye por completo                              |

Los resultados obtenidos muestran que al utilizar PhysX se tiene el tiempo menor de finalización de la tarea, 129 segundos con una desviación estándar de 4 segundos, mientras que al utilizar Bullet se tiene un tiempo promedio de 257 segundos con una desviación estándar de 16 segundos. Respecto a la evaluación cualitativa, se observó que PhysX presenta una mejor estabilidad del componente ensamblado. Sin embargo, se percibe una mejor respuesta a colisiones del objeto manipulado al utilizar Bullet.

### 5.2.5 Prueba de tamaño de la malla

En esta prueba se analiza el tamaño mínimo de los objetos que soporta cada simulador físico. Se evalúan dos configuraciones cinemáticas: 1) comportamiento dinámico de la pieza base y 2) comportamiento estático de la pieza base. En ambos simuladores los objetos son representados por medio de mallas triangulares. Se realizaron cinco repeticiones por cada escala, cada simulador de física y cada configuración cinemática. Los resultados se muestran en la Tabla 5.5, en la que se muestra el tiempo de finalización de la tarea (TFI) en segundos y el tiempo de la simulación física (TSF) en milisegundos, el cual es el tiempo promedio que tarda el simulador en resolver la nueva posición de los objetos en cada ciclo. Los datos marcados con asterisco \*, indican que la detección de colisiones no fue tan precisa y se observó un ligero traslape entre los objetos. “NP” indica que no fue posible realizar el ensamble debido a traslape excesivo entre los objetos. Las celdas marcadas en azul indican que se observa un incremento notable en el tiempo de simulación física con respecto al tiempo obtenido en la escala anterior.

Tabla 5.5. Resultados de la prueba de tamaño de la malla.

| Escala     | Bullet-dinámico |         | PhysX-dinámico |         | Bullet-estático |         | PhysX-estático |         |
|------------|-----------------|---------|----------------|---------|-----------------|---------|----------------|---------|
|            | TFT (s)         | TSF(ms) | TFT(s)         | TSF(ms) | TFT(s)          | TSF(ms) | TFT(s)         | TSF(ms) |
| 1x         | 6.1             | 1.80    | 12.7           | 16.46   | 11.2            | 2.36    | 5.2            | 0.67    |
| 0.1x       | 5.4             | 1.84    | 6.2            | 11.02   | 9.1             | 2.40    | 5.2            | 0.59    |
| 0.01x      | 5.1             | 1.80    | 11.0*          | 20.12   | 12.0            | 3.44    | 5.5            | 0.52    |
| 0.001x     | 4.4             | 1.96    | NP             | NP      | 7.7             | 2.72    | 5.7            | 0.54    |
| 0.0001x    | 4.0             | 5.05    | NP             | NP      | NP              | NP      | 8.0            | 0.54    |
| 0.00001x   | 4.7             | 5.24    | NP             | NP      | NP              | NP      | 6.3*           | 1.05    |
| 0.000001x  | 4.9*            | 4.85    | NP             | NP      | NP              | NP      | NP             | NP      |
| 0.0000001x | 6.1*            | 6.30    | NP             | NP      | NP              | NP      | NP             | NP      |

Analizando los resultados se observa que al emplear Bullet y la pieza base con comportamiento dinámico, el ensamble pudo ser realizado utilizando incluso la escala más pequeña de 0.0000001x que corresponde a piezas con un volumen de  $1.69 \times 10^{-13} \text{ mm}^3$ . Sin embargo, un desempeño satisfactorio fue observado hasta la escala de 0.001x, pues para objetos más pequeños el tiempo de simulación física se incrementó considerablemente. En el caso de PhysX con comportamiento dinámico de la pieza base, la escala más pequeña para un desempeño satisfactorio fue de 0.1x, pues a escalas más pequeñas la detección de colisiones no fue precisa y el tiempo de simulación física se incrementó considerablemente.

Para el caso donde la pieza base permanece estática, el ensamble pudo ser realizado con un desempeño satisfactorio con escalas de 0.001x o mayores al utilizar Bullet. En el caso de PhysX el ensamble pudo ser realizado hasta una escala de 0.0001x. Al utilizar escalas más pequeñas ocurrió traslape entre los objetos.

### 5.2.6 Ensamble de extractor de poleas

Mediante esta prueba se pretende evaluar el desempeño de los simuladores físicos al realizar tareas de ensamble que representan componentes reales. Se utilizan los métodos GIMPACT en Bullet y ConvexFT en PhysX para crear la representación física de los modelos. Se realizaron seis repeticiones para cada simulador físico, en las cuales el proceso de ensamble fue llevado a cabo por un usuario con experiencia en ensambles virtuales hápticos, esto con el fin de minimizar los efectos de la curva de aprendizaje, y utilizando solo una mano para la manipulación de los objetos virtuales. Se midió el TFT, la fuerza de retroalimentación y el TSF con el fin de evaluar el desempeño de cada simulador. Los resultados de estas pruebas se presentan en la Tabla 5.6 donde se muestran los valores promedio de cada parámetro medido y la desviación estándar (DE) de las repeticiones.

Tabla 5.6. Resultados del ensamble del extractor de poleas, evaluación de simuladores físicos.

| Parámetro                                   | Bullet |      | PhysX |      |
|---|--------|------|-------|------|
|   | Valor  | DE   | Valor | DE   |
| Tiempo de finalización de la tarea, TFT (s) | 79     | 12   | 219   | 32   |
| Fuerza de retroalimentación media (N)       | 0.55   | 0.1  | 0.62  | 0.1  |
| Fuerza de retroalimentación máxima (N)      | 1.88   | 0.16 | 3.55  | 0.34 |
| Tiempo de simulación física, TSF (ms)       | 4.7    | 1.0  | 4.4   | 0.4  |

Del análisis de estos resultados se observa que el menor tiempo de ensamble (y con la menor desviación estándar) se obtuvo al utilizar el simulador Bullet. Asimismo, las fuerza de retroalimentación promedio y máxima son menores al utilizar este simulador, lo que sugiere que Bullet ofrece una respuesta a colisiones y una manipulación del objeto más estable que PhysX. El tiempo de simulación física es una medida de la tasa de actualización del simulador físico, en este caso ambos simuladores muestran tiempos de actualización similares, alrededor de 4.5 milisegundos. Mediante estos resultados es posible concluir que Bullet ofrece un mejor desempeño para ensambles con características similares al caso del extractor de poleas. Finalmente, el ensamble real de este mismo componente fue llevado a cabo en 18 segundos con una desviación estándar de 2 segundos, el cual es menor que el tiempo de ensamble virtual.

### 5.2.7 Ensamble de tornillo de banco

Este caso también involucra modelos que representan partes reales con características especiales de ensamble debido a su complejidad geométrica. La representación física de los objetos virtuales fue creada utilizando los métodos GIMPACT y HACD para Bullet y PhysX respectivamente. A diferencia del caso anterior, en este ensamble no fue posible utilizar la representación ConvexFT debido a que presenta una respuesta a colisiones excesiva y poco realista al ensamblar las dos mordazas, lo que imposibilita el proceso de ensamble de estos dos elementos. Por lo tanto, se utilizó el método HACD para objetos dinámicos y la representación por malla triangular para objetos estáticos en PhysX.

Un usuario con experiencia realizó seis repeticiones del ensamble por cada simulador físico, utilizando solo una mano para la manipulación de los objetos. Se midió el TFT, la fuerza de retroalimentación y el TSF, los resultados se muestran en la Tabla 5.7.

Los resultados muestran que los valores de TFT, fuerza promedio y fuerza máxima son menores al utilizar Bullet; indicando que Bullet permite una mejor manipulación de los objetos virtuales y una respuesta a colisiones más estable. Por otro lado, el TSF es mucho menor al utilizar PhysX, 0.9 ms, que al utilizar Bullet, 4.4 ms, lo cual se debe al uso de elementos estáticos para poder realizar el ensamble. Sin embargo, aunque PhysX ofrece mayor rapidez, sus algoritmos de representación de modelos con fines de detección de colisiones deben ser mejorados.



Tabla 5.7. Resultados del ensamble del tornillo de banco, evaluación de simuladores físicos.

| Parámetro                                   | Bullet |      | PhysX |      |
|---|--------|------|-------|------|
|   | Valor  | DE   | Valor | DE   |
| Tiempo de finalización de la tarea, TFT (s) | 75     | 10   | 78    | 22   |
| Fuerza de retroalimentación media (N)       | 0.5    | 0.1  | 0.6   | 0.04 |
| Fuerza de retroalimentación máxima (N)      | 1.89   | 0.24 | 2.76  | 1.14 |
| Tiempo de simulación física, TSF (ms)       | 4.4    | 0.3  | 0.9   | 0.2  |

Este caso de ensamble también fue realizado utilizando componentes reales, obteniéndose un tiempo promedio de 11 segundos para completar el ensamble, con una desviación estándar de 1 segundo. Nuevamente el ensamble real es realizado más rápido que el ensamble virtual.

### 5.2.8 Resumen de la evaluación de los simuladores físicos

Los resultados de las pruebas realizadas para evaluar el desempeño de los simuladores físicos son presentados de manera condensada en la Tabla 5.8. En esta tabla los valores resaltados en color rojo indican el resultado más favorable para el parámetro evaluado.

Tabla 5.8. Resultados de la evaluación a los simuladores físicos.

| Prueba                              | Bullet v2.81                 | PhysX v2.8.4   | Parámetro de evaluación          |
|-------------------------------------|------------------------------|----------------|----------------------------------|
| Caída libre, complejidad geométrica | <b>60 %</b>                  | 100 %          | Incremento en el tiempo de caída |
| Caída libre, ciclo háptico          | 50 %                         | <b>2 %</b>     | Incremento en el tiempo de caída |
| Balanza                             | Estable solo para primitivos |                | Inclinación de la balanza        |
| Pila de cajas (HACD)                | 167s                         | <b>117s</b>    | Tiempo de ensamble (TFT)         |
| Pila de cajas (Malla triangular)    | No posible                   | <b>114s</b>    | Tiempo de ensamble (TFT)         |
| Empaquetado (TFT)                   | 257s                         | <b>130s</b>    | Tiempo de ensamble (TFT)         |
| Empaquetado, estabilidad            | 2                            | <b>3</b>       | Estabilidad del ensamble         |
| Empaquetado, colisiones             | <b>3</b>                     | 2              | Respuesta a colisiones           |
| Tamaño de malla, dinámico           | <b>0.001x</b>                | 0.1x           | Escala                           |
| Tamaño de malla, estático           | 0.001x                       | <b>0.0001x</b> | Escala                           |
| Extractor de poleas, TFT            | <b>79s</b>                   | 219s           | Tiempo de ensamble (TFT)         |
| Extractor de poleas, TSF            | 4.7ms                        | <b>4.4ms</b>   | Tiempo de simulación (TSF)       |
| Tornillo de banco, TFT              | <b>75s</b>                   | 78s            | Tiempo de ensamble (TFT)         |
| Tornillo de banco, TSF              | 4.4ms                        | <b>0.9ms</b>   | Tiempo de simulación (TSF)       |

De la Tabla 5.8 se puede observar que PhysX muestra un mejor desempeño que Bullet cuando la tarea involucra componentes de baja complejidad, por ejemplo, en el apilado de cajas donde se utilizan modelos muy sencillos. Para tareas de ensamble más complejas, como los casos del extractor de poleas y el tornillo de banco, Bullet muestra un mejor desempeño, el cual se ve reflejado en una mejor respuesta a colisiones y consecuentemente un menor tiempo de finalización de la tarea. También se puede observar en la prueba de caída libre que el método integrador utilizado en Bullet es menos sensible a la complejidad de los modelos utilizados que el integrador de PhysX. Sin embargo, este último es menos sensible al ciclo de renderizado háptico.

### 5.3 Evaluación de los métodos de representación física

Esta evaluación tiene como objetivo determinar la influencia del método utilizado para crear la representación física de los modelos en el desempeño del ensamble virtual háptico. La evaluación se dividió en dos etapas: evaluación de la precisión geométrica y evaluación del desempeño.

#### 5.3.1 Evaluación de precisión geométrica

La precisión geométrica es evaluada al comparar visualmente el modelo original contra el modelo creado por el método de representación seleccionado, es decir, se compara la representación gráfica original contra la representación física. La inspección visual consiste en verificar que las características geométricas más relevantes del modelo, como agujeros, esquinas, bordes y concavidades, sean reproducidas correctamente en la representación física creada en el simulador físico por medio de cualquiera de los algoritmos disponibles -ACD, HACD, GIMPACT o ConvexFT-. Se utilizaron nueve modelos con características diferentes para esta etapa de la prueba: una caja, una caja con agujero, un perno, un engrane, un cojinete, la carcasa de la bomba, la base del extractor, el tornillo del extractor y el brazo del extractor, Figura 5.2.

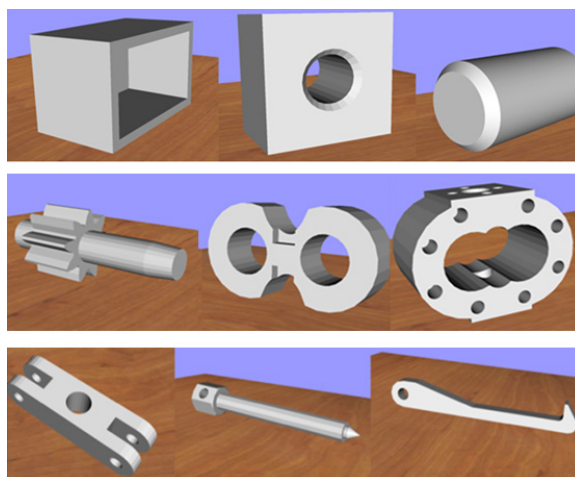


Figura 5.2. Modelos originales utilizados en la evaluación de precisión geométrica.

El primer algoritmo evaluado es ACD (*Approximate Convex Decomposition*), el cual utiliza la malla triangular original del modelo para crear un nuevo objeto compuesto de múltiples elementos convexos. ACD es la solución principal que ofrece el simulador PhysX para resolver la detección de colisiones de objetos no convexos con comportamiento dinámico. Los resultados se muestran en la Figura 5.3 y fueron obtenidos utilizando parámetros que permiten realizar la descomposición con la mayor precisión posible. De estos resultados se puede observar que en algunos modelos, por ejemplo la carcasa y la caja con agujero, las concavidades no son reproducidas con buena precisión, incluso existen agujeros que no se reproducen en la representación física, como es el caso de las partes del extractor de poleas. Además en otros objetos como la caja, el engrane y el cojinete, se pueden observar excesos de material que no existen en el modelo original, lo cual imposibilitaría realizar el proceso de ensamble de manera correcta. Por lo tanto, se puede decir que la representación física creada por el algoritmo ACD no es adecuada para tareas de ensamble virtual, en las cuales se requiere una buena precisión geométrica del modelo.

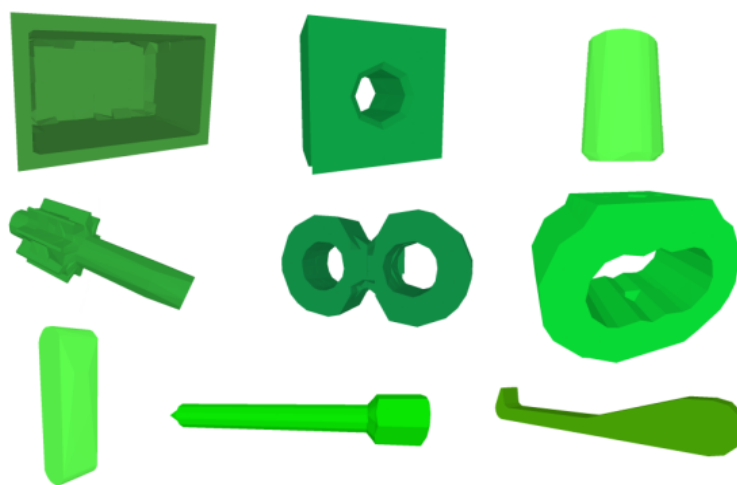


Figura 5.3. Representación física de los objetos utilizando el algoritmo ACD en PhysX.

El siguiente método evaluado es HACD (*Hierarchical Approximate Convex Decomposition*) que utiliza la malla triangular que describe al modelo para descomponerlo en varios elementos convexos, los cuales son unidos para crear así la representación física del objeto. Este algoritmo está implementado tanto en Bullet como en PhysX, por lo que se analizaron los resultados producidos en ambos simuladores. Los resultados de HACD en Bullet se muestran en la Figura 5.4. Estos resultados fueron obtenidos utilizando los parámetros de descomposición sugeridos por el desarrollador de este algoritmo. De estos resultados se puede observar que el algoritmo HACD es efectivo para crear objetos no convexos que no poseen agujeros, como el engrane o el tornillo del extractor. Sin embargo, la representación de concavidades internas, como agujeros, no es lo suficientemente precisa pues éstas se encuentran muy distorsionadas.

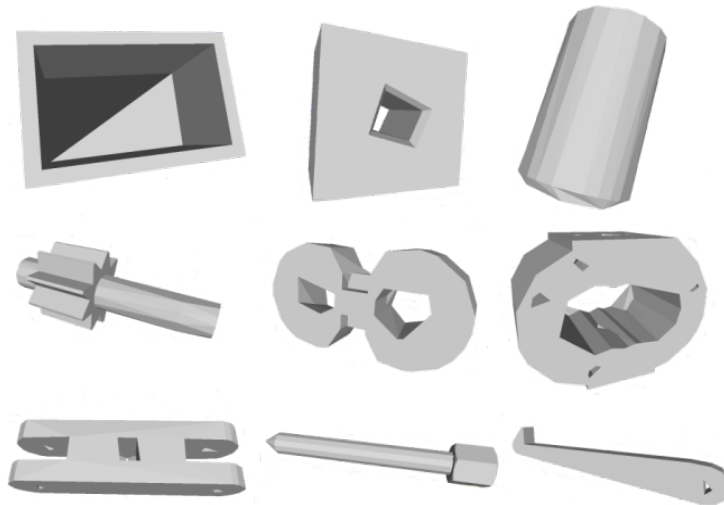


Figura 5.4. Representación física de los objetos utilizando el algoritmo HACD en Bullet con parámetros de descomposición recomendados.

Para obtener una mejor precisión en los resultados de la descomposición, el algoritmo HACD permite manipular el número de grupos o secciones mínimas en las que se divide el modelo, así, al aumentar este número se aumentara la precisión de la descomposición. Con el fin de determinar qué número mínimo de grupos produce una representación física adecuada para el ensamble, se varió este número gradualmente hasta obtener la precisión geométrica deseada. En la Tabla 5.9 se presenta el número de triángulos que componen a cada objeto y la cantidad de grupos utilizados para obtener una representación física adecuada. Se puede observar que cuando el número de grupos es alrededor del 90% del número de triángulos, se produce una representación física con buena precisión geométrica. Los resultados de esta descomposición se observan en la Figura 5.5.

Tabla 5.9. Número de grupos mínimos para una buena precisión geométrica.

| Objeto                 | No. triángulos | No. grupos |
|------------------------|----------------|------------|
| Engrane                | 276            | 2          |
| Cojinete               | 294            | 260        |
| Carcasa                | 1939           | 1825       |
| Base del extractor     | 460            | 420        |
| Tornillo del extractor | 254            | 190        |
| Brazo del extractor    | 132            | 88         |
| Caja con agujero       | 160            | 130        |

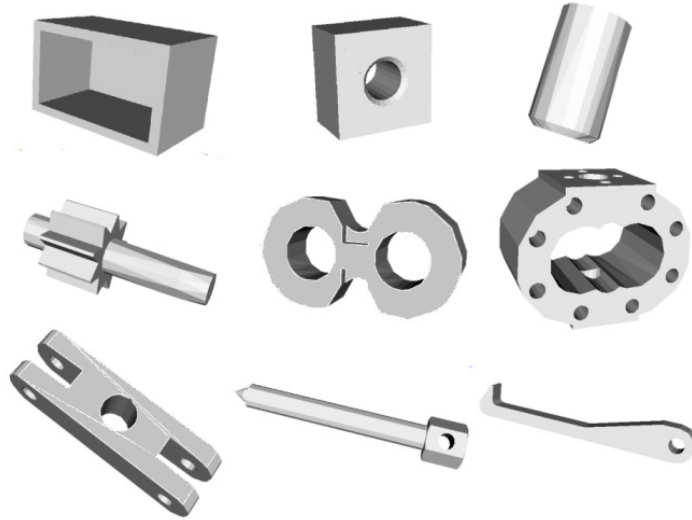


Figura 5.5. Representación física de los objetos utilizando el algoritmo HACD en Bullet con un número de grupos alrededor del 90% del número de triángulos en la malla.

El algoritmo HACD también fue evaluado en el simulador PhysX, los resultados se muestran en la Figura 5.6. Para realizar la descomposición se utilizaron los parámetros recomendados por el desarrollador de HACD. En estos resultados se puede observar que algunos objetos como el contenedor, la caja con agujero, el perno y el engrane son representados de manera correcta. Sin embargo, objetos como los cojinetes, la carcasa, la base del extractor y el brazo del extractor no son representados de manera precisa, pues sus concavidades internas aparecen incompletas.

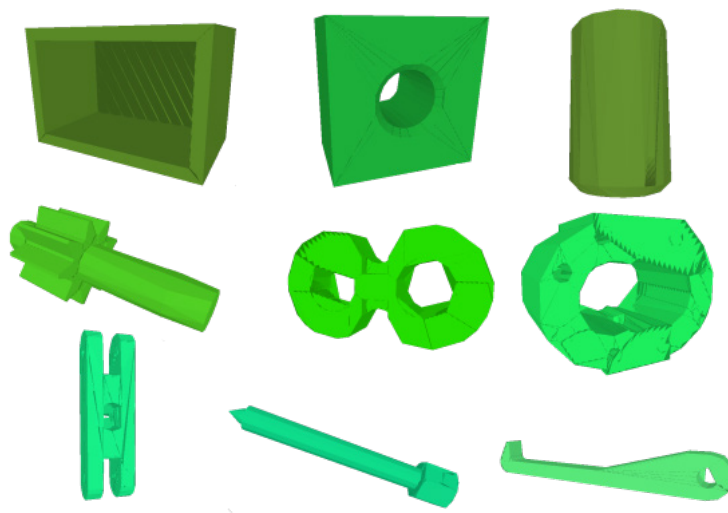


Figura 5.6. Representación física de los objetos utilizando el algoritmo HACD en PhysX.

Al igual que en el caso anterior, se intentó incrementar el número de grupos de la descomposición, sin embargo los resultados no fueron satisfactorios, pues los objetos

resultantes son completamente diferentes al objeto original, apareciendo deformes o incompletos, Figura 5.7. Estos errores son causados porque la forma en que el algoritmo HACD presenta los resultados de la descomposición no es compatible con la forma en que PhysX crea los elementos convexos. Para crear un elemento convexo en PhysX se requieren definir al menos 4 vértices los cuales no deben de estar en el mismo plano, estar muy cerca uno del otro o formar bordes muy delgados. Estas condiciones no siempre se cumplen en los resultados arrojados por HACD al aumentar el número de grupos.

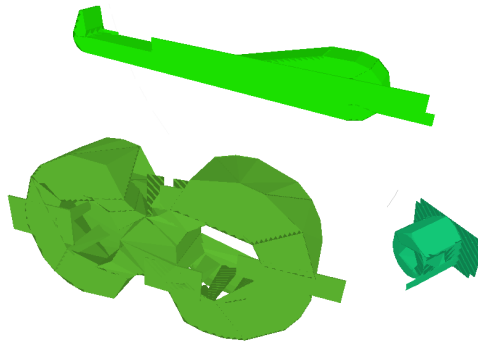


Figura 5.7. Errores en la representación de objetos al incrementar el número de grupos en la descomposición HACD-PhysX.

También se evaluó el algoritmo GIMPACT, integrado originalmente en el simulador Bullet, el cual utiliza la misma malla triangular que describe al objeto para realizar la detección de colisiones, por tanto la representación física de este método debe ser muy similar a la representación gráfica. Los resultados de este método de representación se muestran en la Figura 5.8, donde se puede observar que GIMPACT provee un alto nivel de precisión para la representación física de los modelos.

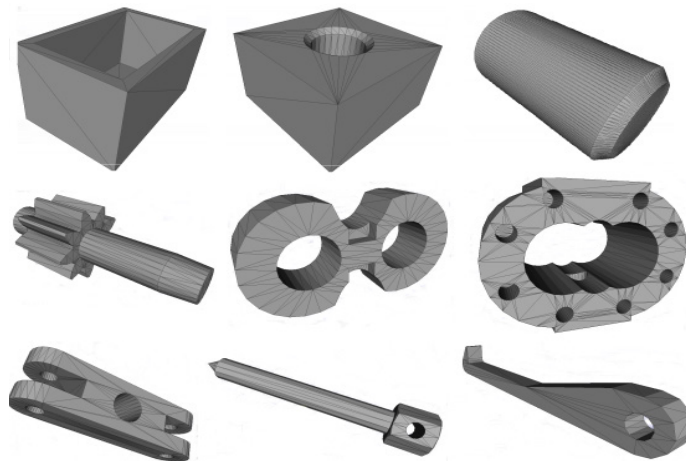


Figura 5.8. Representación física de los objetos utilizando el algoritmo GIMPACT en Bullet.

Finalmente se evaluó el método de representación propuesto en este trabajo, ConvexFT, en el cual cada triángulo que contiene la malla que describe al objeto es transformado en un elemento convexo. Los resultados del algoritmo ConvexFT para crear la representación física en el simulador PhysX se muestran en la Figura 5.9, en donde se puede observar una buena precisión geométrica de la representación obtenida, incluso la calidad de la representación es superior a los resultados obtenidos con los métodos HACD y ACD en PhysX.

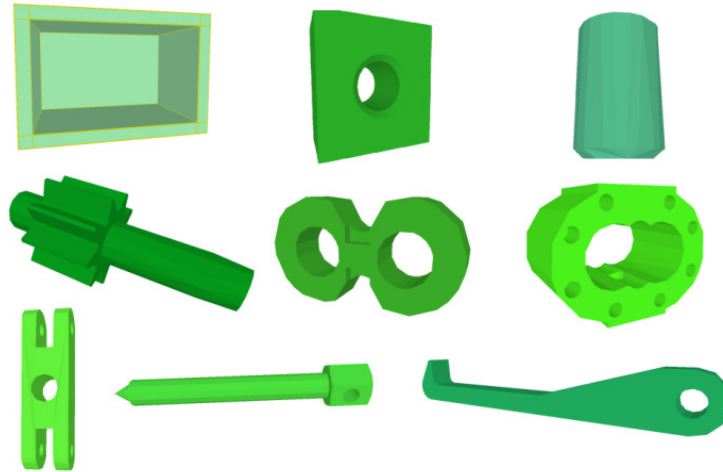


Figura 5.9. Representación física de los objetos utilizando el algoritmo ConvexFT en PhysX.

Los resultados obtenidos mediante esta evaluación muestran que los algoritmos GIMPACT y HACD ofrecen la mejor precisión geométrica de la representación física de objetos no convexos al utilizar el simulador Bullet. Por otro lado, el algoritmo ConvexFT ofrece la mejor precisión geométrica al utilizar el simulador PhysX.

Una manera alternativa de incrementar la precisión geométrica de los objetos, es aumentar el número de triángulos que componen la malla del modelo CAD original. Este incremento resultaría en una detección de colisiones más precisa. Sin embargo, al incrementar la resolución de la malla también aumenta el número de elementos que deben ser verificados por el sistema de detección de colisiones en cada paso de la simulación, resultando en un mayor tiempo de simulación física y por consecuencia una disminución del desempeño general del sistema desarrollado. Por lo tanto, se debe buscar un equilibrio entre la precisión geométrica del modelo y el desempeño del sistema.

### 5.3.2 Evaluación del desempeño

En esta etapa se evalúa la influencia de cada método de representación en el desempeño de tareas de ensamble virtual. Con este fin se realizaron pruebas similares a las llevadas a cabo en la evaluación de los simuladores físicos: caída libre, balanza, empaquetado y casos de ensamble de modelos que representan componentes reales. Los resultados obtenidos se muestran a continuación.

### 5.3.2.1. Influencia sobre el método integrador

Para evaluar la influencia del método de representación sobre el integrador del simulador de física, se llevó a cabo la prueba de caída libre. En esta prueba se utilizaron tres objetos, cada uno de los cuales fue representado por diferentes métodos en Bullet y en PhysX. Cada objeto virtual fue soltado desde una altura de 500 unidades y se midió el tiempo que tarda en alcanzar el suelo. Se realizaron cinco repeticiones del experimento por cada modelo y cada método de representación. Las pruebas se realizaron con los tres módulos gráfico, físico y háptico activos. El tiempo de caída teórico es 0.32 segundos. Los resultados de la prueba de caída libre virtual se muestran en la Tabla 5.10.

Tabla 5.10. Resultados de la prueba de caída libre respecto al método de representación.

| Modelo  | Representación  | Bullet              |        | PhysX               |        |
|---------|-----------------|---------------------|--------|---------------------|--------|
|         |                 | Tiempo de caída (s) | DE (s) | Tiempo de caída (s) | DE (s) |
| Caja    | Primitivo (box) | 1.00                | 0.004  | 5.63                | 0.018  |
|         | GIMPACT         | 0.99                | 0.004  | n/a                 | n/a    |
|         | ConvexFT        | n/a                 | n/a    | 2.11                | 0.004  |
| Perno   | Primitivo       | 1.00                | 0.006  | 5.61                | 0.008  |
|         | GIMPACT         | 1.01                | 0.001  | n/a                 | n/a    |
|         | ConvexFT        | n/a                 | n/a    | 2.11                | 0.006  |
| Engrane | GIMPACT         | 1.45                | 0.049  | n/a                 | n/a    |
|         | HACD            | 1.43                | 0.032  | 2.12                | 0.019  |
|         | ACD             | n/a                 | n/a    | 2.14                | 0.017  |
|         | ConvexFT        | n/a                 | n/a    | 2.12                | 0.03   |

Estos resultados muestran que al utilizar el simulador Bullet, los tiempos de caída libre no se ven afectados por el método de representación utilizado; pues el tiempo de caída es muy similar entre las diferentes representaciones de un mismo objeto. Por ejemplo, al representar el engrane por medio de GIMPACT el tiempo es de 1.45 segundos, mientras que al representarlo por medio de HACD se obtienen 1.43 segundos. La diferencia entre estos dos tiempos es menor a la desviación estándar de cada repetición. Sin embargo, la complejidad geométrica del modelo sí influye en el integrador, incrementando el tiempo de caída.

En el caso de PhysX se observa que al utilizar primitivos el tiempo de caída aumenta drásticamente respecto a los otros métodos de representación. Por ejemplo, en el caso del perno, al utilizar la representación ConvexFT se obtiene un tiempo de caída promedio de 2.11 segundos, mientras que al representarlo como primitivo tarda un promedio de 5.61 segundos en caer. Este efecto no se ha reportado en la literatura revisada ni en la documentación o información disponible del simulador PhysX. Teóricamente al utilizar primitivos el desempeño del simulador debería ser óptimo, pues son objetos más sencillos que cualquier otro tipo de representación. Este efecto puede ser causado porque la configuración del sistema HAMS,



definida por el tamaño del paso de la simulación, número de iteraciones, etc., está optimizada para trabajar con modelos representados por malla triangular y no con primitivos. Por otro lado, en el caso del engrane los resultados obtenidos al utilizar PhysX muestran que el método de representación empleado no afecta el tiempo de caída libre, pues se obtienen tiempos cercanos a 2.12 segundos para cualquiera de los métodos de representación -HACD, ACD o ConvexFT-.

### 5.3.2.2. Precisión de la respuesta a colisiones

Para evaluar la precisión de la respuesta a colisiones de cada método de representación se utilizó la prueba de la balanza. En esta prueba cada lado de la balanza contiene una esfera representada por diferentes métodos de representación. Por ejemplo, el lado derecho de la balanza contiene una esfera representada por primitivos, mientras que el otro extremo contiene una esfera representada por HACD, y de manera similar para cada uno de los métodos de representación.

Los resultados de esta prueba al utilizar Bullet se muestran en la Tabla 5.11. En esta tabla se indica el método de representación utilizado para crear la esfera en cada lado de la balanza y la inclinación de esta en cada una de las repeticiones realizadas. Los resultados de la Tabla 5.11 muestran que la fuerza de reacción producida por la representación por primitivos es ligeramente de menor magnitud que la reacción producida por la representación HACD, pues la balanza tiende a inclinarse hacia el lado de esta última representación. De manera similar, al comparar la representación por primitivos contra GIMPACT se observa que la balanza se tiende a inclinar hacia el lado de la representación por primitivos, por lo tanto, estos tienden a ser ligeramente más pesados, es decir, producen una mayor fuerza de reacción. Se puede concluir que GIMPACT produce una fuerza de reacción más pequeña mientras que HACD produce una fuerza de reacción de mayor magnitud.

Tabla 5.11. Resultados de la prueba de balanza respecto al método de representación, Bullet.

| Representación lado izquierdo | Representación lado derecho | Izquierda+ | Izquierda | Equilibrado | Derecha | Derecha+ |
|-------------------------------|-----------------------------|------------|-----------|-------------|---------|----------|
| Primitivo                     | HACD                        | 0          | 0         | 3           | 2       | 0        |
| Primitivo                     | GIMPACT                     | 2          | 2         | 1           | 0       | 0        |
| HACD                          | GIMPACT                     | 1          | 0         | 4           | 0       | 0        |

Por otro lado, los resultados al utilizar el simulador PhysX se muestran en la Tabla 5.12, en la cual se puede observar que la representación por primitivos es la más ligera de los métodos de representación evaluados, pues al compararlo contra los demás algoritmos la balanza tiende a inclinarse hacia el lado opuesto de esta representación. Por otro lado, los algoritmos HACD y ConvexFT se muestran más pesados que ACD. Además, ConvexFT muestra una mayor fuerza

de reacción que HACD. Por lo tanto, se puede concluir que la representación ConvexFT presenta la mayor fuerza de reacción, seguida de HACD, ACD y finalmente la representación por primitivos, la cual presenta la fuerza de reacción más pequeña.

Tabla 5.12. Resultados de la prueba de balanza respecto al método de representación, PhysX.

| Representación lado izquierdo | Representación lado derecho | Izquierda+ | Izquierda | Equilibrado | Derecha | Derecha+ |
|-------------------------------|-----------------------------|------------|-----------|-------------|---------|----------|
| Primitivo                     | HACD                        | 0          | 0         | 2           | 3       | 0        |
| Primitivo                     | ACD                         | 0          | 0         | 1           | 3       | 2        |
| Primitivo                     | ConvexFT                    | 0          | 0         | 1           | 1       | 3        |
| ConvexFT                      | HACD                        | 4          | 0         | 1           | 0       | 0        |
| HACD                          | ACD                         | 5          | 0         | 0           | 0       | 0        |
| ConvexFT                      | ACD                         | 3          | 1         | 1           | 0       | 0        |

Cabe mencionar que en estas pruebas la posición de las esferas fue intercambiada de derecha a izquierda y el resultado expresado en orden invertido. Lo anterior con el fin de verificar el correcto funcionamiento de la balanza y validar los resultados obtenidos.

5.3.2.3. Respuesta a contactos acumulativos en múltiples direcciones

Con el fin de evaluar la respuesta a colisiones de los diversos métodos de representación física cuando ocurren contactos acumulativos en diversas direcciones, se realiza la prueba de empaquetado. El TFT y TSF son los parámetros medidos para evaluar el desempeño de cada método de representación. En esta prueba se realiza el ensamble utilizando cajas con diferentes medidas, lo que da lugar a tres casos de tolerancias al realizar el ensamble: 1.5mm, 1.2mm y 0mm.

En la Figura 5.10 se muestran los resultados con respecto al TFT de esta prueba. En la primera sección de la gráfica (lado izquierdo) se presentan los resultados correspondientes al simulador PhysX y cada uno de sus métodos de representación: primitivos (Px-Prim), HACD (Px-HACD), ACD (Px-ACD) y ConvexFT (Px-CFT) en ese mismo orden. En la segunda parte de la gráfica (lado derecho) se presentan los valores TFT al utilizar el simulador Bullet y sus métodos de representación: primitivos (Bt-Prim), HACD (Bt-HACD) y GIMPACT (Bt-GIM). En esta gráfica se incluyen los valores de TFT obtenidos para cada método de representación y para cada caso de tolerancia. En algunas configuraciones el ensamble no pudo llevarse a cabo con todas las tolerancias, en estos casos la barra del resultado del TFT correspondiente a dicho caso de tolerancia no aparece en la gráfica.

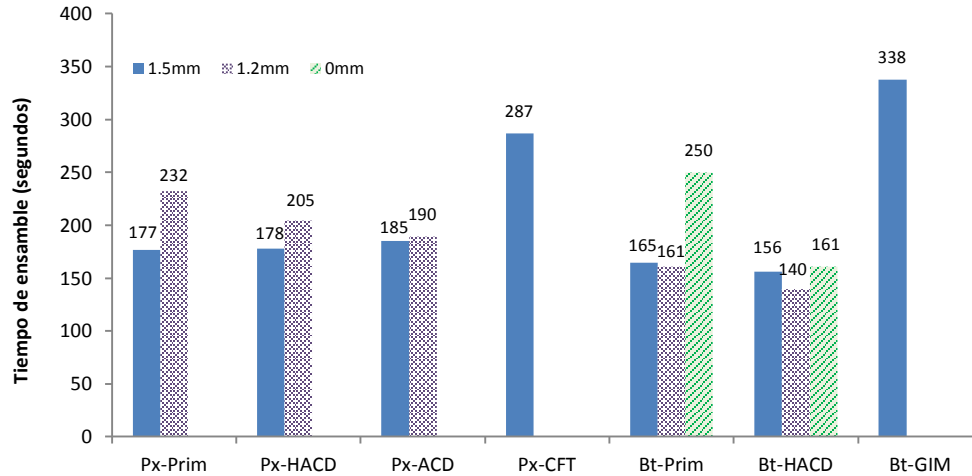


Figura 5.10. TFF prueba de empaquetado, evaluación de métodos de representación.

De los resultados obtenidos se puede observar que el menor TFF corresponde a la configuración Bullet HACD con un tiempo de 140 segundos y una desviación estándar de 5.6s para el caso donde la tolerancia es de 1.2mm. Por otro lado, el mayor tiempo de ensamble corresponde a la configuración Bullet-GIMPACT con 338 segundos, seguido por la configuración PhysX-ConvexFT, con 287 segundos, cada uno con una desviación estándar de 27.6s y 82.7s respectivamente. El incremento en el TFF al utilizar estas últimas configuraciones se debe a que existe inestabilidad y saltos repentinos de las cajas insertadas en el contenedor, haciendo más difícil su alineación y ensamble en la posición final. También se puede observar que el ensamble con tolerancia de 1.2 mm no pudo ser llevado a cabo con las configuraciones PhysX-ConvexFT y Bullet-GIMPACT. Además, el ensamble sin tolerancias (0mm) únicamente pudo ser realizado mediante las configuraciones Bullet-Primitivos y Bullet-HACD.

Además, también se observó cualitativamente que las representaciones por primitivos, HACD y ACD muestran una detección de colisiones precisa en ambos simuladores, es decir, no se observó traslape entre los objetos virtuales. La configuración Bullet-primitivos mostró además la mejor respuesta a colisiones, es decir, se observó una fuerza de reacción y comportamiento dinámico realistas. Finalmente las configuraciones PhysX con primitivos, HACD y ACD mostraron la mejor estabilidad del ensamble, es decir, los objetos se mantienen estáticos en su posición después de ser ensamblados.

Para cada configuración evaluada en esta prueba se midió también el TSF. En la Figura 5.11 se muestran los valores del TSF por cada configuración, en la cual se puede observar que el simulador PhysX muestra un menor tiempo de simulación, alrededor de 1.1 milisegundos por ciclo, excepto en la configuración PhysX- ConvexFT donde el tiempo de simulación aumenta hasta 4.2 milisegundos. Por otro lado, al utilizar Bullet se puede observar que el tiempo de simulación aumenta conforme disminuye la tolerancia. Al utilizar el método GIMPACT se obtiene el menor tiempo de simulación en Bullet, 1.5 milisegundos; sin embargo esta

configuración no permite el ensamble con bajas tolerancias. Finalmente, se observa que los valores obtenidos para el TSF son adecuados para un buen desempeño del sistema, pues de acuerdo a la literatura, cada actualización de la simulación física debe tardar menos de 40 milisegundos (Laurell, 2008 [106]).

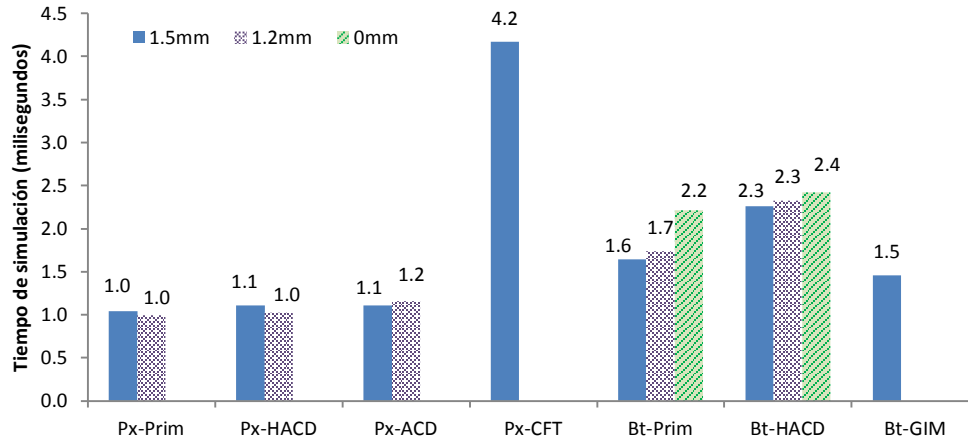


Figura 5.11. TSF prueba de empaquetado, evaluación de métodos de representación.

De manera similar, se midió el tiempo que tarda la simulación háptica en renderizar los objetos virtuales, Figura 5.12. De los resultados se observó que al utilizar el simulador PhysX el tiempo de simulación háptica (TSH) es alrededor de 2.8 milisegundos para los métodos de representación por primitivos, HACD y ACD, pero aumenta hasta 3.6 milisegundos al utilizar la representación ConvexFT. Por otro lado, al utilizar el simulador Bullet el TSH es alrededor de 2.7 milisegundos para cualquier método de representación y cualquier caso de tolerancia, siendo el más bajo al utilizar GIMPACT con 2.6 milisegundos.

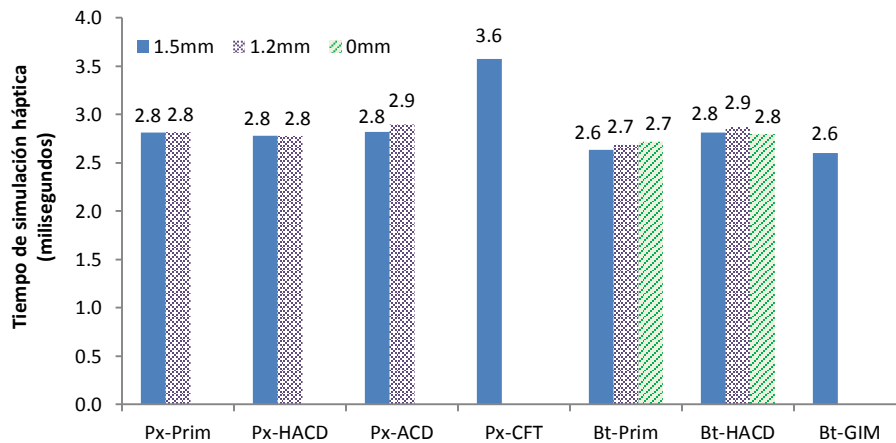


Figura 5.12. TSH prueba de empaquetado, evaluación de métodos de representación.

### 5.3.2.4. Desempeño en tareas de ensamble virtual háptico

Para esta prueba se realizó el ensamble de cuatro componentes distintos: un extractor de poleas, una prensa de banco, una bomba de engranes y un cilindro neumático, permitiendo evaluar el desempeño de cada método de representación en tareas de ensamble virtual que involucran objetos reales. En estas pruebas se utilizaron los métodos de representación HACD, GIMPACT y ConvexFT, los cuales obtuvieron los mejores resultados en la evaluación de precisión geométrica y en las pruebas anteriores. También se incluye el método de representación mixta, el cual combina la representación por malla triangular y la descomposición en elementos convexos en un mismo objeto dependiendo del estado cinemático de éste. Para realizar los ensambles se utilizaron los tres simuladores de física disponibles en el sistema HAMS, Bullet v2.82, PhysX v2.8 y PhysX v3.1.

#### *Ensamble del extractor de poleas*

Los resultados del ensamble del extractor de poleas se muestran en la Tabla 5.13, en la cual se muestra el simulador físico utilizado y el método de representación. Los resultados obtenidos se muestran en términos del TFT en segundos, la fuerza de retroalimentación háptica promedio en Newtons y el TSF en milisegundos. De los resultados se puede observar que los tiempos TFT y TSF son los más bajos al utilizar el simulador PhysX v3.1 y la representación mixta. Asimismo, el tiempo TSF es menor cuando se utiliza el modo de representación mixta comparado contra los otros métodos de representación empleando cualquier simulador físico – Bullet, PhysX v2.8 o PhysX v3.1–.

Tabla 5.13. Resultados del ensamble del extractor de poleas.

| Simulador  | Modo de representación          | TFT (s)   | Fuerza media (N) | TSF (ms)   |
|------------|---------------------------------|-----------|------------------|------------|
| Bullet     | GIMPACT                         | 85        | 0.6              | 5.3        |
|            | HACD                            | 191       | 0.95             | 10.1       |
|            | GIMPACT & HACD (mixta)          | 129       | <b>1.14</b>      | <b>1.7</b> |
| PhysX v2.8 | ConvexFT                        | 229       | 0.62             | 4.4        |
|            | HACD                            |           | No completado    |            |
|            | Malla triangular & HACD (mixta) | 82        | 0.37             | <b>0.7</b> |
| PhysX v3.1 | ConvexFT                        | 110       | 0.71             | 8.0        |
|            | HACD                            |           | No completado    |            |
|            | Malla triangular & HACD (mixta) | <b>72</b> | 0.51             | <b>0.9</b> |

El ensamble no pudo ser completado al utilizar el algoritmo HACD en cualquiera de las dos versiones de PhysX, esto debido a que los resultados producidos por la descomposición no son completamente compatibles con la creación de elementos convexos en este simulador, lo

que resulta en una representación física con baja aproximación geométrica la cual no puede ser ensamblada. Finalmente, al utilizar la representación mixta en el simulador Bullet se observa una mayor fuerza de retroalimentación promedio, lo cual indica que el usuario puede percibir con más claridad la fuerza de reacción a las colisiones. Sin embargo, también puede indicar una respuesta excesiva a los contactos, lo cual dificulta el proceso de ensamble y se ve reflejado en el TFT.

#### *Ensamble de la bomba de engranes*

Para el caso de la bomba de engranes los resultados se muestran en la Tabla 5.14. De manera similar al caso anterior, el menor tiempo de ensamble se obtuvo al utilizar el simulador PhysX v3.1 y la representación mixta. También se observa un TSF más pequeño al utilizar la representación mixta, independientemente simulador físico. Respecto a la fuerza de retroalimentación promedio, se observa un comportamiento similar al ensamble del extractor de poleas, pues el valor más alto de fuerza se obtiene al utilizar el simulador Bullet y los métodos de representación mixta y HACD.

Tabla 5.14. Resultados del ensamble de la bomba de engranes.

| Simulador  | Modo de representación          | TFT (s)   | Fuerza media (N) | TSF (ms)   |
|------------|---------------------------------|-----------|------------------|------------|
| Bullet     | GIMPACT                         | 53        | 0.43             | 5.4        |
|            | HACD                            | 126       | <b>0.86</b>      | 19.1       |
|            | GIMPACT & HACD (mixta)          | 85        | <b>0.81</b>      | <b>1.8</b> |
| PhysX v2.8 | ConvexFT                        | 77        | 0.43             | 8.6        |
|            | HACD                            |           | No completado    |            |
|            | Malla triangular & HACD (mixta) | 53        | 0.33             | <b>1.7</b> |
| PhysX v3.1 | ConvexFT                        | 94        | 0.52             | 19.6       |
|            | HACD                            |           | No completado    |            |
|            | Malla triangular & HACD (mixta) | <b>51</b> | 0.48             | <b>1.7</b> |

#### *Ensamble de cilindro neumático*

Los resultados obtenidos para el ensamble del cilindro neumático se muestran en la Tabla 5.15. De la misma manera que en los casos anteriores, el menor tiempo de ensamble se observa al utilizar la representación mixta y PhysX v3.1. Asimismo, el TSF es reducido drásticamente al utilizar la representación mixta con cualquier simulador físico. La mayor fuerza de retroalimentación háptica se obtiene al utilizar el algoritmo HACD y el simulador Bullet.

Tabla 5.15. Resultados del ensamble del cilindro neumático.

| Simulador  | Modo de representación          | TFT (s)    | Fuerza media (N) | TSF (ms)   |
|------------|---------------------------------|------------|------------------|------------|
| Bullet     | GIMPACT                         | 296        | 0.62             | 10.0       |
|            | HACD                            | 363        | 1.1              | 29.5       |
|            | GIMPACT & HACD (mixta)          | <b>154</b> | 0.64             | <b>2.0</b> |
| PhysX v2.8 | ConvexFT                        | 678        | 0.39             | 14.7       |
|            | HACD                            |            | No completado    |            |
|            | Malla triangular & HACD (mixta) | 219        | 0.35             | <b>2.3</b> |
| PhysX v3.1 | ConvexFT                        | 309        | 0.54             | 31.2       |
|            | HACD                            |            | No completado    |            |
|            | Malla triangular & HACD (mixta) | <b>124</b> | 0.56             | <b>2.0</b> |

*Ensamble de tornillo de banco*

Los resultados del ensamble del tornillo de banco se muestran en la Tabla 5.16, los cuales confirman las observaciones anteriores. Al utilizar PhysX v3.1 y la representación mixta se obtiene un menor TFT. A diferencia de los casos anteriores, al utilizar HACD y Bullet el ensamble no pudo ser completado debido a que la respuesta a colisiones fue muy fuerte, imposibilitando ensamblar las dos mordazas.

Tabla 5.16. Resultados del ensamble de la prensa de banco.

| Simulador  | Modo de representación          | TFT (s)   | Fuerza media (N) | TSF (ms)   |
|------------|---------------------------------|-----------|------------------|------------|
| Bullet     | GIMPACT                         | 126       | 0.63             | 4.0        |
|            | HACD                            |           | No completado    |            |
|            | GIMPACT & HACD (mixta)          | 97        | 0.94             | <b>1.5</b> |
| PhysX v2.8 | ConvexFT                        |           | No completado    |            |
|            | HACD                            |           | No completado    |            |
|            | Malla triangular & HACD (mixta) | 78        | 0.57             | <b>0.9</b> |
| PhysX v3.1 | ConvexFT                        | 245       | 0.92             | 17.4       |
|            | HACD                            |           | No completado    |            |
|            | Malla triangular & HACD (mixta) | <b>72</b> | 0.64             | <b>1.0</b> |

De los resultados obtenidos en las pruebas anteriores, se pueden establecer las siguientes conclusiones respecto al desempeño de los métodos de representación en tareas de ensamble virtual:

1. El método de representación mixta ofrece una mejora sustancial en el desempeño de la simulación al disminuir el tiempo de actualización de la simulación física, sin importar el simulador utilizado.
2. La configuración de método de representación mixta y simulador PhysX v3.1 ofrecen el mejor desempeño en términos de rapidez de realización del ensamble virtual.
3. La configuración Bullet-GIMPACT también ofrece buenos resultados en términos del tiempo de finalización de la tarea, tiempo de simulación y fuerza de retroalimentación.
4. El uso de la configuración Bullet-HACD resulta en una respuesta a colisiones de mayor magnitud, lo que puede llegar a dificultar e incluso imposibilitar el proceso de ensamble.

#### 5.4 Resultados del análisis y evaluación del efecto del peso en tareas de ensamble

Otro de los factores que pueden afectar el desempeño del proceso de ensamble virtual es la forma en que el usuario percibe el peso de los objetos virtuales. Para cuantificar esta influencia se realizó la evaluación del efecto del peso que consiste en realizar el ensamble de la bomba de engranes utilizando diferentes niveles de pesos para cada una de las partes, Tabla 4.4.

Se evalúa el efecto del peso al observar el comportamiento de dos factores: 1) la trayectoria de ensamble y 2) el TFT. La trayectoria de ensamble se observa cualitativamente mediante los crono-ciclos, descritos en la sección 2.6.5 del capítulo II. Una vez que el ensamble ha sido completado se pueden mostrar los crono-ciclos y por medio de estos observar las trayectorias seguidas por el objeto manipulado. Para realizar esta evaluación se realizaron cuatro repeticiones por cada nivel de peso y cada simulador físico. Además, también se incluyó la evaluación del efecto de realizar el ensamble con una o con dos manos. En la Figura 5.13a se muestran las trayectorias de ensamble de cada componente de la bomba para el nivel de peso N1. En la Figura 5.13b se observan las trayectorias de ensamble para el nivel de peso N8. Las esferas de color rojo representan la trayectoria de ensamble de cada objeto virtual cuando es manipulado mediante el dispositivo háptico. La distancia entre cada una de las esferas representa la rapidez del movimiento, cuando la rapidez es más baja las esferas se encuentran más juntas.

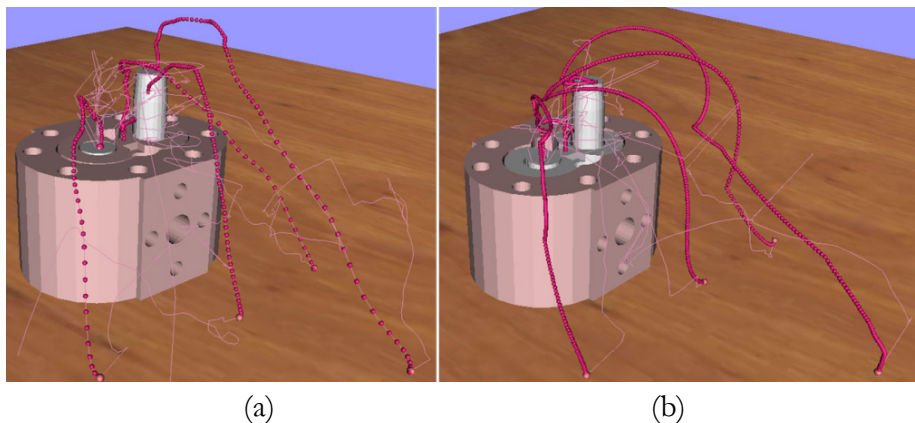


Figura 5.13. Crono-ciclos para la evaluación del efecto del peso: (a) nivel de peso N1, (b) nivel de peso N2.



En la Figura 5.13a se puede observar que al comenzar a manipular un objeto con el fin de llevarlo a una ubicación cercana a su posición de ensamble, el movimiento se realiza con una alta rapidez. Sin embargo conforme el objeto se acerca a la posición de ensamble esta rapidez disminuye. La rapidez de manipulación también disminuye cuando el peso de los objetos aumenta, lo cual se verifica por medio de los resultados obtenidos para el nivel de peso N8, Figura 5.13b.

Mediante la comparación de los resultados obtenidos para los diferentes niveles de peso, por ejemplo los mostrados en la Figura 5.13, es posible observar que los objetos más pesados poseen mayor inercia y por lo tanto resulta más difícil cambiar su estado de movimiento. Debido a esto los movimientos correspondientes al ensamble de objetos con mayor peso poseen una menor velocidad y siguen trayectorias más redondeadas. Por lo tanto, se puede decir que la inercia de los objetos sí influye en el proceso de ensamble virtual; a mayor inercia mayor dificultad para manipular un objeto, justo como ocurre en el mundo real.

Por otro lado, también se evaluó el efecto del peso de los objetos en el TFT o tiempo de ensamble. En la Figura 5.14 se muestran los resultados obtenidos al realizar el ensamble de la bomba de engranes utilizando una mano. En esta gráfica se muestra el TFT promedio en segundos para cada uno de los niveles de peso y para cada simulador físico, así como el tiempo de ensamble utilizando objetos reales, el cual es de 37 segundos.

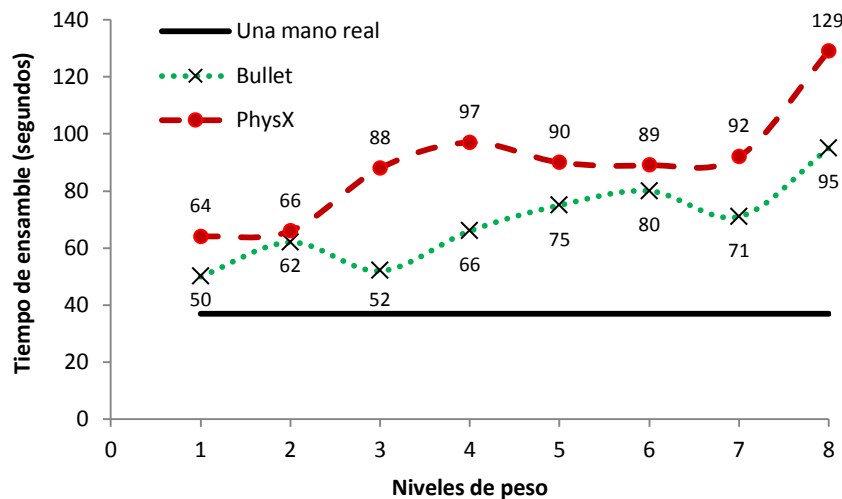


Figura 5.14. Tiempo de finalización de la tarea para cada nivel de peso utilizando una mano.

Mediante estos resultados es posible observar que el menor tiempo de ensamble al utilizar solo una mano corresponde al nivel de peso N1, donde el peso de cada uno de los objetos es mínimo, igual a 0.02 N, solo lo necesario para mantener un comportamiento dinámico estable de los objetos manipulados. Al utilizar el simulador Bullet el tiempo de ensamble promedio mínimo es de 50 segundos, mientras que al utilizar PhysX el tiempo promedio mínimo es de

64 segundos. Cabe mencionar que en el caso del ensamble real el peso de los objetos no fue variable.

El mismo caso de ensamble también fue llevado a cabo bajo las mismas condiciones pero ahora haciendo uso de las dos manos para manipular los objetos virtuales. En la Figura 5.15 se muestran los resultados de esta configuración. En la experimentación virtual, el menor tiempo de finalización de la tarea corresponde también al caso donde el peso de los objetos es mínimo. Al utilizar el simulador Bullet el TFT correspondiente al nivel de peso N1 es de 52 segundos, mientras que al utilizar el simulador PhysX el TFT mínimo es de 75 segundos. El ensamble real utilizando las dos manos tomó un TFT promedio de 27 segundos. Al realizar el ensamble con el nivel de peso N6 se puede observar una caída en la tendencia del TFT para ambos simuladores Bullet y PhysX, esto puede ser debido a una compensación en la rigidez del dispositivo háptico, pues en el nivel N6 el peso de cada una de las partes del ensamble sobrepasa los 0.88 N, lo cual es la fuerza nominal que puede soportar de manera continua el dispositivo háptico. Este valor no debe ser confundido con el valor de fuerza máxima que puede soportar el dispositivo, 3.3N.

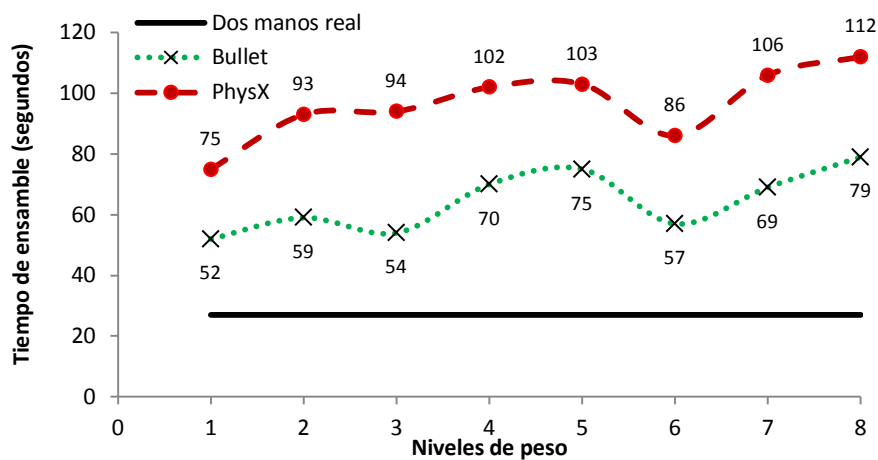


Figura 5.15. Tiempo de finalización de la tarea para cada nivel de peso utilizando dos manos.

Es importante mencionar que el procedimiento para realizar el ensamble de la bomba utilizando las dos manos en el mundo real es muy similar al procedimiento empleado en el ambiente virtual. En ambos casos las dos manos fueron utilizadas principalmente para alinear y ensamblar los dos engranes en la carcasa y en el cojinete inferior. Sin embargo, se puede observar que el proceso de ensamble real tomó menos tiempo al utilizar dos manos que al utilizar una sola, mientras que en el proceso virtual esta condición se invierte, es decir, realizar el ensamble con una mano fue más rápido que realizarlo con dos manos. Esta diferencia puede deberse a que la respuesta a colisiones se vuelve más lenta al realizar un segundo ciclo de renderizado háptico, lo que ocasiona que esporádicamente los objetos manipulados se sientan pegajosos, con mucha fricción o con respuestas exageradas al tener contacto con otros objetos,

lo que resulta en una mayor dificultad para alinear y ensamblar dos objetos, como es el caso de los engranes.

De manera general, se puede concluir que el tiempo de ensamble real es aún menor que el TFT virtual, incluso al modelar los objetos manipulados con un peso mínimo. La percepción del peso afecta al tiempo de finalización de la tarea de manera directa, es decir, al aumentar el peso de los objetos virtuales incrementa el TFT. Al aumentar el peso de los objetos virtuales su inercia incrementa, dificultando su manipulación. Se puede observar que el simulador físico utilizado también afecta el desempeño del proceso de ensamble virtual.

### 5.5 Evaluación de restricciones de ensamble dinámicas

Las restricciones de ensamble dinámicas, DACs, fueron desarrolladas e implementadas en el sistema HAMS con el fin de guiar al usuario en el proceso de ensamble y así reducir la carga computacional del sistema de detección de colisiones. Para evaluar la influencia de estas restricciones en el proceso de ensamble virtual, se consideraron tres ensambles virtuales: el extractor de poleas, la bomba de engranes y el cilindro neumático. Estos ensambles se realizaron primeramente sin utilizar las DACs, es decir, solo el modelado basado en física PBM. Posteriormente se repitieron las pruebas de ensamble pero ahora utilizando también las DACs, es decir combinando el PBM y el CBM. Para cada prueba se midió el TFT, el TSF y la fuerza de retroalimentación promedio de cada pieza por separado.

#### 5.5.1 Ensamble del extractor de poleas

El proceso de ensamble del extractor de poleas utilizando únicamente el PBM se muestra en la Figura 5.16a, mientras que en la Figura 5.16b se muestra el ensamble utilizando las DACs. El ensamble final se muestra en la Figura 5.16c, mientras que la Figura 5.16d muestra el ensamble real utilizando piezas creadas por prototipado rápido.

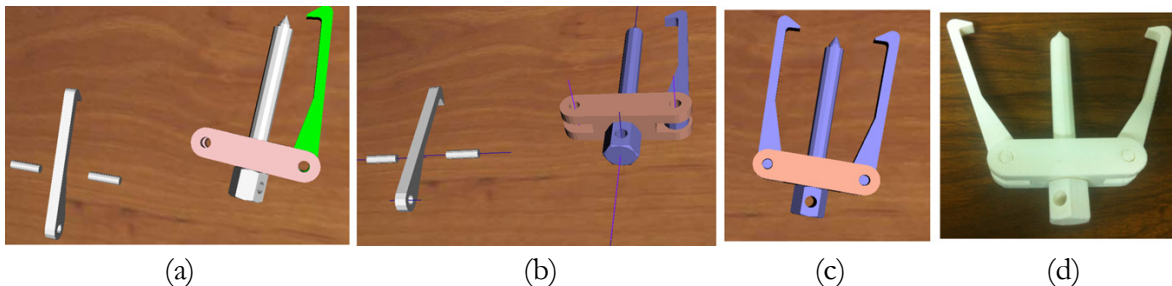


Figura 5.16. Caso de ensamble del extractor de poleas: (a) ensamble utilizando únicamente PBM, (b) ensamble utilizando DACs, (c) ensamble final, (d) ensamble real.

Los resultados para el ensamble del extractor de poleas se presentan en la Tabla 5.17, donde se puede observar el TFT sin utilizar DACs, el TFT utilizando DACs, y la fuerza de retroalimentación media para el ensamble de cada parte. De estos resultados se puede observar

que cuando el ensamble es realizado utilizando las DACs, los tiempos de ensamble individual y total son menores y más cercanos al tiempo de ensamble real, el cual es de 18 segundos. Por otro lado, la fuerza de retroalimentación es similar para ambos casos de modelado PBM y DACs.

Tabla 5.17. Resultados del ensamble del extractor de poleas, con restricciones de ensamble.

| Pieza              | TFT PBM (s) | TFT DACs (s) | Fuerza Media PBM (N) | Fuerza Media DACs (N) |
|--------------------|-------------|--------------|----------------------|-----------------------|
| Base del extractor | 4           | 1            | 0.29                 | 0.36                  |
| Tornillo           | 18          | 6            | 0.54                 | 0.49                  |
| Brazo derecho      | 14          | 6            | 0.50                 | 0.56                  |
| Perno derecho      | 12          | 2            | 0.56                 | 0.66                  |
| Brazo izquierdo    | 10          | 5            | 0.64                 | 0.38                  |
| Perno izquierdo    | 10          | 2            | 0.70                 | 0.65                  |
| Total              | 77          | 32           | 0.54                 | 0.52                  |

### 5.5.2 Ensamble de la bomba de engranes

La Figura 5.17a representa el ensamble de la bomba de engranes utilizando únicamente el PBM, mientras que la Figura 5.17b presenta el ensamble utilizando las DACs. La Figura 5.17c muestra el ensamble final mientras que el ensamble real puede ser observado en la Figura 5.17d.

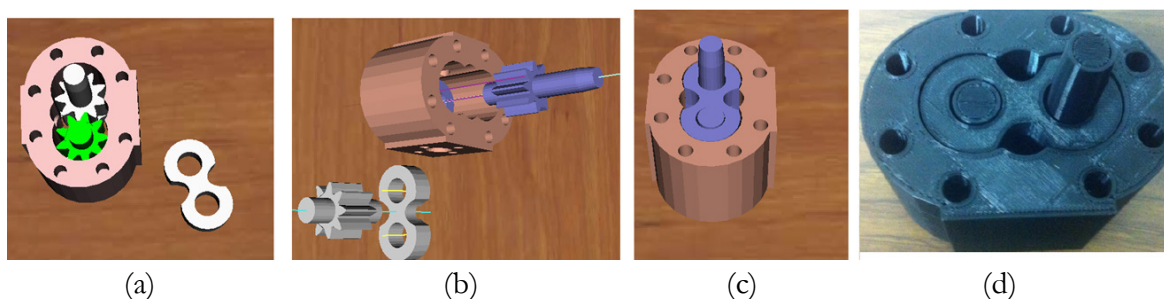


Figura 5.17. Bomba de engranes: (a) ensamble utilizando únicamente PBM, (b) ensamble utilizando PBM y DACs, (c) ensamble final, (d) ensamble real.

En la Tabla 5.18 se presentan los resultados para el ensamble de la bomba de engranes. De manera similar al caso anterior, se puede observar que cuando se utilizan las DACs, los tiempos de ensamble son menores y más cercanos al tiempo de ensamble real, de 13 segundos. Sin embargo, la fuerza de retroalimentación háptica es menor. Lo que puede ser un indicativo de que la respuesta a colisiones es más estable, pero puede disminuir la percepción de colisiones a través del dispositivo háptico.

Tabla 5.18. Resultados del ensamble de la bomba de engranes, con restricciones de ensamble.

| Pieza             | TFT PBM (s) | TFT DACs (s) | Fuerza Media PBM (N) | Fuerza Media DACs (N) |
|-------------------|-------------|--------------|----------------------|-----------------------|
| Carcaza           | 5           | 2            | 0.27                 | 0.25                  |
| Cojinete inferior | 15          | 8            | 0.68                 | 0.32                  |
| Engrane largo     | 14          | 4            | 0.62                 | 0.47                  |
| Engrane corto     | 31          | 8            | 0.78                 | 0.34                  |
| Cojinete superior | 11          | 10           | 0.88                 | 0.35                  |
| Total             | 89          | 42           | 0.65                 | 0.35                  |

El impacto de las restricciones de ensamble sobre el desempeño de la simulación también fue evaluado para este caso de estudio. En la Figura 5.18 se muestra la tendencia de los tiempos que tarda cada módulo del sistema HAMS en completar la simulación durante el proceso de ensamble de la bomba de engranes.

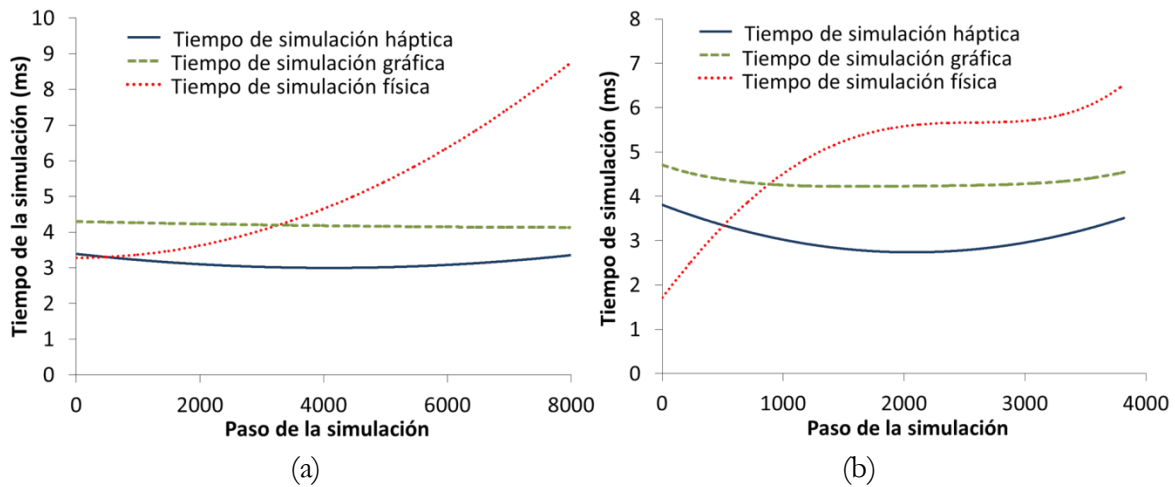


Figura 5.18. Tiempos de simulación en el ensamble de la bomba de engranes: (a) utilizando únicamente PBM, (b) utilizando PBM y DACs.

Se puede observar que los tiempos de simulación gráfica y háptica tienden a permanecer relativamente constantes durante todo el proceso de ensamble. Sin embargo, el tiempo de simulación física tiende a incrementar conforme se va realizando el ensamble, esto se debe a que al ir ensamblando las piezas se genera un mayor número de puntos de contacto que deben ser resueltos en cada paso de la simulación física. Sin embargo, se puede observar que cuando se utilizan las DACs el tiempo de simulación física es menor.

### 5.5.3 Ensamble del cilindro neumático

La Figura 5.19a muestra el ensamble utilizando únicamente el PBM, mientras que la Figura 5.19b representa el ensamble utilizando las DACs. La Figura 5.19c muestra el ensamble final mientras que el ensamble real se presenta en la Figura 5.19d.

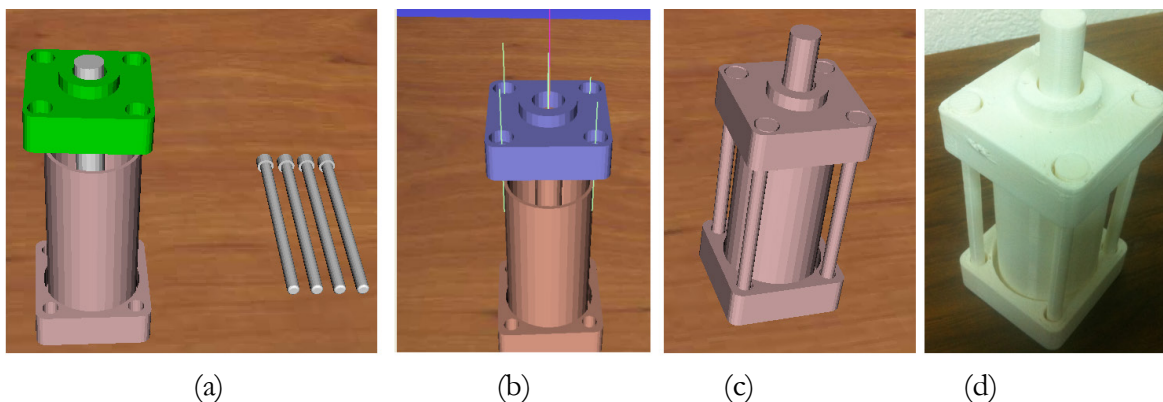


Figura 5.19. Ensamble de cilindro neumático: (a) ensamble utilizando únicamente PBM, (b) ensamble utilizando DACs, (c) ensamble final, (d) ensamble real.

En la Tabla 5.19 se observan los resultados para el ensamble del cilindro neumático, al igual que en los casos anteriores, cuando se utilizan las DACs los tiempos de ensamble son menores y más cercanos al tiempo de ensamble real, de 24 segundos. En cuanto a la fuerza de retroalimentación háptica no se observó alguna variación significativa al utilizar DACs.

Tabla 5.19. Resultados del ensamble del cilindro neumático, con restricciones de ensamble.

| Pieza          | TFT PBM (s) | TFT DACs (s) | Fuerza Media PBM (N) | Fuerza Media DACs (N) |
|----------------|-------------|--------------|----------------------|-----------------------|
| Tapa trasera   | 3           | 1            | 0.46                 | 0.66                  |
| Cilindro       | 10          | 8            | 0.49                 | 0.33                  |
| Embolo         | 12          | 5            | 0.53                 | 0.44                  |
| Tapa delantera | 31          | 8            | 0.70                 | 0.44                  |
| Tornillo 1     | 23          | 9            | 0.33                 | 0.45                  |
| Tornillo 2     | 18          | 8            | 0.40                 | 0.46                  |
| Tornillo 3     | 47          | 10           | 0.37                 | 0.48                  |
| Tornillo 4     | 19          | 7            | 0.34                 | 0.45                  |
| Total          | 191         | 80           | 0.45                 | 0.46                  |

### 5.6 Resultados de la evaluación de la percepción del usuario sobre el sistema HAMS

Con el fin de evaluar el desempeño general del sistema HAMS, se pidió a un grupo de participantes realizar el ensamble de la bomba de engranes en el sistema. Cada participante tuvo una sesión previa de entrenamiento y después realizó el ensamble seis veces: tres utilizando únicamente el PBM y las otras tres utilizando el PBM y las DACs. Durante el ensamble se midió el TFT y se pidió a cada usuario contestar dos cuestionarios a cerca de su percepción y experiencia en el uso del sistema HAMS. Al utilizar el PBM se obtuvo un TFT promedio para todos los usuarios de 209 segundos, con una desviación estándar de 58 segundos. Al utilizar las DACs el TFT promedio se redujo a 141 segundos y una desviación estándar de 31 segundos. Con lo cual se confirma el beneficio de utilizar las restricciones de ensambles dinámicas.

En la Figura 5.20 se muestran los resultados de la evaluación cualitativa al sistema HAMS, donde se muestran los valores que los usuarios asignan a cada parámetro cualitativamente. Estos valores se encuentran normalizados, es decir, su máximo valor es 1.0, esto con el fin de mantener uniformidad en la presentación de los resultados, pues algunos de los parámetros fueron evaluados con escalas diferentes.

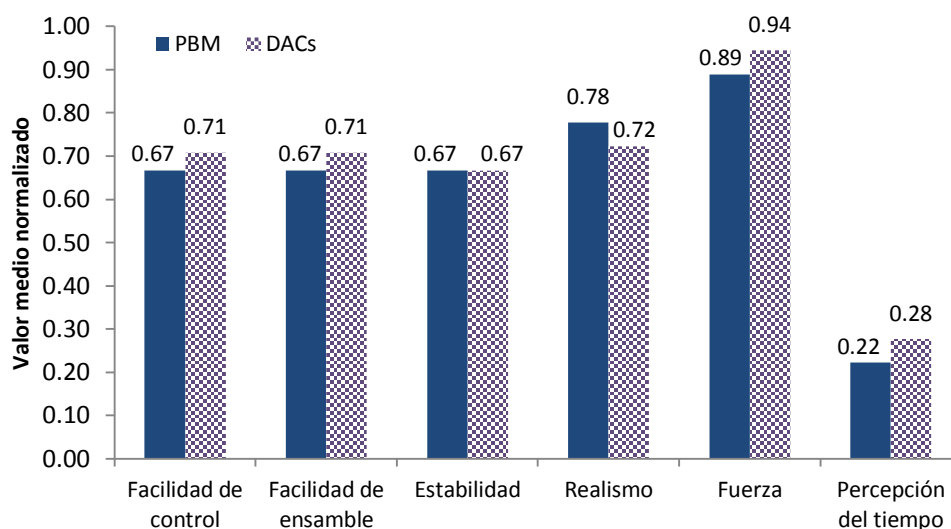


Figura 5.20. Resultados de las pruebas con varios usuarios en el sistema HAMS.

De estos resultados se puede observar que el parámetro mejor evaluado por los usuarios es la retroalimentación de fuerza. Además, los usuarios percibieron que el tiempo de ensamble virtual es mayor al tiempo de ensamble real, lo cual concuerda con los resultados cuantitativos de la experimentación realizada. Los demás parámetros muestran valores satisfactorios, por encima de 0.5, lo que indica un buen desempeño de la plataforma de ensambles virtuales. Se observa una mejor percepción general del proceso de ensamble cuando se utilizan restricciones de ensamble excepto en el parámetro de realismo, lo cual es la desventaja de este método.

Con respecto a la percepción global del sistema, los usuarios comentaron lo siguiente:

1. Consideran la experiencia de ensamble virtual como útil e interesante.
2. Consideran que la precisión del sistema es buena, es decir, perciben que la mayoría del tiempo los objetos se comportan como objetos reales.
3. Sienten confianza al utilizar el sistema, es decir, intuyen fácilmente el procedimiento para manipular los objetos y realizar el ensamble.
4. Consideran que el sistema reproduce de manera correcta los movimientos hechos por la mano.
5. Consideran que, entre diversos parámetros, la rapidez del sistema es el principal factor para poder utilizarlo en casos reales.
6. No consideran relevante que el tiempo de ensamble virtual deba ser igual al tiempo de ensamble real para que el sistema pueda tener una aplicación práctica.
7. Están de acuerdo en que el sistema puede ser utilizado en planeaciones o análisis reales de ensamble.
8. Entre las ventajas del sistema menciona que el sistema es intuitivo y amigable, además ofrece una simulación realista del proceso de ensamble. Entre las desventajas se menciona la dificultad para manipular la orientación de la cámara y obtener un punto de vista diferente. Además de que se requiere de práctica para poder realizar los ensambles con mayor fluidez. Puede ocasionar fatiga.
9. Consideran que el sistema puede ser empleado como una plataforma de diseño, planeación y entrenamiento en operaciones riesgosas, repetitivas o donde se requiera alta precisión. Así como en casos de rehabilitación.
10. Finalmente los usuarios sugieren mejorar la rapidez de la detección de colisiones y agregar la manipulación de los objetos por medio de la interacción completa de la mano y los dedos. Además mejorar la simulación de materiales para observar deformaciones y posibles daños al material cuando se aplican fuerzas excesivas.

### **5.7 Comparativa del sistema HAMS contra plataformas de ensamble similares**

Una de las principales características que identifica a cualquier sistema de ensambles virtuales es la técnica utilizada para el modelado del ensamble, PBM o CBM. El sistema HAMS utiliza el enfoque híbrido, el cual combina ambas técnicas de modelado. Por lo tanto, para realizar la comparativa se han seleccionado tres sistemas que también utilizan el enfoque híbrido: SHARP, HVAS e HIDRA. Además se agrega a la comparativa el sistema de ensambles HIVEx, que aunque no incluye ambas técnicas de modelado, es uno de los sistemas más completos en la actualidad, de acuerdo a la literatura revisada.

En la Tabla 5.20 se muestran las características principales de cada uno de los sistemas considerados. En esta tabla se puede observar el propósito de cada sistema, la técnica de modelado utilizada y la forma en cómo se definen las restricciones de ensamble, entre otros



parámetros de evaluación, los cuales han sido identificados en la literatura como los principales que debe tener un sistema de ensambles virtuales.

Tabla 5.20. Comparativo de la plataforma HAMS contra sistemas similares.

| Funciones                                      | HAMS  | SHARP  | HVAS  | HIVEx   | HIDRA                                     |
|--|---|--|---|---|---|
| <b>Propósito</b>                               | Entrenamiento, evaluaciones de ensamble y análisis de simulación  | Planeación y evaluación de ensambles                   | Planeación y evaluación de ensambles                  | Entrenamiento y aprendizaje cognitivo                             | Planeación y evaluación de ensambles      |
| <b>Técnica de modelado del ensamble</b>        | Hibrido (PBM + CBM)   | Hibrido (PBM + CBM)                                    | Hibrido (PBM + CBM)                                   | PBM   | PBM/CBM limitado                          |
| <b>Definición de restricciones de ensamble</b> | Manual y automático basado en las características del modelo  | Manual y automático, basado en intenciones de ensamble | Importación de restricciones de ensamble del CAD      | No soportado  | Manualmente, solo en ejes ortogonales     |
| <b>Registro de datos</b>                       | Posición y orientación de la pieza manipulada, tiempo de manipulación, fuerza, torque y tasas de actualización. | Módulo de grabado y reproducción de trayectorias       | Posiciones, fuerzas y torques                         | Tiempo de ensamble  | No mencionado                             |
| <b>Visualización del ensamble</b>              | Crono-ciclos  | Barrido de volúmenes                                   | Barrido de volúmenes y puntos discretos               | Demostración del ensamble   | No soportado                              |
| <b>Configuración de háptico dual</b>           | Ensamble a dos manos  | Ensamble a dos manos                                   | No soportado  | No soportado  | Dual, con dedos índice y pulgar           |
| <b>Importación de modelos CAD</b>              | Geometría importada utilizando formatos STL, OBJ o VTK  | Geometría importando utilizando modelos Voxel o B-Rep  | Interfaz de importación automática de datos desde CAD | Formato VRLM, incluyendo geometría, y datos físicos y de ensamble | Importa geometría utilizando formato VRLM |
| <b>Método para crear el modelo físico</b>      | Malla triangular, descomposición en elementos convexos, y primitivos, mixta                                     | B-Rep y VPS  | Descomposición en elementos convexos                  | SmartCollision  | Descomposición en elementos convexos      |
| <b>Modelos complejos</b>                       | Soportado   | Soportado  | Soportado   | Soportado   | Solo primitivos o elementos convexos      |
| <b>Ensamble colaborativo</b>                   | Local   | Soportado, por módulo de red                           | No mencionado   | No soportado  | No Soportado                              |
| <b>Diversidad de materiales</b>                | Aluminio, acero, plomo o madera   | No mencionado  | Desde el modelo CAD                                   | Desde el modelo CAD   | Soportado                                 |
| <b>Manipulación de cámara completa</b>         | Por mouse o dispositivo háptico   | No mencionado  | No mencionado   | No mencionado   | Por teclado                               |
| <b>Manejo de sub-ensambles</b>                 | Soportado   | Requiere de la redefinición de propiedades físicas     | No soportado  | No soportado  | No soportado                              |

Aunque los tres sistemas SHARP, HVAS e HIDRA, utilizan el enfoque híbrido para modelar el proceso de ensamble, la definición de las restricciones de ensamble es diferente en cada caso. En el sistema HVAS, el ensamble debe ser previamente realizado en un sistema CAD para después poder importar las restricciones y los modelos en la plataforma virtual. En el sistema SHARP, las restricciones son creadas automáticamente en base a los intentos de ensamble que realiza el usuario, es decir, el sistema identifica lo que el usuario pretende ensamblar. En el sistema HAMS las restricciones son creadas automáticamente al cargar un modelo en la plataforma o manualmente mediante el dispositivo háptico. Los resultados obtenidos en estos sistemas coinciden en que el enfoque híbrido produce un mejor desempeño del proceso de ensamble virtual.

Entre las diferencias más sobresalientes que posee el sistema HAMS respecto a los otros sistemas es la manipulación libre de la cámara, esto es, el usuario puede modificar libremente el punto de vista respecto al ensamble. Esta función puede ser realizada utilizando el dispositivo háptico sin necesidad de soltarlo, ahorrando así tiempo de manipulación y permitiendo un ensamble más intuitivo y realista. Otra de las características importantes de HAMS es la creación de sub-ensambles de manera automática, sin necesidad de actualizar manualmente los valores de masa e inercia. Los sub-ensambles en HAMS son creados mediante juntas cinemáticas que unen la pieza base con la pieza ensamblada. El sistema SHARP también integra la creación de sub-ensambles, sin embargo, estos deben ser manualmente definidos y sus propiedades dinámicas deben ser actualizadas cuando son creados. Finalmente, HAMS es el único sistema de ensambles virtuales que utiliza el método de representación mixta del modelo físico para llevar a cabo el ensamble. Los demás sistemas utilizan la descomposición en elementos convexos o la representación de modelos por voxels.

Entre las deficiencias del sistema HAMS se puede mencionar que la complejidad de los modelos utilizados no es tan alta como en el sistema HIVEx. Además, el sistema HAMS no posee un módulo de entrenamiento que permita guiar al usuario para llevar a cabo un proceso de ensamble óptimo. Por otro lado, el sistema HAMS permite la importación de modelos en formato STL, OBJ o VTK, pero no permite importar datos como restricciones de ensamble previamente creadas en el sistema CAD u otro tipo de datos.

Finalmente se puede concluir que el sistema HAMS posee características y funcionalidades que lo sitúan al mismo nivel de plataformas de ensamble virtual reportadas en la literatura actual.

En el presente trabajo de investigación se ha desarrollado un nuevo sistema de ensambles virtuales asistido por sistemas hápticos, llamado HAMS (*Haptic Assembly and Manufacturing System*). El sistema desarrollado consta de tres módulos principales: el módulo háptico, el módulo físico y el módulo gráfico. La integración de estos tres módulos permitió realizar el proceso de ensamble virtual de una manera intuitiva y realista, por medio de la interacción del usuario con el ambiente virtual a través de un par de dispositivos hápticos, con los cuales se obtiene retroalimentación de fuerza desde el ambiente virtual. Además, el sistema posee paneles de control y *widgets* que permiten manipular los parámetros y variables de simulación, incluyendo el simulador físico, en tiempo de ejecución con el fin de analizar el efecto que tienen sobre el desempeño del ensamble. Estas características hacen del sistema HAMS una plataforma de ensambles virtuales completa, flexible y robusta para la planeación, evaluación, simulación y entrenamiento de tareas de ensamble de una manera natural e intuitiva, la cual está al nivel de otros sistemas desarrollados por grupos de investigación a nivel internacional.

En la evaluación realizada a los simuladores físicos empleados en este trabajo, se pudo observar que para ensambles que utilizan objetos con baja complejidad geométrica, el simulador PhysX ofrece el mejor desempeño, mientras que en caso de modelos con geometrías más complejas, el simulador físico Bullet tiene un mejor rendimiento. Este desempeño de los simuladores puede ser invertido si se mejoran los métodos de representación de objetos en el simulador PhysX. Por tanto se puede concluir que el tipo de simulador físico afecta en gran medida el proceso de ensamble virtual.

La representación de modelos dentro del simulador físico también tiene una gran influencia en el desempeño del proceso de ensamble virtual, al grado de llegar a imposibilitar el ensamble en caso de utilizar una representación no adecuada. En la evaluación realizada a los métodos de representación de modelos en los simuladores físicos se observa que los algoritmos GIMPACT y ConvexFT ofrecen la mejor precisión geométrica en Bullet y PhysX respectivamente.

El uso simultáneo de dos técnicas para el modelado de ensambles virtuales, el ensamble basado en física y el ensamble basado en restricciones, conduce a un mejor desempeño de la plataforma de ensambles al reducir el costo computacional de la detección de colisiones. Como resultado de esta mejora se tiene un proceso de ensamble más fácil e intuitivo, manteniendo un buen nivel de realismo de la aplicación, y reduciendo el tiempo de ensamble hasta en un 58%.

### **Contribuciones de la investigación**

El desarrollo e implementación del sistema HAMS condujo a diversas contribuciones científicas y tecnológicas, las cuales son:

**a) Plataforma de ensamble virtual HAMS y su evaluación**

Se desarrolló el sistema HAMS el cual permite la simulación del ensamble de componentes con geometría y complejidad variable, ofrece facilidad y estabilidad para controlar los objetos virtuales, y provee retroalimentación de fuerza al usuario por medio de los dispositivos hápticos. En términos generales se considera que el sistema tiene un desempeño satisfactorio y puede ser empleado de manera exitosa en la planeación de ensamblajes reales.

Respecto al análisis comparativo del sistema desarrollado contra otros sistemas similares reportados en la literatura, se puede observar que HAMS cuenta con ciertas características que lo sitúan al mismo nivel de los sistemas analizados, e incluso posee algunas funciones que no son reportadas en los otros sistemas como son:

- ✓ El enfoque híbrido para realizar el ensamble virtual (DACs).
- ✓ La definición automática de restricciones de ensamble.
- ✓ La definición manual de restricciones de ensamble por medio del dispositivo háptico.
- ✓ El uso de la representación mixta para el modelo físico.
- ✓ La manipulación libre de la cámara virtual.
- ✓ La creación automática de sub-ensambles.

**b) Metodología de evaluación de simuladores físicos**

La metodología propuesta para evaluar los simuladores físicos ha permitido identificar las fortalezas y debilidades que poseen estos simuladores cuando son utilizados en aplicaciones de ensamble virtual. Esta metodología puede ser utilizada para la evaluación de cualquier simulador físico y no está limitada a los simuladores y plataforma de ensamble considerados en este trabajo.

**c) Restricciones dinámicas de ensamble (DACs)**

Una de las mayores contribuciones es el desarrollo, implementación y evaluación de restricciones dinámicas de ensamble (DACs), las cuales se activan de manera automática y asisten al usuario para alcanzar la posición y orientación de ensamble final, reduciendo la carga computacional de los algoritmos de detección de colisiones, y por tanto mejorando el desempeño del sistema de ensamble virtual. Además, estas restricciones de ensamble pueden ser definidas de manera automática al cargar un modelo nuevo en el sistema, o de forma manual por medio del dispositivo háptico. Aunque el tiempo de finalización de la tarea es mayor en el ambiente virtual que en el ambiente real, el uso de las DACs ayuda a reducir esta diferencia. Los resultados obtenidos muestran que el tiempo de finalización de la tarea de ensamble se redujo hasta en un 58% respecto al ensamble realizado sin DACs.

**d) Algoritmo ConvexFT**

Se propuso un nuevo algoritmo de representación física, llamado ConvexFT, que habilita la detección y respuesta a colisiones de objetos no convexos en simuladores físicos que no soportan este tipo de contactos. Los resultados muestran que al utilizar este algoritmo se obtiene una representación física muy semejante al modelo original. Aunque el algoritmo propuesto presenta ciertas debilidades, éste ha servido como referencia para comparar el desempeño de otros algoritmos y como punto de partida para la creación de un nuevo algoritmo más eficiente para la detección de colisiones de objetos no convexos.

**e) Representación mixta de modelos virtuales**

La representación mixta, mediante la cual un objeto se representa por medio algoritmos de descomposición en elementos convexos (HACD o convexFT) cuando tiene comportamiento dinámico, y por medio de una malla triangular cuando tiene comportamiento estático, condujo a reducir el tiempo de finalización de la tarea hasta en un 65%, y el tiempo de simulación física hasta en un 94%, mejorando el desempeño general del sistema.

**f) Modelo masa-resorte-amortiguador**

El sistema masa-resorte-amortiguador (MRA) permitió acoplar la representación háptica con la representación física para lograr una interacción estable del usuario con los objetos virtuales. La correcta aplicación y calibración de este sistema permite obtener una retroalimentación de fuerza realista mediante la cual el usuario puede experimentar la inercia, peso y colisiones de los objetos virtuales.

**g) Influencia del peso en tareas de ensamble**

De las pruebas realizadas se observó que el peso del objeto manipulado afecta de manera directa el desempeño del ensamble virtual. Los objetos con mayor inercia o peso tienden a seguir su estado de movimiento o reposo, por tanto resulta más difícil moverlos o cambiar su trayectoria. Esto se ve reflejado en movimientos más lentos y trayectorias no lineales, lo que resulta en mayores tiempos de finalización de la tarea.

Las metodologías o algoritmos propuestos en este trabajo de investigación no son exclusivos del sistema desarrollado, pueden ser exportadas a otro tipo de sistemas y/o aplicaciones. Además, las evaluaciones realizadas permiten identificar la influencia de los diferentes parámetros evaluados en el desempeño de tareas de ensamble virtual, generando información valiosa para el desarrollo de sistemas de realidad virtual más confiables, rápidos y tecnológicamente viables para su uso en aplicaciones industriales.

## Trabajo a futuro

Con el propósito de continuar con el trabajo de investigación presentado en esta tesis, las siguientes actividades se identifican como trabajo futuro:

1. Detección de colisiones. La limitación más importante en los sistemas de realidad virtual es el proceso de detección de colisiones, por lo que la mejora de los algoritmos de detección de colisiones actuales, así como su implementación en el sistema HAMS es de importancia en futuros desarrollos.
2. Mejora del algoritmo ConvexFT. El algoritmo ConvexFT presentado en este trabajo muestra grandes áreas de oportunidad, como la solución de errores que se presentan en las uniones de cada elemento creado. Además debe ser simplificado y optimizado para consumir la menor cantidad de recursos, por ejemplo se debe de incluir la detección de colisiones por jerarquías, donde solo se analizan elementos que estén próximos a una zona de colisión.
3. Manipulación de cámara. Actualmente la cámara del espacio gráfico puede ser manipulada mediante el mouse o el dispositivo háptico, sin embargo para los usuarios sin experiencia esta operación es confusa, por tanto se propone implementar un algoritmo y dispositivo que permita al usuario manipular la cámara con movimientos de la cabeza.
4. Restricciones de ensamble. Se requiere implementar y evaluar otros tipos de restricciones, por ejemplo de paralelismo, angulares, de contacto, etc., con el fin de facilitar el proceso de ensamble de partes con diferentes características geométricas.
5. Algoritmos de representación de modelos. Se considera la implementación de algoritmos de representación, nuevos o ya existentes, con el fin de optimizar el proceso de detección de colisiones.
6. Importación de datos desde sistema CAD. Para lograr la implementación del sistema HAMS en ambientes de diseño industrial se considera el desarrollo de un módulo que permita importar datos relativos al ensamble y propiedades de materiales desde el sistema CAD, por ejemplo restricciones de ensamble, ubicación de las piezas, tipos de material, etc.

7. Uso de elementos de sujeción y herramientas. Se prevé incluir una base de datos que contenga diversos elementos de sujeción, por ejemplo tornillos, remaches, e incluso soldadura, que el usuario pueda agregar al componente ensamblado.
8. Funcionalidad de la plataforma. Se considera la implementación de nuevas características que permitan hacer de la plataforma un sistema más amigable e intuitivo, por ejemplo, minimizar el uso de paneles de control y sustituirlos por *widgets*. También se pueden agregar funciones para crear elementos básicos como cubos, cilindros, etc., y modificar su forma a través de funciones booleanas. Adicionalmente para incrementar el realismo del ambiente virtual, se considera la implementación de audio en el sistema HAMS.
9. Implementación. Parte del trabajo futuro considerado es la implementación y evaluación del sistema HAMS como una herramienta en tareas de diseño, manufactura y ensamble.
10. Entrenamiento. Como trabajo futuro también se considera la evaluación del sistema HAMS como una herramienta para el entrenamiento de ensambles.
11. Extensión de la plataforma. Se considera la integración de módulos que permitan realizar análisis de manufacturabilidad e incluso simulación de otros procesos de manufactura, por ejemplo el ensamble o maquinado robótico.
12. Ensamble colaborativo. Se contempla añadir al sistema HAMS un módulo de red que permita trabajar con distintos usuarios en diferentes ubicaciones geográficas.
13. Análisis y síntesis de mecanismos. El modelado basado en física habilita a los objetos virtuales con un comportamiento realista, por tanto se puede aprovechar esta característica para crear un módulo que permita evaluar la funcionalidad de mecanismos.
14. Evaluaciones con modo háptico dual. El sistema HAMS permite realizar ensambles utilizando las dos manos, sin embargo esta función no ha sido estudiada completamente. Se sugiere evaluar el uso de dos dispositivos hápticos para determinar si su uso mejora o no el proceso de ensamble virtual.
15. Tipos de dispositivos hápticos. La implementación de dispositivos hápticos con múltiples puntos de contacto, por ejemplo un guante háptico, es una alternativa de implementación a futuro en el sistema HAMS.

- 
- 
- [1] Ellis, S.R. (1994). What are virtual environments? *IEEE Computer Graphics and Applications*, vol. 14(1), pp. 17-22.
- [2] Jayaram, S. Connacher, H.I. y Lyons, K.W. (1997). Virtual assembly using virtual reality techniques. *Computer Aided Design*, vol. 29(8), pp. 575-584.
- [3] Burdea, G. C. (1996). Force and touch feedback for virtual reality. *John Wiley & Sons, Inc. New York, USA*.
- [4] Havok, "Havok physics". Consultado el 17 de enero de 2014 en: <http://www.havok.com/products/physics>.
- [5] Real Time Physics Simulation, "Bullet Physics Library". Consultado el 17 de enero de 2014 en: <http://bulletphysics.org/wordpress/>.
- [6] Nvidia Developer Zone, "PhysX SDK". Consultado el 17 de enero de 2014 en: <http://developer.nvidia.com/physx>.
- [7] Perez, J. y Suero, A. "Newton game dynamics". Consultado el 17 de enero de 2014 en: <http://newtondynamics.com/forum/newton.php>.
- [8] Smith, R. "Open Dynamics Engine". Consultado el 17 de enero de 2014 en: <http://www.ode.org/>.
- [9] Kim, C.E., y Vance, J.M. (2004). Collision detection and part interaction modeling to facilitate immersive virtual assembly methods. *Journal of Computing and Information Science in Engineering*, vol. 4(2), pp. 83-90.
- [10] Hayward, V., Astley, O. R., Cruz-Hernandez, M., Grant, D., y Robles-De-La-Torre, G. (2004). Haptic interfaces and devices. *Sensor Review, Emerald group publishing*, vol. 24(1), pp. 16-29.
- [11] Thurffjell, L., McLaughlin, J., Mattson J., y Lammertse P. (2002). Haptic interaction with virtual objects: the technology and some applications. *Industrial robot: An International Journal*, vol. 29(3), pp. 210-215.
- [12] Boothroyd, G. (2005), Assembly automation and product design. *Cambridge University press*, vol. 536. Boca Raton, Fl, Taylor & Francis.
- [13] Xia, P., Lopes, A.M., y Restivo, M.T. (2013). A review of virtual reality and haptics for product assembly (part 1): rigid parts. *Assembly Automation*, vol. 33(1), pp. 68 -77.



- [14] Gutiérrez, T., Barbero, J.I., Aizpitarte, M., Carrillo, A.R., y Eguidazu A. (1998). Assembly simulation through haptic virtual prototypes. *Proceedings of the third PHANTOM users group workshop, Cambridge, Massachusetts.*
- [15] Liu, M., Wang, D., y Zhang, Y. (2010). A novel haptic rendering algorithm for stable and precise 6-dof virtual assembly. *Proceedings of the ASME 2010 World Conference on Innovative Virtual Reality (WINVR2010), Ames, IA, USA, may 12-14*, pp. 251-257.
- [16] Lim, T., Ritchie, J.M., Dewar, R.G. y Corney, J.R. (2007). Factors affecting user performance in haptic assembly. *Virtual Reality*, vol. 11(4), pp. 241-252.
- [17] Seth, A., Vance, J.M., y Oliver, J.H. (2007). Combining geometric constraints with physics modeling for virtual assembly using SHARP. *ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2007/CIE-34681)*. Las Vegas, NV, USA.
- [18] Xia, P., Lopes, A.M., y Restivo M. (2011). Design and implementation of a haptic-based virtual assembly system. *Assembly Automation*, vol. 31(4), pp. 369-384.
- [19] Bhatti, A., Nahavandi, S., Khoo, Y.B., Creighton, D., Anticev, J., y Zhou, M. (2009). Haptically enabled interactive virtual assembly training system development and evaluation. *SIMTEC 2009: Proceedings of the 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Adelaide, S. Aust.*, pp. 1-6.
- [20] Coutee, A.S., McDermott, S.D., y Bras, B. (2001). A haptic assembly and disassembly simulation environment and associated computational load optimization techniques. *Journal of Computing and Information Science in Engineering*, vol.1 (2), pp.113-122.
- [21] Henderson, M., y Kattethota, G. (1999). Manufacturability evaluation for rapid fabrication: A system form traditional machining and layered manufacturing planning. *CAD Conference, Neuchatel, Switzerland*, Feb. 22-24.
- [22] Wang, Y.X. (2003). Conceptual design supported automatic virtual prototype design method for mechanical devices. *ASME 2003 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2003)*, September 2–6, Chicago, Illinois, USA, Paper no. DETC2003/DTM-48639, pp. 443-448.
- [23] Song, Y., Vergeest, J.S.M., y Wieggers, T. (2004). Interactive freeform clay modeling supported by 3d scanning and robot machining. *ASME 2004 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2004)*, September 28–October 2, Salt Lake City, Utah, USA Paper no. DETC2004-57765, pp. 805-812.

- [24] Chan, C.K., y Tan, S.T. (2005). Volume decomposition of CAD models for rapid prototyping, technology. *Rapid Prototyping Journal*, vol. 11(4), pp. 221–234.
- [25] Aras, E., y Yip-Hoi, D. (2007). Geometric modeling of cutter/work piece engagements in 3-axis milling using polyhedral models. *ASME 2007 International Mechanical Engineering Congress and Exposition (IMECE2007)*, November 11–15, Seattle, Washington, USA. Paper no. IMECE2007-41414, pp. 371-382.
- [26] Acal, A.P., y Lobera, A.S. (2007). Virtual reality simulation applied to a numerical control milling machine. *International Journal on Interactive Design and Manufacturing (IJIDEM)*. vol. 1(3), pp. 143-154.
- [27] Medellín, H., Lim, T., Corney, J., Ritchie J.M., y Davies J.B.C. (2007). Automatic subdivision and refinement of large components for rapid prototyping production. *Journal of Computing and Information Science in Engineering*, September 2007, vol. 7(3), pp. 249-258.
- [28] Doruk-Merdol, S., y Altintas, Y. (2008). Virtual simulation and optimization of milling operations-Part I: process simulation. *Journal of Manufacturing Science and Engineering*, vol. 130(5).
- [29] Chen, Z.C., y Cai, W. (2008). An efficient, accurate approach to representing cutter-swept envelopes and its applications to three-axis virtual milling of sculptured surfaces. *Journal of Manufacturing Science and Engineering*, vol. 130(3).
- [30] Sundararajan, V., y Wright, P.K. (2008). Applications of software engineering to manufacturing process planning. *Journal of Computing and Information Science in Engineering*, vol. 8(3).
- [31] Molcho, G., Schneur, R., Zipori, Y., Kowalsi, P., Denkena, B., y Shpitalni, M. (2008). Computer aided manufacturability analysis closing the CAD-CAM knowledge gap. *9th Biennial Conference on Engineering Systems Design and Analysis (ESDA2008)*, Haifa, Israel, Paper no. ESDA2008-59280, pp. 309-315.
- [32] Di Gironimo, G., y Lanzotti, A. (2009). Design in VR. *International Journal of Interactive Design and Manufacturing*, vol. 3(2), pp. 51-53.
- [33] Payandeh, S., y Azouz, N. (2001). Finite elements, mass spring-damper systems and haptic rendering. *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation 2001*, pp. 224-229.
- [34] Vance, J.M., y Dumont G. (2011). A conceptual framework to support natural interaction for virtual assembly tasks. *Proceedings of the ASME 2011 World Conference on Innovative Virtual Reality, WINVR 2011*, June 27-29, Milan, Italy, pp. 273-278.

- [35] Potter, K., Johnson, D., y Cohen, E. (2004). Haptically guided filtering for reverse engineering. *ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference, DETC 2004-57539*, September 28- October 2, Salt Lake City, Utha, USA.
- [36] Yang, Z., y Chen, Y. (2005). A reverse engineering method based on haptic volume removing. *Computer Aided Design*, vol. 37(1), pp. 45-54.
- [37] Yang, Z., Chen, Y. y Lian, L. (2005). On the development of a haptic system for rapid product development. *Computer Aided Design*, vol. 37(5), pp. 559-569.
- [38] He, X. (2008). Haptic modeling for virtual manufacturing. (Tesis Doctoral). *Universidad de Hong Kong*.
- [39] Yang, Z., Lian, L., y Chen, Y. (2005). Haptic function evaluation of multi-material part design. *Computer-Aided Design*, vol. 37(7), pp. 727-736, 2005.
- [40] Balasubramaniam, M., Ho, S., Sarma, S., y Adachi, Y. (2002). Generation of collision-free 5-axis tool paths using a haptic surface. *Computer Aided Design*, vol. 34(4), pp. 267-279.
- [41] Zhu, W., y Lee, Y.S. (2004). Product prototyping and manufacturing planning with 5-dof haptic sculpting and dixel volume updating. *IEEE, Proceedings of the 12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS'04)*.
- [42] Ramesh, K. (2004). A haptic stencil for manufacturing applications. (Tesis de maestría), *Departamento de Ingeniería Mecánica, Massachusetts Institute of Technology*.
- [43] Ren, Y.F., Lai-Yuen, S.K., y Lee, Y.S. (2006). Virtual prototyping and manufacturing planning by using tri-dixel models and haptic force feedback. *Virtual and Physical Prototyping*, vol. 1(1).
- [44] Chen, K., y Tang, K. (2008). Interactive haptic refinement of a five-axis finishing cut. *Journal of Computing and Information Science in Engineering*, vol. 8(4).
- [45] Chen, Y.H., Wang, Y.Z., y Yang, Z.Y. (2004). Towards a haptic virtual coordinate measuring machine. *International Journal of Machine Tools & Manufacture*, vol. 44, pp. 1009-1017.
- [46] Arbabtafti, M., Moghaddam M., Nahvi A., Mahvash M., Richardson B., Shirinzadeh B. (2011). Physics-based haptic simulation of bone machining. *IEEE Transaction on Haptics*, vol. 4(1), pp. 39-49.

- [47] Zerbato, D., Baschiroto, D., Baschiroto, D., Botturi, D., y Fiorini, P. (2011). GPU-based physical cut in interactive haptic simulations. *International Journal Computer Assisted Radiology and Surgery*, vol. 6(2), pp. 265-272.
- [48] Seth, A., Vance, J.M. y Oliver, J.H. (2011). Virtual reality for assembly methods prototyping: a review. *Virtual Reality*, vol. 15(1), pp. 5-50.
- [49] Xinhua, L., Youhui, L., y Qi, L. (2011). Virtual assembly and simulation of vibration sieve based on a human-interface environment. *Advances in Control Engineering and Information Science, Procedia Engineering*, vol. 15, pp. 2988-2992.
- [50] Jayaram, S., Jayaram, U., Wang, Y., Tirumali, H., Lyons, K., y Hart, P. (1999). VADE: a Virtual Assembly Design Environment. *Computer Graphics and Applications*, IEEE, vol. 19(6), pp. 44 - 50.
- [51] Zhu, H., Wu, D., y Fan, X. (2010). Assembly semantics modeling for assembling process planning in virtual environment. *Assembly Automation*, vol. 30(3), pp. 257 – 267.
- [52] Simmons, J.E.L., Dewar, R., Ritchie, J.M., y Meng Ng, F. (1998). The application of virtual reality to tasks in manufacturing and assembly engineering. *KSME International Journal*, vol. 12(1), pp. 1-11.
- [53] Yuan, X., y Yang, S.X. (2004). Interactive assembly planning with automatic path generation. *International Conference on Intelligent Robots and Systems (IROS 2004)*. IEEE/RSJ. Issue: 28, Sept.-2 Oct. 2004, vol. 4, pp. 3965 - 3970.
- [54] Shen, Z., Shah, J.J., y Davidson, J.K. (2005). Simulation-based tolerance and assemblability analyses of assemblies with multiple pin/hole floating mating conditions. *ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2005)*, September 24–28, Long Beach, California, USA, Paper no. DETC2005-85398, pp. 429-440.
- [55] Shen, Z., Shah, J.J., y Davidson, J.K. (2005). A complete variation algorithm for slot and tab features for 3d simulation-based tolerance analysis. *ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2005)*, September 24–28, Long Beach, California, USA, Paper no. DETC2005-85541, pp. 587-597.
- [56] Sung, R.C.W., Ritchie, J.M., Lim, T., y Medellin, H. (2009). Assembly planning and motion study using virtual reality. *ASME World Conference on Innovative VR*, Chalon-sur-Saône, France, *WINVR2009-713*, pp. 31-38.

- [57] Brough, J., Schwartz, M., Gupta, S.K., Anand, D. K., Kavetsky, R., y Pettersen, R. (2007). Towards the development of a virtual environment-based training system for mechanical assembly operations. *Virtual Reality*, vol. 11(4), pp. 189-206.
- [58] Fischer, A., Vance, J.M., Vo, D.M. (2009). Haptic feedback to guide interactive product design. *ASME-AFM 2009 World Conference on Innovative Virtual Reality (WINVR2009)* February 25–26, 2009 , Chalon-sur-Saône, France.
- [59] Lim, T., Medellin, H., Sung R., Ritchie, J.M. y Corney, J. (2009). Virtual boxing assembly rapid prototyping of near net shapes. *ASME-AFM 2009 World Conference on Innovative Virtual Reality (WINVR2009)* February 25–26, 2009 , Chalon-sur-Saône, France.
- [60] Iacob, R., Mitrouchev, P., y Léon, J. (2011). Assembly simulation incorporating component mobility modeling. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, vol. 5(2), pp. 119-132.
- [61] Sato, K., Date, H., y Onosato, M. (2011). Fast matching, combinations extraction and configuration of mesh models using graph-based feature representation. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, vol. 5(2), pp. 133–136.
- [62] Zaldivar-Colado, y U., Garbaya, S. (2009). Virtual assembly environment modelling. *Proceedings of the ASME/AFM 2009 World Conference on Innovative Virtual Reality WINVR2009*, February 25-26, Chalon-sur-Saône, France.
- [63] Gutiérrez, T., Rodríguez, J., Vélaz, Y., Casado, S., Suescun, A., y Sánchez, E.J. (2010). IMA-VR: A multimodal virtual training system for skills transfer in industrial maintenance and assembly tasks. *19th IEEE International Symposium on Robot and Human Interactive Communication*, Principe di Piemonte - Viareggio, Italy.
- [64] Tching, L., Dumont, G., y Perret, J. (2010). Interactive simulation of CAD models assemblies using virtual constraint guidance. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, vol. 4(2), pp. 95-102.
- [65] Aleotti, J., y Caselli, S. (2011). Physics based virtual reality for task learning and intelligent disassembly planning. *Virtual Reality*, vol. 15(1), pp. 41–54.
- [66] Xia, P., Lopes, A.M., Restivo, M.T., y Yao, Y. (2012). A new type haptics-based virtual environment system for assembly training of complex products. *The International Journal of Advanced Manufacturing Technology*, vol. 58(1), pp. 379-396.
- [67] Chen, C.J., Ong, S.K., Nee, A.Y.C., y Zhou, Y.Q. (2010). Haptic-based interactive path planning for a virtual robot arm. *Int. Journal on Interactive Design and Manufacturing*, vol. 4(2), pp. 113–123.

- [68] Cheng-jun, C., Yun-feng, W., y Niu, L. (2010). Research on interaction for virtual assembly system with force feedback. *Proceedings of the 2010 Third International Conference on Information and Computing, ICIC '10*, vol. 02, pp. 147-150.
- [69] Ritchie, J.M., Lim, T., Sung, R.S., Corney, J.R., y Rea, H. (2008). The analysis of design and manufacturing tasks using haptics and immerse VR: some case studies. In *Talaba Doru and Amditis Angelos (eds.): Product Engineering: Tools and Methods Based on Virtual Reality*, Springer, Netherlands. pp. 507-522.
- [70] Bordegoni, M., Cugini, U., Belluco, P., y Aliverti, M. (2009). Evaluation of a haptic-based interaction system for virtual manual assembly. *Virtual and Mixed Reality, Lecture Notes in Computer Science*, vol. 5622, pp. 303–312.
- [71] Christiand, y Yoon, J. (2011). Assembly simulations in virtual environments with optimized haptic path and sequence. *Robotics and Computer-Integrated Manufacturing*, vol. 27(2), pp. 306–317.
- [72] Ji, P., Choi, A.C.K. y Tu, L. (2002). VDAS: a virtual design and assembly system in a virtual reality environment. *Assembly Automation*, vol. 22(4), pp. 337-342.
- [73] Wan, H., Gao, S., Peng, Q., Dai, G., y Zhang, F. (2004). MIVAS: a multi-modal immersive virtual assembly system. *Proceedings of the ASME Design Engineering Technical Conference*, Salt Lake City, UT, pp.113-122.
- [74] Seth, A., Su H.J., y Vance J.M. (2006). SHARP: a system for haptic assembly & realistic prototyping. *Proceedings of the ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Philadelphia, PA, USA, vol. 6, pp. 1-9.
- [75] Lim, T., Ritchie, J.M., Corney, J.R., Dewar, R.G., Schmidt, K., y Bergsteiner, K. (2007). Assessment of a haptic virtual assembly system that uses physics-based interactions. *Proceedings of the 2007 IEEE International Symposium on Assembly and Manufacturing*, July 22-25, AnnArbor, Michigan, USA, pp.147-153.
- [76] Ritchie, J.M., Lim, T., Sung, R.S. y Medellin, H. (2008). Generation of assembly process plans and associated gilbreth motion study data. *2nd International Workshop Virtual Manufacturing VirMan 08 as part of the 5th INTUITION International Conference: Virtual Reality in Industry and Society: From Research to Application*, October 6-8, Torino, Italy, ISBN 978-960-89028-7-9.
- [77] Garbaya, S., y Zaldivar-Colado, U. (2009). Modeling dynamic behavior of parts in virtual assembly environment. *Proceedings of the ASME/AFM 2009 World Conference on Innovative Virtual Reality WINVR2009-742*, February 25-26, Chalon-sur-Saône, France, pp. 89-99.

- [78] Hu, Y., Wu, D., Fan, X. y Zhen, X. (2010). Grid-enabled collaborative virtual assembly environment. *Assembly Automation*, vol. 30(4), pp. 352 – 364.
- [79] Hassan, S. y Yoon, J. (2010). Haptic based optimized path planning approach to virtual maintenance assembly / disassembly (MAD). *The 2010 IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, October 18-22, Taipei, Taiwan.
- [80] Gupta, R., Whitney, D., y Zeltzer, D. (1997). Prototyping and design for assembly analysis using multimodal virtual environments. *Computer-Aided Design*, vol. 29(8), pp. 585-597.
- [81] McDermott, S. D., y Bras, B. (1999). Development of a haptically enabled dis/re-assembly simulation environment. *Proceedings of the ASME Design engineering Technical Conferences, DETC*, September 12-15, Las Vegas, Nevada, USA.
- [82] Chen, E. (1999). Six degree-of-freedom haptic system for desktop virtual prototyping applications. *Proceedings of the First International Workshop on Virtual Reality and Prototyping*, Laval, France, pp. 97-106.
- [83] Bloomfield, A., Deng, Y., Wampler, J., Rondot, P., Harth, D., McManus, M., y Badler, N. (2003). A taxonomy and comparison of haptic actions for disassembly tasks. *Virtual Reality, proceedings IEEE*, pp. 225-231.
- [84] Seth, A., Su H.J., y Vance, J.M. (2005). A desktop networked haptic VR interface for mechanical assembly. *Proceedings of IMECE '05 ASME Int. Mechanical Engineering Congress, IMECE2005-81873*, Orlando, Florida, USA, pp. 173-180.
- [85] He, X., y Chen, Y. (2009). Haptic-aided robot path planning based on virtual tele-operation. *Robotics and Computer-Integrated Manufacturing*, vol. 25(4), pp. 792-803.
- [86] Howard, B.M., y Vance, J.M. (2007). Desktop haptic virtual assembly using physically based modeling. *Virtual Reality*, vol. 11(4), pp. 207-215.
- [87] Jayaram, S., Jayaram, U., Kim, Y. J., DeChenne, C., Lyons, K. W., Palmer, C., y Mitsui, T. (2007). Industry case studies in the use of immersive virtual assembly. *Virtual Reality*, vol. 11(4), pp. 217–228.
- [88] Iglesias, R., Prada, E., Uribe, A., Garcia-Alonso, A., Casado, S., y Gutierrez, T. (2007). Assembly simulation on collaborative haptic virtual environments. *15th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, pp. 241-247.
- [89] Iglesias R., Casado, S., Gutiérrez, T., García-Alonso, A., Yu, W., y Marshall, A. (2008). Simultaneous remote haptic collaboration for assembling tasks. *Multimedia Systems*, vol. 13(4), pp. 263-274.

- [90] Joshi, H., Jayaram, S., y Jayaram, U. (2008). An open architecture for embedding VR-based mechanical tools into cad applications. *ASME 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2008)*, August 3–6, Brooklyn, New York, USA, Paper no. DETC2008-50138, pp. 1625-1635,
- [91] Yoon, J., y Kumar, P. (2009). A novel optimal assembly algorithm for the haptic interface application of a virtual maintenance system. *Journal of Mechanical Science and Technology*, vol. 23(1), pp. 183-194.
- [92] Yoon, J., Auralius, M., y Yu, W. (2009). An enhanced haptic assembly simulation system for the efficiency of assembly tasks. *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, pp. 1241-1246.
- [93] Bhatti, A., Khoo, Y.B., Creighton, D., Anticev, J., Nahavandi, S., y Zhou, M. (2008). Haptically enabled interactive virtual reality prototype for general assembly. *World Automation Congress, WAC 2008*, Sept. 28 -Oct. 2, pp. 1-6.
- [94] Jia, D., Bhatti, A., y Nahavandi, S. (2009). Design and evaluation of a haptically enable virtual environment for object assembly training. *IEEE International Workshop on Haptic Audio visual Environments and Games, HAVE 2009*, 7-8 Nov. 2009, pp. 75-80.
- [95] Radi, M., y Reinhart G. (2009). Industrial haptic robot guidance system for assembly processes. *IEEE International Workshop on Haptic Audio Visual Environments and Games, HAVE 2009*. 7-8 Nov. 2009, pp. 69-74.
- [96] Abate, A.F., Guida, M., Leoncini, P., Nappi, M., y Ricciardi, E. (2009). A haptic-based approach to virtual training for aerospace industry. *Journal of Visual Languages and Computing*, vol. 20, pp. 318-325.
- [97] Ladeveze, N., Fourquet, J., y Puel, B. (2010). Interactive path planning for haptic assistance in assembly tasks. *Computers & Graphics*, Vol. 34(1), pp. 17–25.
- [98] Wang, Y., Jayaram, S., Jayaram, U. Lyons, K., y Hart, P. (2001). Physically based modeling in virtual assembly. *The International Journal of Virtual Reality*, vol. 5(1), pp. 1-14.
- [99] Seth, A., Su, H.J., y Vance, J.M. (2008). Development of a dual-handed haptic assembly system: SHARP. *Journal of Computing and Information Science in Engineering*, vol. 8(4).
- [100] Seugling, A., y Rölin, M. (2006). Evaluation of physics engines and implementation of a physics module in a 3D authoring tool. (Tesis de maestría en ciencias de la computación), *Umea University*.



- [101] Ritchie, J.M., Lim, T., Medellín, H., y Sung, R.S. (2009). A haptic based virtual assembly system for the generation of assembly process plans. *XV Congreso Internacional SOMIM*, 23-25 septiembre 2009, Cd. Obregón, Sonora, México.
- [102] Glondu, L., Marchal, M., y Dumont, G. (2010). Evaluation of physical simulation libraries for haptic rendering of contacts between rigid bodies. *Proceedings of ASME, WINVR 2010*, Ames, Iowa, USA, WINVR201-3726, pp. 41-49.
- [103] Nourian, S., Shen, X., y Georganas, N. D. (2005). Role of extensible physics engine in surgery simulations. *IEEE International workshop on Haptic Audio Visual Environments and their Applications*, HAVE 2005, Ottawa, Ontario, Canada.
- [104] Yeh, T.Y., Faloutsos, P., y Reinman, G. (2006). Enabling real-time physics simulation in future interactive entertainment. *Proceedings of the 2006 ACM SIGGRAPH Symposium on Videogames*, Boston, Massachusetts, pp. 71-81.
- [105] Kaufmann, H., y Meyer, B. (2007). Physics education in VR an example. *Themes in science and technology education: virtual reality in education, Klídarithmos Computer Books, Athen, Griechenland*, vol. 2(1-2), pp. 117-130.
- [106] Laurell, B. (2008). The inner workings of real-time physics simulation engines. *IRCSE '08 IDT Workshop on Interesting Results in Computer Science and Engineering*, Mälardalen University, Vasteras, Sweden.
- [107] Choi, J.H., Shin, D., Heo, W., y Shin, D. (2004). Performance evaluation of numerical integration methods in the physics engine. *Computational and Information Science, Lectures Notes in Computer Science*, vol. 3314, pp. 238-244.
- [108] Coumans, E., y Victor, K. (2007). COLLADA Physics. *Web3D '07 Proceedings of the Twelfth International Conference on 3D web Technology*, pp. 101-104.
- [109] Boeing, A., y Bräunl, T. (2007). Evaluation of real-time physics simulation systems. *Graphite '07 Proceedings of the 5th international conference on Computer graphics and interactive techniques*, Australia y South West Asia, pp. 281-288.
- [110] Chan, L.S.H., y Choi, K.S. (2009). Integrating PhysX and OpenHaptics efficient force feedback generation using physics engine and haptic devices. *Joint Conferences on Pervasive Computing (JCPC 09)*, pp. 853-858.
- [111] Chen, T., (2010). Virtual assembly of mechanical components and collision detection. *International Conference on Electronics and Information Engineering (ICEIE 2010)*, IEEE vol.1, pp. 443-447.

- [112] Chandru, V., y Manohar, S. (1997). Volume modelling for emerging manufacturing technologies. *Sadhana: Academy Proceedings in Engineering Sciences*, vol. 22(2), pp. 199-216.
- [113] Lock, S.M, y Wills, D.P.M. (2000). Voxcollide, voxel collision detection for virtual environments. *Virtual Reality*, vol. 5(1), pp. 8-22.
- [114] Tesic, R. y Banerjee, p. (2001). Exact collision detection for a virtual manufacturing simulator. *Institute of Industrial Engineers, IIE Transactions*, vol. 33(1), pp. 43-54.
- [115] Johnson, D.E., y Willemsen, P. (2003). Six degree of freedom haptic rendering of complex polygonal models. *Proceedings of the 11<sup>th</sup> Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS'03)*, pp. 229-235.
- [116] Johnson, D.E., Willemsen, P., y Cohen, E. (2004). A haptic system for virtual prototyping of polygonal models. *Design Engineering Technical Conferences and Computers and Information in Engineering Conference, DETC 2004-57507*.
- [117] Kim, C.E., y Vance, J.M. (2004). Collision detection and part interaction modeling to facilitate immersive virtual assembly methods. *Journal of Computing and Information Science in Engineering*, vol. 4, pp. 83-90.
- [118] Hippmann, G. (2004). An algorithm for compliant contact between complexly shaped bodies. *Multibody System Dynamics*, Springer, vol. 12(4), pp. 345–362.
- [119] Yoon, S.E., Salomon, B., Lin, M., y Manocha, D. (2004). Fast collision detection between massive models using dynamic simplification. *Proceedings of the Eurographics Symposium on Geometry Processing, SGP 04*, pp. 136-146.
- [120] Jimenez, J.J., y Segura, R.J. (2008). Collision Detection between complex polyhedral. *Computers & Graphics*, vol. 32(4), pp. 402– 411.
- [121] Choi, j., Ryu, H.S., Kim, C.W., Y Choi, J.H. (2010). An efficient and robust contact algorithm for a compliant contact force model between bodies of complex geometry. *Multibody System Dynamics*, Springer, vol. 23(1), pp. 99–120.
- [122] Weller, R., Mainzer, D. y Zachmann, G. (2010). A benchmarking suite for 6-dof real time collision response algorithms. *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology (VRST 10)*, pp. 63-70.
- [123] Tching, L., Dumont, G., y Perret, J. (2010). Haptic assembly of CAD models using virtual constraint guidance. *Proceedings of the ASME 2010 World Conference on Innovative Virtual Reality, WINVR 2010*, Ames, Iowa, USA, pp. 11-19.

- [124] Patil, A., y Sen, D. (2009). Haptics aided kinematic assembly modeling and efficient determination of joint ranges of motion. *14th National Conference on Machines and Mechanisms (NaCoMM09)*, pp. 68-75.
- [125] Baraff, D. (1989). Analytical methods for dynamic simulation of non-penetrating rigid bodies. *Computer Graphics*, vol. 23(3), pp. 223-232.
- [126] Ruspini, D.C., y Khatib, O. (1997). Collision/contact models for the dynamic simulation of complex environments. *9th International Symposium of Robotics Research (ISRR'99)*, Snowbird, pp. 185-195.
- [127] Johnson, D.E., Willemsen, P., Cohen, E. (2005). Six degree-of-freedom haptic rendering using spatialized normal cone search. *IEEE, Trans. on Visualization and Computer Graphics*, vol. 11(6), pp. 661-670.
- [128] Zhu, Z., Gao, S., Wan, H., y Luo, Y. (2004). Grasp identification and multi-finger haptic feedback for virtual assembly. *ASME 2004 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2004)*, Salt Lake City, Utah, USA, Paper no. DETC2004-57718, pp. 661-669.
- [129] Otaduy, M.A., y Lin, M.C. (2006). A modular haptic rendering algorithm for stable and transparent 6-dof manipulation. *IEEE Transactions on Robotics*, vol. 22(4), pp. 751-762.
- [130] Garbaya, S., Zaldivar-Colado, U. (2007). The affect of contact force sensations on user performance in virtual assembly tasks. *Virtual Reality*, Springer, Vol. 11(4), pp. 287-299.
- [131] Barbic, J., y James, D.L. (2008). Six-DoF haptic rendering of contact between geometrically complex reduced deformable models. *IEEE Transaction on Haptics*, vol. 1(1), pp. 39-52.
- [132] McNeely, W.A., Puterbaugh, K.D. y Troy, J.J. (1999). Six degree-of-freedom haptic rendering using voxel sampling. *Proceedings of ACM SIGGRAPH*, pp. 401-408.
- [133] Adams, R. J., Klowden, D., y Hannaford, B. (2001). Virtual training for a manual assembly task. *Haptics-e*, vol. 2(2), pp. 1-7.
- [134] Unger, B. J., Nicolaidis, A., Berkelman, P. J., Thompson, A., Klatzky, R. L., y Hollis, R. L. (2001). Comparison of 3-D haptic peg-in-hole tasks in real and virtual environments. *In Intelligent Robots and Systems, IEEE/RJSJ*, vol. 3, pp. 1751-1756.
- [135] Huang, F., Gillespie, R.B., y Kuo, A. (2002). Haptic feedback and human performance in a dynamic task. *Proceedings of the 10th Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems, (HAPTICS'02)*, IEEE, pp. 24-31.

- [136] Ruffaldi, E., Morris, D., Edmunds, T., Barbagli, y F. Pai, D. (2006). Standardized evaluation of haptic rendering systems. *IEEE 14th Symposium on Haptic interfaces for Virtual Environment and Teleoperator Systems*, pp. 225-232.
- [137] Sagardia, M., Hulin, T., Preusche, C., y Hirzinger, G. (2009). Benchmark of force quality in haptic rendering. *Proc. Of the 13th International Conference on Human Computer Interaction, (HCI)*, San Diego, USA.
- [138] Aleotti, J., Denaro, F., Caselli, S. (2010). Object manipulation in visuo-haptic augmented reality with physics-based animation. *19th IEEE International Symposium on Robot and Human Interactive Communication*, Principe di Piemonte - Viareggio, Italy.
- [139] Sagardia, M., Weber, B., Hulin, T., y Hirzinger, G. (2012). Evaluation of visual and force feedback in virtual assembly verifications. *IEEE Virtual Reality Short Papers and Posters (VRW)*, Costa Mesa, CA, USA, pp. 23-26.
- [140] Giachristis, C., Barrio, J., Ferre, M., Wing, A., y Ortego, J. (2009). Evaluation of weight perception during unimanual and bimanual manipulation of virtual objects. *Third Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, Salt Lake City, UT, USA, pp. 629-634.
- [141] Tzafestas, C.S., (2003). Whole hand kinesthetic feedback and haptic perception in dextrous virtual manipulation. *Transactions on Systems, Manufacturing and Cybernetics part A: Systems and Humans*, IEEE, vol. 33(1), pp. 100-113.
- [142] Dominjon, J., Lécuyer, A., Burkhardt, J.M., Richard, P., y Richir, S. (2005). Influence of control/display ratio on the perception of mass of manipulated objects in virtual environments. *VR '05 Proceedings of the 2005 IEEE Conference on Virtual Reality*, pp. 19-25.
- [143] Hara, M., Ashitaka, N., Tambo, N., Huang, J., y Yabuta, T., (2008). Consideration of weight discriminative powers for various weight changes using a haptic device. *International Conference on Intelligent Robots and Systems, IEEE/RSJ*, Acropolis convention center, Nice, France, Sept 22-26, pp. 3971-3976.
- [144] Vo, D.M., Vance, J.M., y Marasinghe, M.G. (2009). Assessment of haptics-based interaction for assembly tasks in virtual reality. *EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. World Haptics, pp. 494-499.
- [145] Visualization Toolkit, "VTK", Consultado el 17 de enero de 2014 en: <http://www.vtk.org/>.
- [146] Geomagic OpenHaptics, "OpenHaptics Toolkit v3.0". Consultado el 17 de enero de 2014 en: <http://geomagic.com/en/products/open-haptics/overview>.

- [147] Gonzalez, G., Medellín, H.I., Lim, T. y Ritchie, J.M. (2012). 3D object representation for physics simulation engines and its effect on virtual assembly tasks. *Proceedings of the ASME 2012 IDETC/CIE*, agosto 12-15, Chicago, IL, USA, pp. 1449-1459.
- [148] Mamou, K. y Ghorbel F. (2009). A simple and efficient approach for 3D mesh approximate convex decomposition. *16th IEEE International Conference on Image Processing (ICIP '09)*, Cairo, Nov. 7-10, pp. 3501-3504.
- [149] GIMPACT, “GIMPACT, geometric tools for VR”, Consultado el 17 de enero de 2014 en: [http://gimpact.sourceforge.net/reference\\_html/features.html](http://gimpact.sourceforge.net/reference_html/features.html).
- [150] Ratcliff J., “John Ratcliff’s code suppository”, Consultado el 8 de noviembre de 2011 en: <http://codesuppository.blogspot.com/2006/04/approximate-convex-decomposition.html>.
- [151] Xiuzhi, Q., y Brent, S. (2005). Circular hole recognition for STL-based toolpath generation. *Rapid Prototyping Journal*, vol. 11(3), pp. 132-139.
- [152] Nvidia, Developer Z. “PhysX Visual Debugger”, Consultado el 17 de enero de 2014 en: <http://developer.nvidia.com/physx-visual-debugger>.
- [153] Meshlab.com, “Meshlab”, Consultado el 17 de enero de 2014 en: <http://meshlab.sourceforge.net/>.

## Cuestionario de evaluación del sistema HAMS (Haptic Assembly and Manufacturing System)

Nombre: \_\_\_\_\_

Nivel de estudios: \_\_\_\_\_

Edad \_\_\_\_\_

Sexo \_\_\_\_\_

Diestro o Zurdo \_\_\_\_\_

Orden: PBM-CBM \_\_\_\_\_ CBM-PBM \_\_\_\_\_

1. Completa la siguiente tabla, indicado en cada caso años de experiencia:

| Tienes experiencia en:           | Alta | Media | Nula |
|----------------------------------|------|-------|------|
| Video-juegos                     |      |       |      |
| CAD / CAE (NX, solidworks, etc.) |      |       |      |
| Aplicaciones de realidad virtual |      |       |      |
| Dispositivos hápticos            |      |       |      |
| Diseño de componentes mecánicos  |      |       |      |
| Diseño de software               |      |       |      |

## **Ensamble de la bomba de engranes sin DAC**

1. Contesta las siguientes preguntas:

**a) ¿Qué tan fácil fue controlar el dispositivo háptico? \_\_\_\_\_**

- 0 → Muy difícil
- 1 → Difícil
- 2 → Medio
- 3 → Fácil
- 4 → Muy fácil

**b) ¿Qué tan fácil fue realizar el ensamble de la bomba de engranes en el sistema virtual? \_\_\_\_\_**

- 0 → Muy difícil
- 1 → Difícil
- 2 → Medio
- 3 → Fácil
- 4 → Muy fácil

**c) ¿Cómo fue la estabilidad de los objetos virtuales? \_\_\_\_\_**

- 0 → Muy inestable, los objetos saltan mucho y se alejan de su posición
- 1 → Inestable, los objetos saltan ocasionalmente
- 2 → Medio, los objetos saltan poco pero no afecta el proceso de ensamble
- 3 → Estable, Los objetos no presentan saltos y se comportan como objetos reales

**d) ¿Qué tan parecido a lo real fue el proceso de ensamble? \_\_\_\_\_**

- 0 → Bajo, los objetos no reaccionan y se traslapan unos con otros
- 1 → Medio, existe reacciones entre los objetos pero no corresponden a la realidad
- 2 → Medio-Alto, los objetos reaccionan correctamente pero en ocasiones no lo hacen
- 3 → Alto, los objetos reaccionan como lo harían en la realidad

**e) ¿Qué tanto siente a través del dispositivo háptico cuando el objeto manipulado choca con los demás objetos? \_\_\_\_\_**

- 0 → No se sienten las colisiones
- 1 → Las colisiones se sienten muy débiles, no reales
- 2 → Las colisiones se sienten notablemente pero no ayudan al ensamble
- 3 → Las colisiones se sienten de manera clara y ayudan al proceso de ensamble

**f) ¿Considera que el tiempo de ensamble virtual es similar al tiempo de ensamble real? \_\_\_\_\_**

- 0 → No, el proceso virtual toma mucho tiempo.
- 1 → Son similares pero el ensamble virtual toma un poco más tiempo que el real.
- 2 → El ensamble virtual se realiza en el mismo tiempo que el real.
- 3 → El ensamble virtual es más rápido que el ensamble real.

## **Ensamble de la bomba de engranes utilizando DAC**

1. Contesta las siguientes preguntas:

**a) ¿Qué tan fácil fue controlar el dispositivo háptico? \_\_\_\_\_**

- 0 → Muy difícil
- 1 → Difícil
- 2 → Medio
- 3 → Fácil
- 4 → Muy fácil

**b) ¿Qué tan fácil fue realizar el ensamble de la bomba de engranes en el sistema virtual? \_\_\_\_\_**

- 0 → Muy difícil
- 1 → Difícil
- 2 → Medio
- 3 → Fácil
- 4 → Muy fácil

**c) ¿Cómo fue la estabilidad de los objetos virtuales? \_\_\_\_\_**

- 0 → Muy inestable, los objetos saltan mucho y se alejan de su posición
- 1 → Inestable, los objetos saltan ocasionalmente
- 2 → Medio, los objetos saltan poco pero no afecta el proceso de ensamble
- 3 → Estable, Los objetos no presentan saltos y se comportan como objetos reales

**d) ¿Qué tan parecido a lo real fue el proceso de ensamble? \_\_\_\_\_**

- 0 → Bajo, los objetos no reaccionan y se traslapan unos con otros
- 1 → Medio, existe reacciones entre los objetos pero no corresponden a la realidad
- 2 → Medio-Alto, los objetos reaccionan correctamente pero en ocasiones no lo hacen
- 3 → Alto, los objetos reaccionan como lo harían en la realidad

**e) ¿Qué tanto siente a través del dispositivo háptico cuando el objeto manipulado choca con los demás objetos? \_\_\_\_\_**

- 0 → No se sienten las colisiones
- 1 → Las colisiones se sienten muy débiles, no reales
- 2 → Las colisiones se sienten notablemente pero no ayudan al ensamble
- 3 → Las colisiones se sienten de manera clara y ayudan al proceso de ensamble

**f) ¿Considera que el tiempo de ensamble virtual es similar al tiempo de ensamble real? \_\_\_\_\_**

- 0 → No, el proceso virtual toma mucho tiempo.
- 1 → Son similares pero el ensamble virtual toma un poco más tiempo que el real.
- 2 → El ensamble virtual se realiza en el mismo tiempo que el real.
- 3 → El ensamble virtual es más rápido que el ensamble real.



## **Percepción Global del sistema**

- 1. En términos generales, ¿Qué le pareció la experiencia virtual – háptica?, seleccione la que mejor la describa: \_\_\_\_\_**
  - a. Aburrida
  - b. Agotadora
  - c. Estresante
  - d. Interesante
  - e. Útil
  - f. Divertida
  
- 2. En cuanto a funcionamiento, ¿Qué tan preciso le pareció el sistema? \_\_\_\_\_**
  1. *No tiene precisión*, los objetos se intersectan y los ensambles no se pueden realizar, no se siente la fuerza en el dispositivo háptico.
  2. *Precisión baja*, los objetos no se intersectan pero las reacciones entre los objetos y la fuerza del dispositivo háptico no son reales.
  3. *Precisión media*, los objetos actúan como lo hacen en la realidad, pero la fuerza de retroalimentación háptica en ocasiones no es perceptible.
  4. *Precisión alta*, Los objetos se comportan exactamente como en la realidad, y cualquier fuerza es perceptible mediante el dispositivo háptico.
  
- 3. Que tanta confianza sintió al utilizar el sistema: \_\_\_\_\_**
  - a. Baja, No se entiende qué es lo que se debe de hacer, ni cómo utilizarlo, existe nerviosismo
  - b. Media, En ocasiones durante un ensamble, no se sabe el siguiente paso o que es lo que se tiene que hacer.
  - c. Alta, El sistema es intuitivo y amigable, no deja lugar a dudas.
  
- 4. ¿Los movimientos que hacía con la mano, corresponden al comportamiento virtual? \_\_\_\_\_**
  1. No, los objetos se mueven de manera diferente a como se mueve la mano.
  2. A veces, los objetos se trasladan conforme los movimientos de la mano, pero las rotaciones no son hechas conforme al movimiento de la mano
  3. Si, los objetos se mueven de acuerdo a los movimientos de la mano.
  
- 5. ¿Considera que el tiempo de ensamble virtual debe ser igual al tiempo de ensamble real para que el sistema pueda ser utilizado de manera práctica, es decir en casos reales?**  
Si \_\_\_\_\_  
No \_\_\_\_\_
  
- 6. ¿Considera que un sistema como el evaluado pueda ser utilizado en planeaciones o análisis reales de ensamble?**  
Si \_\_\_\_\_  
No \_\_\_\_\_
  
- 7. ¿Qué ventajas y desventajas considera que tiene el sistema?**

**8. ¿Qué considera importante para que el sistema pueda ser utilizado en casos reales?****Ordene de mayor a menor importancia.**

- a. Objetos más complejos
- b. Escenas más complejas.
- c. Mayor nivel de inmersión gráfica.
- d. Mejor interacción háptica.
- e. Mayor rapidez de la simulación.
- f. Que los objetos tengan una visualización más real.

**9. ¿En qué áreas o aplicaciones se pudiera aplicar este tipo de sistemas?****10. ¿Qué otras funciones agregaría al sistema para hacerlo más funcional y/o eficiente?**