



Universidad
Carlos III de Madrid

Departamento de Ingeniería Mecánica

PROYECTO FIN DE CARRERA
INGENIERÍA TÉCNICA INDUSTRIAL
MECÁNICA

**Modelado y simulación virtual de un
diferencial mecánico en entorno VRML**

Leganés, Julio 2012

Autor: D. JESÚS SERRANO TENORIO

Tutora: Dra. Dña. BEATRIZ LÓPEZ BOADA
Directora: Dra. Dña. MARIA JESÚS LÓPEZ BOADA

Título: Modelado y simulación virtual de un diferencial mecánico en entorno VRML

Autor: D. JESÚS SERRANO TENORIO

Tutora: Dra. Dña. BEATRIZ LÓPEZ BOADA

Directora: Dra. Dña. MARIA JESÚS LÓPEZ BOADA

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día __ de _____
de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de
Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

A Beatriz y María Jesús por vuestra paciencia e inestimable ayuda.

A mis padres, por enseñarme a luchar y mirar siempre hacia delante.

A mis hermanos, porque siempre habéis sido y seréis un modelo a seguir para mí.

A Angie, por creer en mí y porque contigo he encontrado mi lugar en el mundo.

Y sobre todo a ti, Jesusito, porque eres la luz de mis días.

Resumen

Este proyecto describe el funcionamiento de un conjunto mecánico diferencial real, cedido por el Departamento de Ingeniería Mecánica de la Universidad Carlos III de Madrid, mediante simulación virtual en entorno VRML (*Virtual Reality Modeling Language*).

Además, a lo largo del proyecto, se describen temas tan importantes como la aplicación de las nuevas tecnologías a la enseñanza, y la realidad virtual como herramienta pedagógica. Así como una breve descripción de las transmisiones de ruedas dentadas y su aplicación al conjunto mecánico diferencial, tipos de diferenciales mecánicos y las mejoras que se están aplicando en la actualidad.

Índice

1. CAPÍTULO 1: INTRODUCCIÓN Y OBJETIVOS	12
1.1. Introducción.....	12
1.2. Objetivo del proyecto	13
1.3. Medios empleados	13
1.4. Estructura del proyecto.....	13
2. CAPÍTULO 2: LA DOCENCIA Y LAS T. I. C.	15
2.1 Introducción.....	15
2.1.1 La necesidad de una enseñanza nueva.....	15
2.1.2 La red de redes.....	16
2.1.3 La globalización del PC.....	17
2.2 Software educativo y E-learning	19
2.2.1 Características distintivas del E-learning	20
2.2.2 Ventajas e inconvenientes	21
2.3 La Realidad Virtual	21
2.3.2 Tipos de realidad virtual	22
2.3.3 Problemas e inconvenientes actuales de los entornos de Realidad Virtual	25
2.3.4 Futuro de la Realidad Virtual	26
3. CAPÍTULO 3: EL DIFERENCIAL MECÁNICO	27
3.1 Teoría de engranajes.....	28
3.1.1 Tipos de Engranajes	28
3.1.1.1 Engranajes para ejes paralelos.....	28
3.1.1.2 Engranajes para ejes que se cortan en el espacio.....	30
3.1.1.3 Engranajes para ejes que se cruzan en el espacio: hiperbólicos.....	31
3.1.2 Partes de los Engranajes	33
3.2 El mecanismo diferencial	36
3.2.1 Descripción del diferencial mecánico libre	37
3.2.2 Diferenciales 100% bloqueables	40
3.2.3 Diferenciales autoblocantes.....	40
3.2.3.1 Diferencial autoblocante por discos de fricción	40
3.2.3.2 <i>Salisbury axle</i> o diferencial de rampas	42
3.2.3.3 Diferencial viscoso o Ferguson	43
3.2.3.4 Diferencial Torsen	44
4. CAPÍTULO 4: MODELADO DEL DIFERENCIAL CON SOLID EDGE	47
4.1. Sistemas CAD-CAE-CAM-CAPP	47
4.1.1. CAD (Computer Aided Design).....	48
4.1.2. CAE (Computer Aided Engineering)	49
4.1.3. CAM (Computer Aided Manufacturing).....	50
4.1.4. CAPP (Computer Aided Process Planning)	51
4.2. Descripción del programa Solid Edge V.18	52
4.2.1 El interfaz Solid Edge V.18 en entorno “pieza sólida”	52
4.2.2 El interfaz SOLID EDGE V.18 en entorno “conjunto”.....	55
4.2.3 Operaciones principales de modelado	57
4.3 Modelado de las piezas del diferencial.....	59

4.3.1 Modelado de la corona hipoidal	60
4.3.2 Modelado del casquete	62
4.3.3 Modelado del planetario	65
4.3.4 Modelado del satélite.....	66
4.3.5 Modelado del piñón de ataque.....	67
4.4 Ensamblaje del conjunto mecánico diferencial	69
5. CAPÍTULO 5: ANIMACIÓN EN ENTORNO VRML	71
5.1 Inicios del VRML y su evolución.....	71
5.2 Metodología de trabajo con VRML.....	74
5.3 Características del lenguaje	76
5.3.1. Software necesario para trabajar con VRML	76
5.3.2. Normas básicas de sintaxis en VRML.....	77
5.3.3. Estructura básica.....	77
5.3.4. Descripción de los nodos.....	80
5.3.4.1 Nodo <i>Shape</i>	80
5.3.4.2 Nodo <i>Group</i>	82
5.3.4.3 Nodo <i>Inline</i>	82
5.3.4.4 Nodo <i>Transform</i>	83
5.3.4.5 Nodo <i>TimeSensor</i>	84
5.3.4.6 Nodo <i>TouchSensor</i>	85
5.3.4.7 Interpoladores	86
5.3.4.8 Nodo <i>Viewpoint</i>	86
5.3.4.9 Nodos <i>Event</i> y <i>Route</i>	87
5.3.4.10 Nodos <i>DEF</i> y <i>USE</i>	88
5.4 Programación en VRML del movimiento del diferencial	88
5.4.1 Identificación de grupos fijos	88
5.4.2 Ensamblaje del conjunto en VRML	91
5.4.2.1 Inserción al entorno VRML del Grupo Casquete-Corona.....	91
5.4.2.2 Inserción al entorno VRML de los Grupos Eje-Planetario	93
5.4.2.3 Inserción al entorno VRML del piñón de ataque	94
5.4.3 Animación del diferencial mecánico en VRML.....	95
5.4.3.1 Programación de los sensores táctiles	95
5.4.3.2 Programación de los cronómetros de movimiento	96
5.4.3.3 Programación del movimiento del diferencial en recta	97
5.4.3.4 Definición de movimiento del diferencial en bloqueo	102
5.4.3.5 Definición de movimiento del diferencial en curva	105
5.5 Resultados obtenidos	106
6. CAPÍTULO 6: CONCLUSIONES Y TRABAJOS FUTUROS	110
6.1 Conclusiones.....	110
6.2 Trabajos futuros.....	111
7. REFERENCIAS	112

Índice de figuras

Fig. 2. 1: Sensores de posición para entornos de realidad virtual	24
Fig. 2. 2: Sensores de posición en extremidades	24
Fig. 3. 1: Diferencial mecánico libre	27
Fig. 3. 2: Dentado recto interno	28
Fig. 3. 3: Dentado recto externo	29
Fig. 3. 4: Helicoidal simple	29
Fig. 3. 5: Helicoidal doble	29
Fig. 3. 6: Herringbone	30
Fig. 3. 7: Cónicos rectos	30
Fig. 3. 8: Cónico helicoidal	31
Fig. 3. 9: Helicoidal cruzado.	31
Fig. 3. 10: Corona Hipoidal	32
Fig. 3. 11: Partes de una rueda dentada	33
Fig. 3. 12: Esquema de engrane.....	35
Fig. 3. 13: Sección de una transmisión completa de automóvil.....	36
Fig. 3. 14: Despiece explosionado de un diferencial mecánico libre.....	37
Fig. 3. 15: Partes de un diferencial	38
Fig. 3. 16: Repartos de par.....	39
Fig. 3. 17: Estructura interna de un diferencial autoblocante por discos de fricción.....	41
Fig. 3. 18: Diferencial de rampas	42
Fig. 3. 19: Esquema interno de un viscoacoplador	44
Fig. 3. 20: Vista en corte de un diferencial Torsen.....	45
Fig. 3. 21: Engranajes internos del Torsen	46
Fig. 4. 1: Entornos de trabajo	52
Fig. 4. 2: Interfaz entorno pieza sólida	53
Fig. 4. 3: Barra de herramientas de archivo.....	53
Fig. 4. 4: Barra de opciones de representación	53
Fig. 4. 5: Desplegable para selección de plano de trabajo	54
Fig. 4. 6: Barra de operaciones de modelado.....	54
Fig. 4. 7: Lista de operaciones.....	54
Fig. 4. 8: Planos de referencia	55
Fig. 4. 9: Interfaz del entorno conjunto	55
Fig. 4. 10: Barra de operaciones del entorno conjunto	56
Fig. 4. 11: Desplegable opciones de ensamblaje	56
Fig. 4. 12: Icono Boceto	57
Fig. 4. 13: Icono Protusión por Proyección	57
Fig. 4. 14: Protusión por revolución	57
Fig. 4. 15: Protusión por barrido.....	58
Fig. 4. 16: Protusión helicoidal.....	58
Fig. 4. 17: Vaciado por revolución	58
Fig. 4. 18: Patrón	58
Fig. 4. 19: Plano Coincidente	58
Fig. 4. 20: Diferencial real a modelar	59
Fig. 4. 21: Diferencial real a modelar	59
Fig. 4. 22: Protusión por revolución de la corona.....	60
Fig. 4. 23: Vaciado por barrido del hueco interdental	60
Fig. 4. 24: Repetición del vaciado con la herramienta patrón	61
Fig. 4. 25: Repetición de la operación agujero	61
Fig. 4. 26: Corona modelada	62
Fig. 4. 27: corona real.....	62
Fig. 4. 28: Protusión por revolución para el casquete.....	63
Fig. 4. 29: Protusión de salientes laterales.....	63

Fig. 4. 30: Generación de agujeros.....	64
Fig. 4. 31: Casquete real.....	64
Fig. 4. 32: Protusión por revolución de los planetarios.....	65
Fig. 4. 33: Planetario modelado.....	65
Fig. 4. 34: Planetario real.....	66
Fig. 4. 35: Modelado del satélite.....	66
Fig. 4. 36: Vaciado interdental de los satélites.....	67
Fig. 4. 37: Protusión por revolución y protusión helicoidal cónica.....	67
Fig. 4. 38: Piñón de ataque modelado.....	68
Fig. 4. 39: Piñón de ataque real.....	68
Fig. 4. 40: Ensamblaje final del diferencial.....	69
Fig. 4. 41: Vista general del conjunto.....	70
Fig. 5. 1: Código de generación de un cubo.....	80
Fig. 5. 2: Código de generación de una esfera.....	80
Fig. 5. 3: Código de generación de un cono.....	81
Fig. 5. 4: Código de generación de un cilindro.....	81
Fig. 5. 5: Código del nodo genérico Group.....	82
Fig. 5. 6: Código del nodo genérico Inline.....	83
Fig. 5. 7: Código del nodo genérico Transform.....	83
Fig. 5. 8: Código del nodo genérico Timesensor.....	85
Fig. 5. 9: Código del nodo genérico Touchsensor.....	85
Fig. 5. 10: Código del nodo OrientatorInterpolator.....	86
Fig. 5. 11: Código del nodo genérico Viewpoint.....	87
Fig. 5. 12: Grupo Casquete-Corona importado a VRML.....	89
Fig. 5. 13: Grupo eje-planetario importado a VRML.....	90
Fig. 5. 14: Piñón de ataque importado a VRML.....	90
Fig. 5. 15: Satélite Importado a VRML.....	91
Fig. 5. 16: Código de Integración del Grupo Casquete –Corona y Satélites.....	92
Fig. 5. 17: Grupo Casquete-Corona y Satélites en VRML.....	92
Fig. 5. 18: Código de integración del Grupo Eje-Planetario en VRML.....	93
Fig. 5. 19: Grupo Eje-planetario ensamblado en VRML.....	93
Fig. 5. 20: Integración del piñón de ataque en el entorno VRML.....	94
Fig. 5. 21: Piñón de ataque ensamblado en VRML.....	94
Fig. 5. 22: Código para la inserción de sensores táctiles.....	95
Fig. 5. 23: Sensores táctiles de activación de regímenes.....	96
Fig. 5. 24: Código para la programación de los cronómetros de movimiento.....	97
Fig. 5. 25: Notación.....	97
Fig. 5. 26: Posición de cada grupo en función de los intervalos de tiempo en recta.....	101
Fig. 5. 27: Rutas caso diferencial en recta.....	102
Fig. 5. 28: Posición de cada grupo en función de los intervalos de tiempo en bloqueo.....	103
Fig. 5. 29: Posición de cada grupo en función de los intervalos de tiempo en bloqueo.....	104
Fig. 5. 30: Rutas de enlace del diferencial en bloqueo.....	104
Fig. 5. 31: Posición de cada grupo en función de los intervalos de tiempo en curva.....	105
Fig. 5. 32: Posición de cada grupo en función de los intervalos de tiempo en curva.....	106
Fig. 5. 33: Rutas de enlace del diferencial en curva.....	106
Fig. 5. 34: Presentación del simulador.....	107
Fig. 5. 35: Cursores del visualizador Cortona.....	107
Fig. 5. 36: Visión de detalle1.....	108
Fig. 5. 37: Visión de detalle 2.....	108
Fig. 5. 38: Visión de detalle 3.....	109
Fig. 5. 39: Visión de detalle 4.....	109

Índice de tablas

Tabla 5. 1: Nodos 1	78
Tabla 5. 2: Nodos 2	78
Tabla 5. 3: Campos de propiedades.....	79
Tabla 5. 4: Radios de las piezas del diferencial.....	98
Tabla 5. 5: Notación de los parámetros cinemáticos	98

CAPÍTULO 1: INTRODUCCIÓN Y OBJETIVOS

1.1. Introducción

La educación actual afronta múltiples retos. Uno de ellos es dar respuesta a los profundos cambios sociales, económicos y culturales que se prevén para la "sociedad de la información". Actualmente se está empezando a aplicar, de forma generalizada, las nuevas tecnologías de la información en todos los ámbitos de la sociedad.

Por ello, las técnicas de enseñanza cambian, y surgen nuevas herramientas que facilitan el aprendizaje al alumno, completan el trabajo docente al profesor y que abren nuevas posibilidades de formación a distancia. Una de esas herramientas es la Realidad Virtual. La interacción entre el alumno y un entorno virtual, relacionado con cualquier materia de aprendizaje, facilita el proceso de comprensión de dicha materia, complementando o sustituyendo métodos tradicionales de enseñanza.

La Teoría de Vehículos es una materia dedicada, entre otras cosas, a la descripción y el funcionamiento de los elementos que conforman un automóvil. La mayoría de las veces dichos elementos son inaccesibles, y por esta razón, el uso de entornos virtuales es útil para conseguir que el alumno asimile los conceptos.

1.2. Objetivo del proyecto

El objetivo de este proyecto es simular un conjunto mecánico diferencial real en un entorno de realidad virtual para que sea utilizado con fines docentes.

Se pretende que el usuario adquiera nociones básicas de lo que es un diferencial, que identifique sus piezas, y que comprenda de manera intuitiva su funcionamiento mediante la interacción con el entorno virtual.

1.3. Medios empleados

Los medios necesarios para llevar a cabo el proyecto son tres programas:

- Solid Edge v.18, versión académica, es un programa de diseño mecánico que permite la representación de piezas, conjuntos en 3D y generación de planos. Dicho programa es necesario para modelar las piezas del diferencial, ya que modelar las piezas en VRML dificultaría enormemente el trabajo y no se obtendrían los resultados esperados.
- El Virtual Reality Modeling Language, más conocido como VRML, es un lenguaje de modelado de mundos virtuales en tres dimensiones que permite generar, dar movimiento e interactuar en ellos.
- VrmIPAD es un editor de VRML que facilita la programación mediante herramientas como rápido autocompletado y detección dinámica de errores entre otras funciones.

1.4. Estructura del proyecto

El proyecto se distribuirá en los siguientes capítulos:

Capítulo 1: Se establece el objetivo del proyecto, así como los medios empleados.

Capítulo 2: En una primera parte, se exponen los factores que han dado lugar el uso de las nuevas tecnologías en el ámbito docente, conceptos de software educativo y *E-learning*. Y en la segunda parte, se introduce al lector a la Realidad Virtual, se hace una clasificación basada en la interfaz del usuario, se describen ventajas y desventajas, y finalmente se muestra el desarrollo de las nuevas tecnologías de Realidad Virtual.

Capítulo 3: En este capítulo se describe el diferencial mecánico libre y los elementos que lo conforman. Para ello, se realiza una breve descripción de la teoría de engranajes en la que se basa este tipo de mecanismo. Después se explica el funcionamiento de forma detallada, sus ventajas e inconvenientes y, por último, se muestran diferenciales mejorados.

Capítulo 4: Consta de una primera parte, en la que se explican los diferentes tipos de diseño asistido por ordenador. Después, una segunda parte, en la que presenta el programa de modelado en tres dimensiones Solid Edge. Y finalmente se explica el proceso de modelado del diferencial real.

Capítulo 5: En este capítulo se presenta VRML como lenguaje de definición de entornos de realidad virtual y su historia. Posteriormente se marcan las reglas básicas de creación de mundos virtuales y por último se describe el código específico utilizado para crear la simulación virtual del diferencial mecánico en diferentes regímenes de funcionamiento.

Capítulo 6: Se enumeran los resultados finales obtenidos, se explican las conclusiones y se proponen trabajos futuros en los que se puede desarrollar el presente proyecto.

CAPÍTULO 2: LA DOCENCIA Y LAS T. I. C.

2.1 Introducción

La educación actualmente está afrontando múltiples retos. Uno de ellos es dar respuesta a los profundos cambios sociales, económicos y culturales que se prevén para la "Sociedad de la Información". Las nuevas tecnologías de la información y la comunicación (TIC) han generado un enorme interés en todos los ámbitos de la sociedad. Su utilización con fines educativos es un campo abierto a la reflexión y a la investigación.

2.1.1 La necesidad de una enseñanza nueva

En los últimos años, la enseñanza abierta y a distancia ha despertado un interés considerable a todos los niveles. El artículo 126 del Tratado de Maastricht, por ejemplo, establece la promoción de esta modalidad como prioridad en la Unión Europea. Las razones de este interés son diversas, pero las más importantes son la necesidad de una formación personalizada y la facilidad al acceso a las nuevas tecnologías.

Existe la convicción generalizada de que las instituciones tradicionales, de ladrillos y cemento, no serán suficientes para responder al desafío en materia de formación, inicial y permanente, inherente a la sociedad de la información [1]. Es necesario desarrollar nuevas modalidades educativas ajustadas a las necesidades y posibilidades de un público adulto que no puede desplazarse hasta los centros de formación por sus obligaciones familiares, laborales o personales. La formación debe flexibilizarse para acomodarse a necesidades crecientemente diversificadas (formación a la carta "*just in time*", en el puesto de trabajo, etc.).

La disponibilidad generalizada de las nuevas tecnologías interactivas de la información y la comunicación abre una inmensa cantidad de posibilidades que se concretan en el desarrollo de nuevos modelos pedagógicos en la formación a distancia. Tradicionalmente basada en la actividad del alumno sobre materiales impresos estandarizados, las nuevas tecnologías enriquecen la formación a distancia con la posibilidad no sólo de difundir información de modo barato y eficiente, sino de dotar a los participantes (profesores, alumnos, expertos, etc.) de herramientas hardware/software para la comunicación personal y grupal que refuercen la acción tutorial y el aprendizaje de doble vía.

2.1.2 La red de redes

De entre las nuevas tecnologías de la información y la comunicación, la que más ha impactado en todos los sectores sociales, culturales y económicos en los últimos años ha sido la de las redes informáticas y, especialmente, Internet. El número de usuarios aumenta de forma continua, en 2011 se alcanzaron los 2000 millones de internautas [2]. Universidades, centros de investigación, instituciones privadas, organismos públicos, empresas y particulares participan de una experiencia tecnológica y social inédita en la historia de la humanidad: Internet es el primer medio de comunicación de masas bidireccional.

Internet encarna una utopía comunicativa en la que toda la información está al alcance de cualquier persona en cualquier momento y en cualquier lugar, una comunidad virtual que puede comunicarse sin, prácticamente, otra limitación que su imaginación.

Sin embargo, el Internet real se ha desarrollado al nivel actual gracias a las expectativas de beneficios económicos de muchos de los participantes en el juego: un enorme y nuevo mercado de bienes y servicios. Pero junto a dichos intereses económicos, Internet es el escenario de otras actividades: informar y prestar servicios a los ciudadanos, difundir ideas y pensamientos, comunicar con otras personas y, evidentemente, educar [1].

2.1.3 La globalización del PC

El tipo de ordenador con el que se comenzó la introducción masiva de las nuevas tecnologías en los sistemas educativos, en la década de los años 80, reunía una serie de características tecnológicas que necesariamente limitaban su grado de aplicación en la educación.

A continuación se enumeran algunas de esas limitaciones [1]:

- Los ordenadores de los que se disponían eran **máquinas aisladas**, monopuesto y monousuario, no conectadas en red, con lo cual sus posibilidades de comunicación eran prácticamente nulas. Además, las líneas de comunicaciones analógicas permitían unas velocidades de interconexión mínimas y sus costes resultaban prohibitivos. En definitiva, no eran herramientas diseñadas para la comunicación ni entre máquinas ni entre humanos.
- Contaba con **recursos de almacenamiento limitados y poco fiables** debido a las características de los medios empleados (discos flexibles de 5 ¼ pulgadas de menos de 1 Mb de capacidad y de conservación delicada).
- Las capacidades multimedia estaban restringidas apenas a algunos sonidos y a modestas animaciones gráficas 2D.
- Los **ordenadores estaban muy vinculados al sistema operativo propietario** que el fabricante instalaba. Mayoritariamente fue DOS el que se impuso y, a su vez, tanto las aplicaciones de ofimática (primer uso inducido de la informática en educación en el que se pensó) como los programas específicos con finalidad educativa que se diseñaban eran completamente dependientes del sistema

operativo y de la arquitectura del ordenador sin ninguna posibilidad de trasladarlos a otros entornos.

En resumen, a pesar del empeño de muchos y de las altas dosis de creatividad de otros que se emplearon a fondo en buscar aplicaciones educativas para el ordenador, el interfaz, el aspecto y las posibilidades de utilización de los productos finales que se desarrollaban eran muy limitadas, se preponderaban las aplicaciones de enseñanza asistida clásicas en formatos pregunta/respuesta llevadas a cabo en la Universidad de Stanford y las simulaciones más o menos conseguidas con los medios existentes.

Internet, pero no sólo esta red de comunicaciones sino también los avances en la tecnología de dispositivos móviles y en otras líneas de desarrollo de periféricos ligados al ordenador (cámaras digitales, dispositivos de domótica, escáneres, etc.), han contribuido conjuntamente a la evolución tecnológica del ordenador personal. Las nuevas máquinas, además de ser dispositivos más veloces y potentes, deben ser concebidas como herramientas funcionalmente distintas.

Se pueden destacar cinco características comunes de todos los ordenadores que se fabrican en la actualidad [1]:

- La elección de **arquitectura y de sistema operativo tiene cada vez menor peso** en el conjunto del equipo. Aunque todavía hoy, en el ámbito de la informática personal, se guarda una compatibilidad básica en cuanto a arquitectura y a nivel de instrucciones del microprocesador con los antiguos ordenadores compatibles, dada la tendencia creciente de desarrollo de aplicaciones informáticas accesibles a través de navegadores y en lenguajes multiplataforma se está evitando los problemas de incompatibilidad. Se ha evolucionado en los interfaces de usuario, ahora siempre gráficos y cada vez más ergonómicos y similares entre sí independientemente del dispositivo y en la posibilidad de elección entre distintos sistemas operativos pudiéndose optar por productos comerciales pero también por software libre y gratuito. Parece que se camina a una especie de dispositivo universal de proceso con apariencias externas muy distintas pero con funcionalidades similares (correo, acceso a Internet, agenda, ofimática, etc.).

- Nacen para estar conectados a redes internas y externas de medio y alto ancho de banda. Por estas redes es posible transmitir flujos de datos que **permiten recepciones y emisiones audiovisuales de calidad** suficiente para muchos fines y las tecnologías que lo hacen posible están estandarizadas y son asequibles. Los ordenadores y dispositivos derivados se configuran así en auténticos centros de comunicación y de control universales que están modificando las formas de interrelación de los humanos.
- Cada vez se cuenta con una **capacidad de almacenamiento más elevada** para todo tipo de datos y los ordenadores ocupan nichos tecnológicos y comerciales de otros dispositivos de forma progresiva (por ejemplo el DVD en lectura y escritura que ha sustituido a los magnetoscopios analógicos en VHS).
- Se transforman de forma versátil en **terminales de comunicaciones** capaces de digerir cualquier sistema imaginable: fax, mensajería instantánea, voz-IP, videoconferencia.
- Con los adecuados periféricos de **bajo coste**, se transforman en dispositivos de acceso a televisión y radio analógicas y digitales y a servicios de datos por satélite. Aparte de las posibilidades propias de las tarjetas gráficas y de sonido actuales, es sencilla su interconexión con televisores y dispositivos de proyección de gran formato y con equipos de alta fidelidad. La edición de video digital de alta calidad es uno de los ámbitos de más reciente integración en lo que se considera hoy día un PC estándar de uso doméstico.

2.2 Software educativo y E-learning

Como es lógico, todos los adelantos tecnológicos, tanto en hardware como en software, no tardaron en entrar en las aulas con la instalación de puestos de ordenador de última generación, ya sea para aprender matemáticas (como por ejemplo el programa Derive), lenguajes de programación que ayudan al estudiante a estructurar su forma de pensar (por ejemplo el lenguaje de programación Pascal), programas de dibujo (Autocad), y algunos más. Pero no hay que confundir este tipo de formación en las aulas con el E-learning.

Como definición de E-learning, según la Dirección General de Telecomunicaciones de Teleeducación, es el desarrollo del proceso de formación a distancia (reglada o no reglada), basado en el uso de las tecnologías de la información y las telecomunicaciones, que posibilitan un aprendizaje interactivo, flexible y accesible, a cualquier receptor potencial. En este informe se entiende por *teleeducación* «una enseñanza a distancia, abierta, flexible e interactiva basada en el uso de las nuevas tecnologías de la información y de la comunicación, y de las comunicaciones, y sobre todo aprovechando los medios que ofrece la red Internet» [3].

Es decir, esta forma de educación promueve la formación autónoma, para lo cual es necesario haber enseñado a los alumnos a aprender.

2.2.1 Características distintivas del E-learning

Para saber como funciona la formación basada en E-learning se van a enumerar a continuación sus características fundamentales [3]:

- La característica principal es que permite al alumno ir a su propio ritmo.
- Es un tipo de aprendizaje asistido por ordenador conectado a Internet con el que es posible el uso de protocolos TCP y HTTP para facilitar la comunicación entre el estudiante, el material didáctico, recursos docentes, y el profesor que normalmente está separado del alumno en el espacio y en el tiempo.
- Es fundamental el uso de herramientas de comunicación síncronas y asíncronas para poder vencer la barrera espacio-temporal con otros alumnos y/o con el profesor.
- Muy común es la utilización de hipermedia, que es resultado de la combinación de hipertexto y multimedia, donde hipertexto se entiende como la organización de una base de información conectados a través de enlaces cuya selección genera distintas formas de recuperar la información de la base; la multimedia consiste en la tecnología que utiliza la información almacenada en diferentes formatos y medios, controlados por un usuario.
- Otro de los rasgos es el almacenamiento, mantenimiento y administración del material didáctico y no didáctico en un servidor Web.

- Al no existir sesiones presenciales, este sistema de aprendizaje se apoya en tutorías para complementar los posibles vacíos que puedan existir en el material didáctico.

2.2.2 Ventajas e inconvenientes

El E-learning tiene muchas razones para establecerse en escuelas y universidades, pero es tan importante disponer de los medios técnicos como que el profesor y el alumno sepan en qué consiste este tipo de formación.

La ventaja más significativa es que facilita la autonomía del estudiante flexibilizando la información, independientemente del espacio y el tiempo. Además permite registrar la actividad realizada por los estudiantes en los servidores, facilita las discusiones, trabajos en grupo ahorrando tiempo, costos y desplazamiento [3].

Pero todas estas virtudes no son gratuitas, es decir, el alumno y el profesor deben establecer una relación de compromiso y esfuerzo ya que requiere más inversión de tiempo por parte de ambos. También es imprescindible que los estudiantes tengan habilidades para el aprendizaje autónomo. Los inconvenientes del proceso de formación mediante E-learning son la dependencia de una conexión a Internet, y por otro lado, los problemas de autenticación a la hora de evaluar a un alumno [3]. Son inconvenientes que progresivamente, con la experiencia, deben ser superados.

2.3 La Realidad Virtual

Dado que se trata de una tecnología en plena evolución, cualquier definición actual de Realidad Virtual debe ser considerada sólo con carácter transitorio, sin embargo, se puede decir que [4]:

“La realidad virtual es simulación por ordenador, dinámica y tridimensional, con alto contenido gráfico, acústico y táctil, orientada a la visualización de situaciones y

variables complejas, durante la cual el usuario ingresa, a través del uso de sofisticados dispositivos de entrada, a "mundos" que aparentan ser reales, resultando inmerso en ambientes altamente participativos, de origen artificial”.

Las características fundamentales de la realidad virtual son [5]:

- Se expresa en lenguaje gráfico tridimensional.
- Su comportamiento es dinámico y opera en tiempo real.
- Posee la capacidad de reaccionar ante el usuario, ofreciéndole, en su modalidad más avanzada, una experiencia inmersiva, interactiva y multisensorial.

Finalmente, lo que se consigue es la posibilidad de crear mundos nuevos, que se perciben mediante dispositivos audiovisuales, basados en objetos con los que se puede interaccionar y establecer la naturaleza de la relación entre los mismos.

En el ámbito de la educación, lo más útil de crear mundos virtuales es la capacidad de poder compartirlos y poder mostrar al los demás lo que antes se tenía que imaginar.

2.3.2 Tipos de realidad virtual

Existen diversas formas de clasificar los actuales sistemas de realidad virtual. A continuación se presenta una clasificación basada en el tipo de interfaz con el usuario. En ese caso pueden mencionarse [5]:

- **Sistema de Ventanas (*WOW: Window on World Systems*)**. Se han definido como sistemas de Realidad Virtual sin Inmersión. Algunos sistemas utilizan un monitor convencional para mostrar el mundo virtual. Estos sistemas son conocidos como *WOW (Window on World)* y también como Realidad Virtual de escritorio. Estos sistemas tratan de hacer que la imagen que aparece en la pantalla luzca real y que los objetos, en ella representada actúen con realismo.
- **Sistemas de mapeo por video**. Este enfoque se basa en la filmación, mediante cámaras de vídeo, de una o más personas y la incorporación de

dichas imágenes a la pantalla del ordenador, donde podrán interactuar, en tiempo real, con otros usuarios o con imágenes gráficas generadas por el ordenador. De esta forma, las acciones que el usuario realiza en el exterior de la pantalla (ejercicios, bailes, etc.) se reproducen en la pantalla del ordenador permitiéndole desde fuera interactuar con lo de dentro. El usuario puede, a través de este enfoque, simular su participación en aventuras, deportes y otras formas de interacción física.

- **Sistemas *immersivos*.** Los más perfeccionados sistemas de Realidad Virtual permiten que el usuario pueda sentirse "sumergido" en el interior del mundo virtual. El fenómeno de inmersión puede experimentarse mediante 4 modalidades diferentes, dependiendo de la estrategia adoptada para generar esta ilusión. Ellas son:
 - El operador aislado.
 - La cabina personal.
 - La cabina colectiva (*pods, group cab*).
 - La caverna o cueva (*cave*).

Estos sistemas inmersivos se encuentran generalmente equipados con un casco o máscara (Fig. 2.1), que contiene recursos visuales, en forma de dos pantallas miniaturas coordinadas para producir visión estereoscópica y recursos acústicos de efectos tridimensionales. Una variante de este enfoque lo constituye el hecho de que no exista casco como tal, sino un visor incorporado en una armadura que libera al usuario del casco, suministrándole una barra (como la de los periscopios submarinos) que permite subir, bajar o controlar la orientación de la imagen obtenida mediante el visor. Otra forma interesante de sistemas inmersivos se basa en el uso de múltiples pantallas de proyección de gran tamaño dispuestas ortogonalmente entre sí para crear un ambiente tridimensional o caverna (*cave*) en la cual se ubica a un grupo de usuarios. De estos usuarios, hay uno que asume la tarea de navegación, mientras los demás pueden dedicarse a visualizar los ambientes de Realidad Virtual dinamizados en tiempo real.



Fig. 2. 1: Sensores de posición para entornos de realidad virtual



Fig. 2. 2: Sensores de posición en extremidades

- **Sistemas de telepresencia (*Telepresence*).** Esta tecnología vincula sensores remotos en el mundo real con los sentidos de un operador humano (Fig. 2.2). Los sensores utilizados pueden hallarse instalados en un robot. De esta forma el usuario puede operar el equipo como si fuera parte de él. Esta tecnología posee un futuro extremadamente prometedor. La NASA se propone utilizarla como recurso para la exploración planetaria a distancia. La telepresencia contempla, obligatoriamente, un grado de inmersión que involucra el uso de control remoto, pero tiene características propias lo suficientemente discernibles como para asignarle una clasificación particular.
- **Sistemas de realidad mixta o aumentada.** Al fusionar los sistemas de telepresencia y realidad virtual se obtiene los denominados sistemas de Realidad Mixta. Aquí las entradas generadas por el ordenador se mezclan con entradas de telepresencia y/o la visión de los usuarios del mundo real. Este tipo de sistema se orienta a la estrategia de realzar las percepciones del operador o usuario con respecto al mundo real. Para lograr esto utiliza un tipo esencial de casco llamado HMD (*Helmet Mounted Display*) de visión transparente, que se apoya en el uso de una “combinadora” que es una pantalla especial, la cual es transparente a la luz que ingresa proveniente del mundo real, pero que a la vez refleja la luz apuntada a ella mediante los dispositivos ópticos ubicados en el interior del HMD. En este sentido se percibe un prometedor mercado para los sistemas de Realidad Mixta en industrias y fábricas donde el trabajador debe llevar a

cabo operaciones complejas de construcción o mantenimiento de equipos e instrumentos.

- **Sistemas de realidad virtual en pecera.** Este sistema combina un monitor de despliegue estereoscópico utilizando lentes LCD (*Liquid Crystal Display*) con obturador acoplados a un rastreador de cabeza mecánico. El sistema resultante es superior a la simple combinación del sistema estéreo WOW debido a los efectos de movimientos introducidos por el rastreador.

En base a esta clasificación, el diferencial mecánico se representará mediante un sistema de realidad virtual sin inmersión a través de ventanas (WOW, Windows on World). Con la ayuda de un ordenador personal se simulará el funcionamiento de un diferencial mecánico en un mundo virtual, respetando las leyes de la física y de la mecánica en la medida de lo posible, y obtener una respuesta coherente del mecanismo ante una situación real, como pueda ser el tránsito de un vehículo en una curva, en una recta o en situación de bloqueo.

2.3.3 Problemas e inconvenientes actuales de los entornos de Realidad Virtual

En términos del estado actual de la tecnología, existe aún un número de problemas por resolver para garantizar nuestra satisfacción como futuros usuarios a nivel sistemático y no casual. Estos problemas están siendo atacados en la actualidad por numerosos equipos humanos, a nivel técnico y científico. Entre ellos [5]:

- Representación.
- Demora en tiempo de respuesta.
- Angulo de visualización.
- Malestar por uso prolongado.

2.3.4 Futuro de la Realidad Virtual

El objetivo de la Realidad Virtual es poder crear, inventar, imponer reglas, simular situaciones, realidades, objetos, etc. en un mundo que sólo es real cuando se percibe mediante los sentidos del oído y la vista.

Dentro del campo de la educación, y de la ciencia en general, será una herramienta de gran valía y tal vez indispensable en los años por venir. Complementada con un par de lentes o cascos con audífonos integrados, así como un par de guantes especiales, se pueden realizar viajes al interior del cuerpo humano. En otras áreas como la historia, paleontología, química, física la posibilidad de aprovechamiento es enorme. Se podrían modelar hipótesis de ciudades ya desaparecidas como Pompeya, Atenas o Tenochtitlán. En áreas como la química se vería beneficiada, ya que los estudiantes serían capaces de abordar el interior mismo de la materia. En definitiva, se consigue una nueva herramienta de estudio y análisis para todos los campos de la ciencia y la cultura.

En lo que concierne a la resolución de dificultades a vencer en el campo de la Realidad Virtual y sus aplicaciones, todavía hay mucho trabajo por hacer. Para conseguir una Realidad Virtual Integral y tener contacto con los logros actuales, se requiere de una gran inversión en equipos y Software; esto sin tener en cuenta que muchos de los avances al respecto no están siendo comercializados y se restringe su uso a las compañías que los desarrollan o los clientes específicos que ordenan su elaboración [5].

Todo parece indicar que la evolución de sistemas comerciales futuros de Realidad Virtual, orientados al mercado de micros, se esforzará en permitir que la participación principal del usuario no especializado se centre en el desarrollo y aplicación de habilidades y de conocimientos orientados a la concepción, diseño y construcción de mundos virtuales [5].

Se debe reflexionar sobre la dirección que debería tomar el desarrollo de aplicaciones de Realidad Virtual ya que, así como puede ser utilizada como herramienta para el desarrollo de la educación, ciencia, medicina, etc. También puede ser utilizada para especialización en el ámbito militar, planificación de devastación y otros que, definitivamente, no creemos hayan sido la intención de los colaboradores y desarrolladores iniciales de la Realidad Virtual.

CAPÍTULO 3: EL DIFERENCIAL MECÁNICO

El diferencial mecánico, como el que se muestra en la figura 3.1, es un conjunto de transmisiones mecánicas formado por engranajes cuyo objetivo es permitir que las ruedas de un vehículo giren a revoluciones diferentes, según éste se encuentre tomando una curva hacia un lado o hacia el otro. Adicionalmente, se denominan diferenciales libres a aquellos que no disponen de sistemas de autobloqueo en caso de pérdida de tracción en una de las ruedas.



Fig. 3. 1: Diferencial mecánico libre

Como tal, es fundamental tratar en este capítulo el concepto de transmisión mecánica, en especial el engranaje como transmisión rígida, cómo funciona, su aplicación específica al diferencial mecánico, tipos de diferenciales y su proyección en el futuro.

3.1 Teoría de engranajes

Se habla de "par de engranajes, ruedas dentadas o engrane" para referirse al acoplamiento que se utiliza para transmitir potencia mecánica entre dos ejes mediante contacto directo entre dos cuerpos sólidos unidos rígidamente a cada uno de los ejes [6].

El diferencial está formado por engranajes, por tanto se expondrán sus partes, sus tipos, y qué peculiaridades tienen dentro del diferencial.

3.1.1 Tipos de Engranajes

Los engranajes se pueden clasificar atendiendo a la posición relativa de sus ejes [7].

3.1.1.1 Engranajes para ejes paralelos

Utilizan ruedas dentadas cilíndricas con dentado recto. En la figura 3.3 se muestra un engranaje de dentado externo que transmite movimiento opuesto, y en la figura 3.2, un engranaje de dentado interno que transmite movimiento del mismo sentido. Se utilizan en transmisiones de ejes paralelos formando así lo que se conoce con el nombre de trenes de engranajes. Este hecho hace que sean unos de los más utilizados, pues no en vano se pueden encontrar en cualquier tipo de máquina: relojes, juguetes, máquinas herramientas, etc. En cualquier sistema de engranajes, la velocidad del eje impulsado depende del número de dientes de cada engranaje.



Fig. 3. 2: Dentado recto interno



Fig. 3. 3: Dentado recto externo

Los engranajes helicoidales (Fig. 3.4) tienen la ventaja de transmitir más potencia que los rectos, y también más velocidad, además son más silenciosos y más duraderos. Por otro lado, se desgastan más que los rectos, son más caros de fabricar y necesitan generalmente más lubricante.



Fig. 3. 4: Helicoidal simple

El engranaje helicoidal simple genera una fuerza axial debido a la forma del dentado que debe ser soportada por un apoyo o cojinete. Para evitar este fenómeno se pueden utilizar el helicoidal doble (Fig. 3.5) o el Herringbone (Fig. 3.6), que con su doble dentado contrarresta dicha fuerza axial.



Fig. 3. 5: Helicoidal doble



Fig. 3. 6: Herringbone

3.1.1.2 Engranajes para ejes que se cortan en el espacio

Se trata de ruedas dentadas en forma de troncos de cono, cuya función consiste en transmitir movimiento entre ejes que se cortan en el espacio. Dentro de este tipo de engranajes se pueden subdividir según el dentado en cónico recto y cónico helicoidal.

El engranaje de dentado del cónico recto (Fig. 3.7) son utilizados para efectuar transmisiones con ejes en 90° , como inconveniente generan más ruido que los engranajes cónicos helicoidales. En la actualidad se usan escasamente.



Fig. 3. 7: Cónicos rectos

El engranaje de dentado cónico helicoidal, representado en la figura 3.8, al igual que en los engranajes rectos, permite una transmisión más silenciosa y progresiva en transmisiones con ejes cruzados.



Fig. 3. 8: Cónico helicoidal

En el diferencial mecánico libre existen dos transmisiones entre engranajes cónicos rectos situado dentro del casquete del diferencial, en este mecanismo se identifican como satélites y planetarios, y son estos los que se ocupan de equilibrar los palieres el movimiento de un automóvil en curva.

3.1.1.3 Engranajes para ejes que se cruzan en el espacio: hiperbólicos

El engranaje helicoidal cruzado como el que se muestra en a figura 3.9 se utiliza en transmisiones donde los ejes se cruzan en el espacio. Los dientes de estos engranajes no son paralelos al eje de la rueda dentada, sino que se enroscan en torno al eje en forma de hélice. Estos engranajes son apropiados para grandes cargas porque los dientes engranan formando un ángulo agudo, en lugar de 90° como en un engranaje recto. Presentan un comportamiento más silencioso que el de los dientes rectos. Poseen una mayor relación de contacto debido al efecto de traslape de los dientes y pueden transmitir mayores cargas a mayores velocidades. La principal desventaja de utilizar este tipo de engranaje es la fuerza axial que este produce para contrarrestar esta reacción se tiene que colocar cojinetes de soporte.



Fig. 3. 9: Helicoidal cruzado.

El engranaje Hipoidal, mostrado en la figura 3.10, es el que se utiliza en la corona del diferencial mecánico libre. Durante cierto tiempo los engranajes espirales se utilizaron exclusivamente en las transmisiones de los ejes traseros de los automóviles.

En 1925 Gleason introdujo el engranaje hiperbólico que lo reemplazó en esta aplicación. Estos son de apariencia semejante a los cónicos espirales excepto que el eje del piñón está desplazado con respecto a la corona de manera que los ejes no se interceptan. Para poder tener este descentramiento a la vez que se mantiene el contacto de la línea, la superficie de contacto de un engranaje hipoidal se aproxima a un hiperboloide de revolución en vez de a un cono como en los engranajes cónicos.

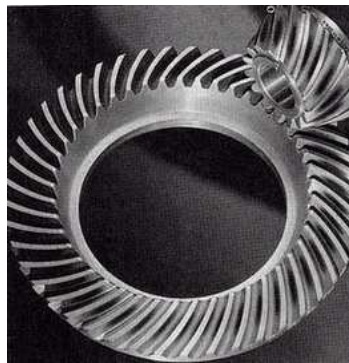


Fig. 3. 10: Corona Hipoidal

El descentramiento es una ventaja en las aplicaciones automotrices debido a que permite disminuir la flecha de la transmisión Cardan, lo que a su vez permite bajar la carrocería. Adicionalmente, los piñones hipoidales son más fuertes que los piñones cónicos espirales, debido a que se pueden diseñar de tal manera que el ángulo espiral del piñón sea mayor que el de la corona, lo que a su vez produce un mayor diámetro del piñón y por tanto más resistente que en el piñón cónico correspondiente.

Otra diferencia es que los engranajes hipoidales tienen efecto deslizante a lo largo de los dientes en tanto que los engranajes cónicos espirales no la tienen. Los engranajes hipoidales actúan más silenciosamente y se pueden emplear en relaciones mayores de velocidad que los engranajes cónicos espirales; aparte de que los engranajes hipoidales pueden ser esmerilados perfectamente.

3.1.2 Partes de los Engranajes

En la figura 3.11 se muestra las partes fundamentales de un engranaje cilíndrico recto [7]:

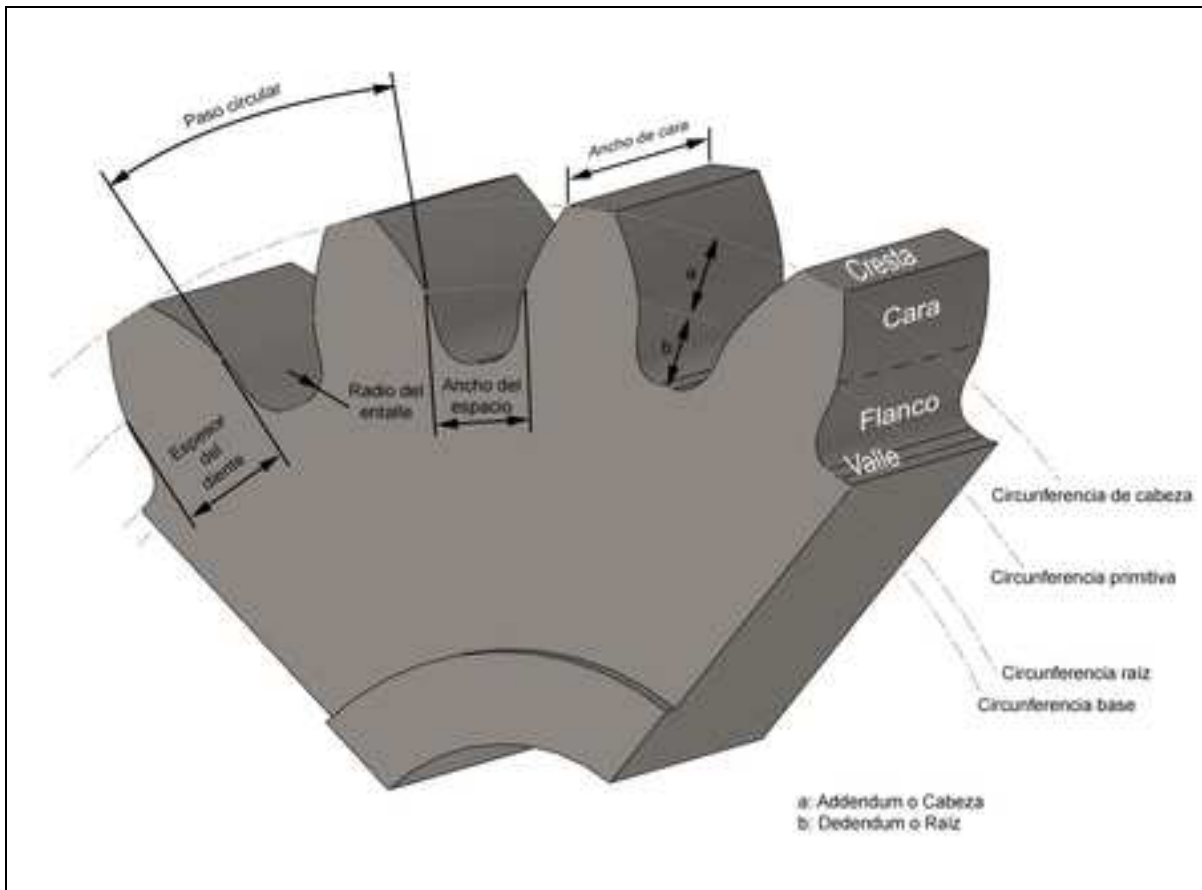


Fig. 3. 11: Partes de una rueda dentada

- *Diente de un engranaje* (Fig. 3.11): es la parte saliente de la rueda dentada y transmite el esfuerzo de empuje y la potencia desde los ejes motrices a los ejes conducidos. El perfil del diente está constituido por dos curvas evolventes de círculo, simétricas respecto al eje que pasa por el centro del mismo.
- *Paso circular* (Fig. 3.11): es la longitud de la circunferencia primitiva correspondiente a un diente y un vano consecutivos.
- *Espesor del diente* (Fig. 3.11): es el grosor del diente en la zona del diámetro primitivo.

- *Número de dientes*: es el número de dientes que tiene el engranaje. Se simboliza como (Z). Es fundamental para calcular la relación de transmisión. El número de dientes de un engranaje no debe estar por debajo de 18 dientes cuando el ángulo de presión es 20° ni por debajo de 12 dientes cuando el ángulo de presión es de 25° .
- *Diámetro exterior*: es el diámetro de la circunferencia que limita la parte exterior del engranaje.
- *Diámetro interior*: es el diámetro de la circunferencia que limita el pie del diente.
- *Pie del diente o dedendum* (Fig. 3.11): Es la parte del diente comprendida entre la circunferencia interior y la circunferencia primitiva.
- *Cabeza del diente o adendum* (Fig. 3.11): Es la parte del diente comprendida entre el diámetro exterior y el diámetro primitivo.
- *Flanco* (Fig. 3.11): Es la cara interior del diente, es su zona de rozamiento.
- *Altura del diente* (Fig. 3.11): es la suma de la altura de la cabeza (*adendum*) más la altura del pie (*dedendum*).
- *Módulo*: el módulo de un engranaje es una característica de magnitud que se define como la relación entre la medida del diámetro primitivo expresado en milímetros y el número de dientes. En los países anglosajones se emplea otra característica llamada *Diametral Pitch*, que es inversamente proporcional al módulo. El valor del módulo se fija mediante cálculo de resistencia de materiales en virtud de la potencia a transmitir y en función de la relación de transmisión que se establezca. El tamaño de los dientes está normalizado. El módulo está indicado por números. Dos engranajes que engranen tienen que tener el mismo módulo.
- *Circunferencia primitiva*: Se muestra en las figuras 3.11 y 3.12, y es la circunferencia a lo largo de la cual engranan los dientes. Con relación a la circunferencia primitiva se determinan todas las características que definen los diferentes elementos de los dientes de los engranajes.

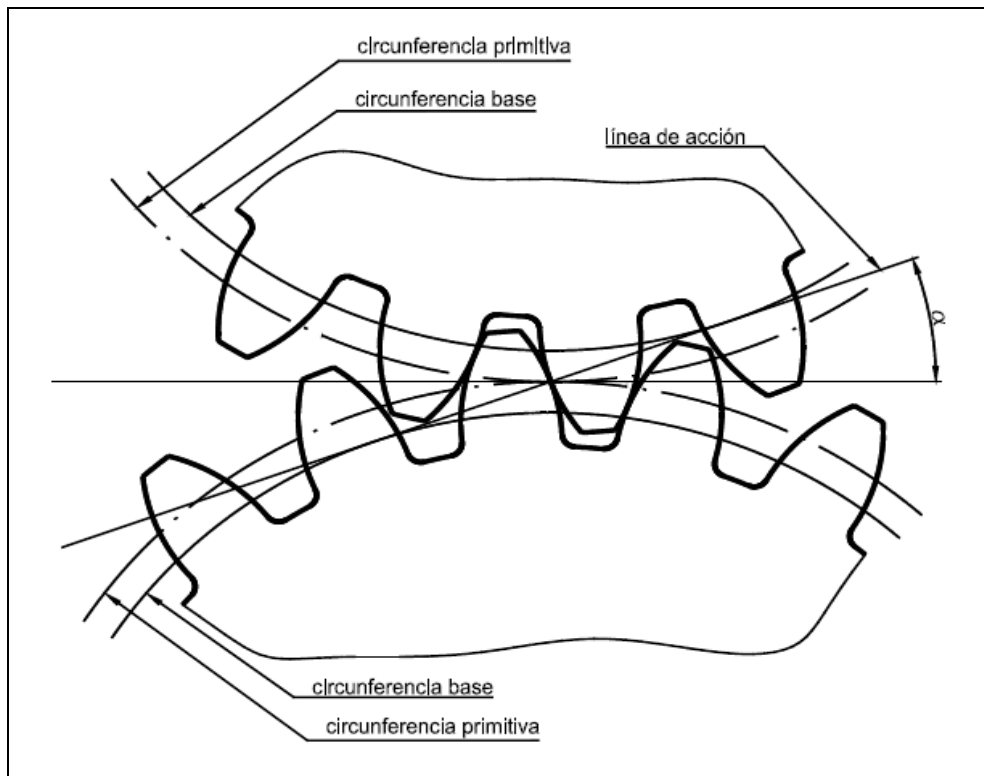


Fig. 3.12: Esquema de engrane

- *Circunferencia base* (Fig. 3.12): es el lugar geométrico de todos los puntos de origen de las evolventes que forman los flancos de los dientes de una rueda dentada.
- *Diámetro base* (Fig. 3.12): es el diámetro de la circunferencia base.
- *Línea de acción* es la tangente común a las dos circunferencias base del engranaje, y a su vez representa el lugar geométrico de los sucesivos puntos de contacto de los flancos conjugados (Fig. 3.12).
- *Ángulo de presión* (Fig. 3.12): el que forma la línea de acción con la tangente a la circunferencia de paso, φ (20° ó 25° son los ángulos normalizados).
- *Largo del diente*: es la longitud que tiene el diente del engranaje
- *Distancia entre centro de dos engranajes*: es la distancia que hay entre los centros de las circunferencias de los engranajes.
- *Relación de transmisión*: es la relación de giro que existe entre el piñón conductor y la rueda conducida.

3.2 El mecanismo diferencial

El diferencial fue inventado en 1827 por el mecánico francés Onesiphore Pecqueur (nacido en Pasde- Calais en 1792 y fallecido en París en 1852), maestro de taller del Conservatorio de Artes y Oficios de París. El diferencial llegó a ser de aplicación bastante común en los primeros vehículos de vapor de la segunda mitad del siglo XIX: uno de los primeros ejemplos fue el del Clayton & Shuttle-worth de 1865 [8].

Está demostrado que las ruedas de un automóvil, ante una trayectoria curva, realizan diferentes trazados, lo cual implica que la rueda interior hace un recorrido menor que la rueda exterior, provocando (si estuvieran unidas directamente a la corona del par cónico) el arrastre o patinado de una de las ruedas. Debido a esto, es necesario montar un mecanismo que permita el giro de las dos ruedas motrices a distintas velocidades, al mismo tiempo que transmite a las mismas el esfuerzo motriz. Esto se consigue con la implantación de un mecanismo diferencial, que en las curvas permite dar un mayor número de vueltas a la rueda exterior y disminuye las de la interior.



Fig. 3. 13: Sección de una transmisión completa de automóvil

El diferencial (Fig. 3.13) siempre va situado en el eje motriz del automóvil, si este es de tracción total, irá un diferencial en cada eje y puede existir un tercer diferencial central para compensar el eje trasero con el delantero.

A continuación se describe de forma básica los componentes del diferencial para posteriormente poder analizar la evolución del mecanismo y las últimas tendencias.

3.2.1 Descripción del diferencial mecánico libre

En la figura 3.14 se puede observar una vista explosionada de un diferencial mecánico libre con sus componentes más importantes numerados.

El diferencial está constituido por la corona (detalle 2 de la Fig. 3.14), que se une a la caja del diferencial por mediación de tornillos (detalle 1 de la Fig. 3.14), y en su interior se aloja el mecanismo diferencial, formado por los satélites (detalle 7 de la Fig. 3.14) y los planetarios (detalles 4 y 7 de la Fig. 3.14). Los satélites se montan sobre el eje (detalle 6 de la Fig. 3.14) que va alojado en la carcasa o casquete (detalle 3 de la Fig. 3.14), de manera que puedan girar libremente en él; pero son volteados por la caja (detalle 3 de la Fig. 3.14) cuando gira la corona (detalle 2 de la Fig. 3.14). Engranados con los satélites se montan los planetarios, cuyos ejes de giro se alojan en la corona y caja del diferencial respectivamente, pudiendo girar libremente en ellos con interposición de casquillos de fricción. A los ejes de los planetarios se unen a su vez los palieres, que transmitirán el movimiento a las ruedas [9].

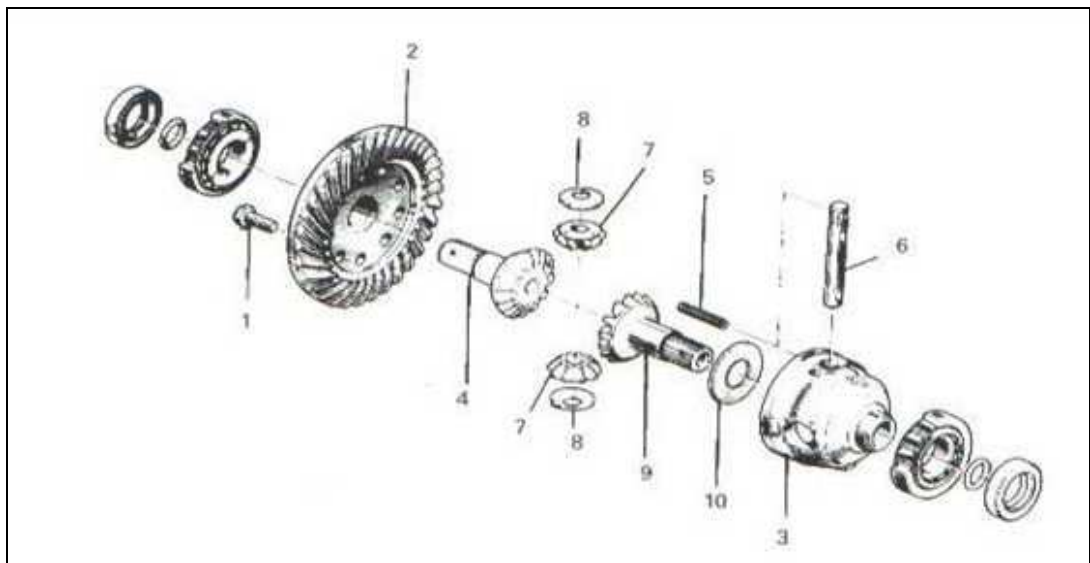


Fig. 3. 14: Despiece explosionado de un diferencial mecánico libre

La corona empieza a girar impulsada por el piñón de ataque (Fig. 3.15), arrastra con ella a la caja del diferencial (B), que en su giro voltea a los satélites (C) y (D) que, actuado como cuñas, arrastran a su vez a los planetarios (E) y (F), los cuales transmiten el movimiento a las ruedas haciéndolas girar en el mismo sentido y con igual velocidad mientras el vehículo marche en línea recta; pero cuando toma una curva, la rueda interior ofrece más resistencia al giro que la exterior (al tener que recorrer distancias desiguales) y, por ello, los satélites (C) y (D) rodarán un poco sobre uno de los planetarios (el correspondiente a la rueda interior) multiplicando el giro en el otro (el de la rueda exterior). De esta manera, lo que pierde en giro una rueda lo gana la otra, ajustándose automáticamente el giro de cada una de ellas al recorrido que le corresponda efectuar en cada curva. Igualmente, las diferencias de trayectoria en línea recta, debidas a diferencias de la presión de inflado de los neumáticos, irregularidades del terreno, etc., son absorbidas por el diferencial [9].

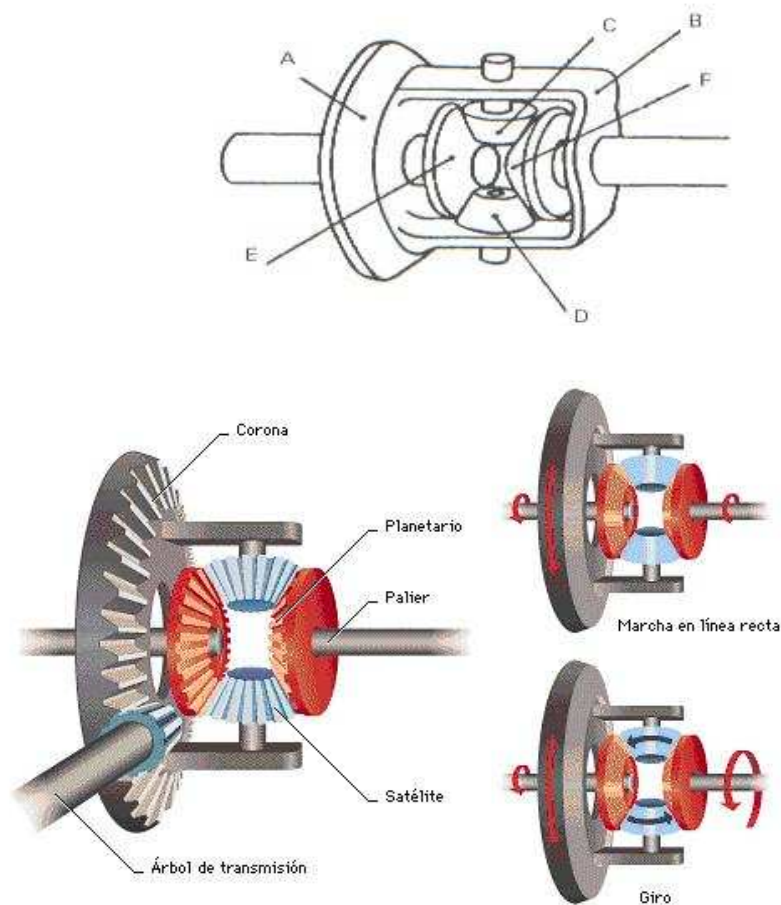


Fig. 3. 15: Partes de un diferencial

El diferencial libre tiene como inconveniente que cuando una de las dos ruedas del eje motriz rueda momentáneamente sobre una superficie deslizante (hielo, barro, etc.), o bien se levanta en el aire (a consecuencia de un bache o durante el trazado de una curva a alta velocidad) la característica de balanza del diferencial da a lugar que el par motor se concentre en la rueda cuya adherencia se ha reducido (Fig. 3.16). Esta rueda tiende a embalsarse, absorbiendo todo el par, mientras que la opuesta permanece inmóvil, lo que se traduce en pérdida de tracción del coche [9].

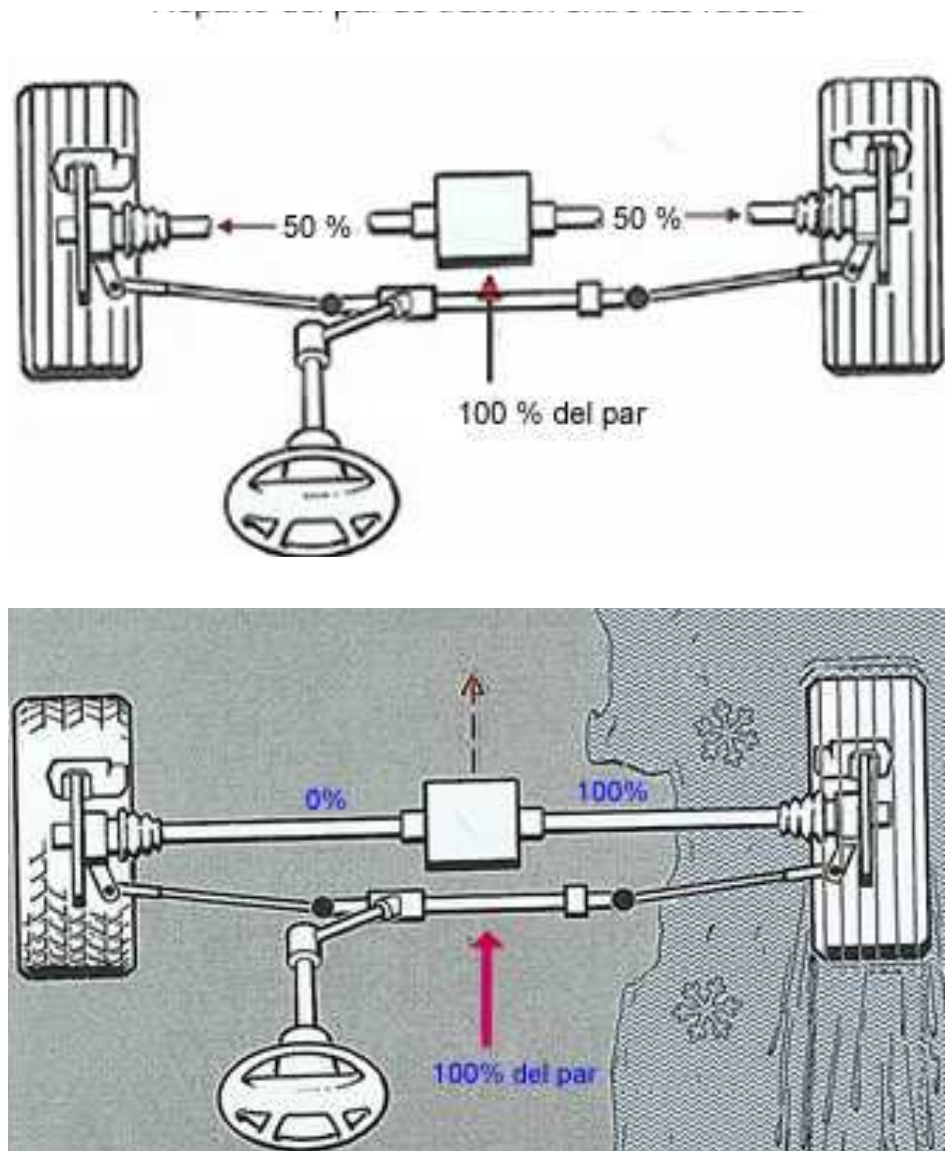


Fig. 3. 16: Repartos de par

3.2.2 Diferenciales 100% bloqueables

Los diferenciales 100% bloqueables son aquellos que funcionan, de manera permanente o transitoria, como un eje solidario. Por lo tanto, ambas ruedas giran a la misma velocidad, la fuerza total tractora capaz de transmitir al suelo es aproximadamente igual a la suma de las adherencias disponibles en cada rueda, y el par que se aplica a cada una de ellas viene determinado por una serie de factores relacionados entre sí como el radio de giro, características del vehículo, el radio de circunferencia de las ruedas, la carga sobre el neumático o la fuerza lateral y longitudinal a la que está siendo sometido en ese momento [10].

Este tipo de diferencial es muy útil cuando las condiciones de adherencia son muy adversas y únicamente se puede usar cuando el vehículo transita a baja velocidad. Actualmente no se usa en turismos, y sólo se monta en algunos vehículos para todo-terreno, en los cuales se activa y se desactiva según las características del pavimento.

3.2.3 Diferenciales autoblocantes

En este apartado se engloban los dispositivos que buscan un compromiso entre el diferencial libre y el diferencial totalmente bloqueado en el que sólo se anula una parte del efecto diferencial, es decir, limitan la posibilidad de que una rueda gire libre respecto a la otra según un tarado fijo predeterminado. Ese tarado se expresa como una relación entre las dos ruedas en tanto por ciento, de forma que el cero corresponde a un diferencial libre, y el 100 a ruedas que giran solidarias, es decir, con el diferencial completamente bloqueado (como un eje rígido) [10].

3.2.3.1 Diferencial autoblocante por discos de fricción

Este tipo de diferencial autoblocante utiliza un conjunto de discos similares a los de un embrague que, en función de la precarga del mismo, bloquean los dos palieres bajo determinadas condiciones de funcionamiento.

El efecto de bloqueo se produce porque el par que transmite el piñón de ataque al diferencial no va directamente al eje porta-satélites (detalle 9, Fig. 3.17) y satélites (detalle 8, Fig. 3.17), como en un diferencial libre, sino a través de dos discos de empuje (detalle 6, Fig. 3.17) [9].

En este caso, la regulación de la carga se consigue mediante dos métodos. El primero es aplicando una precarga a los juegos de discos mediante un muelle helicoidal o una arandela Belleville para proporcionar un valor mínimo de par que haga que se rompa la barrera estática que mantiene los ejes solidarios. La segunda, se consigue mediante un tallado de los dientes de los satélites especial, diseñado cuidadosamente para cargar los juegos de discos a medida que el par aumenta.

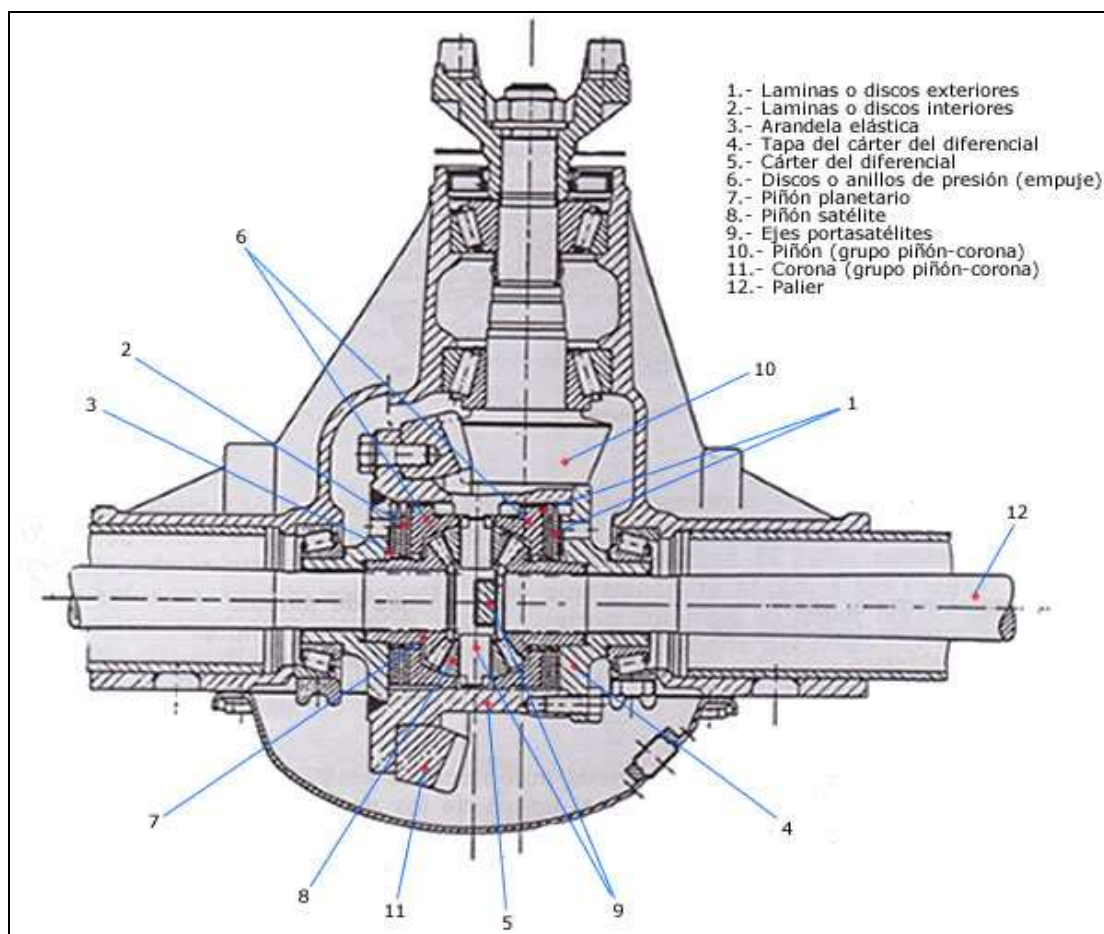


Fig. 3. 17: Estructura interna de un diferencial autoblocante por discos de fricción

3.2.3.2 Salisbury axle o diferencial de rampas

Otra variante de diferencial autoblocante es el llamado *Salisbury axle* o más conocido como diferencial de rampas (Fig. 3.18). Este diferencial también emplea un juego de discos para bloquear progresivamente los dos ejes de salida del diferencial juntos. En esta variante de diferencial de discos, el eje de los satélites descansa sobre unas rampas en “V” que son empujadas por este eje en función del par que entra al diferencial. Con este sistema el reparto de par puede variar en función de si estamos en fase de aceleración o de retención.

Los piñones satélites planetarios son análogos a los de un diferencial convencional. Cada piñón planetario se acopla sobre sendos bujes estriados, que a su vez encajan sobre cada una de las dos mitades de la caja diferencial. Entre cada mitad de la caja y el buje estriado correspondiente existe un embrague compuesto por discos de fricción y arandelas elásticas de acero, o bien pequeños muelles helicoidales alojados en las carcasas [9].

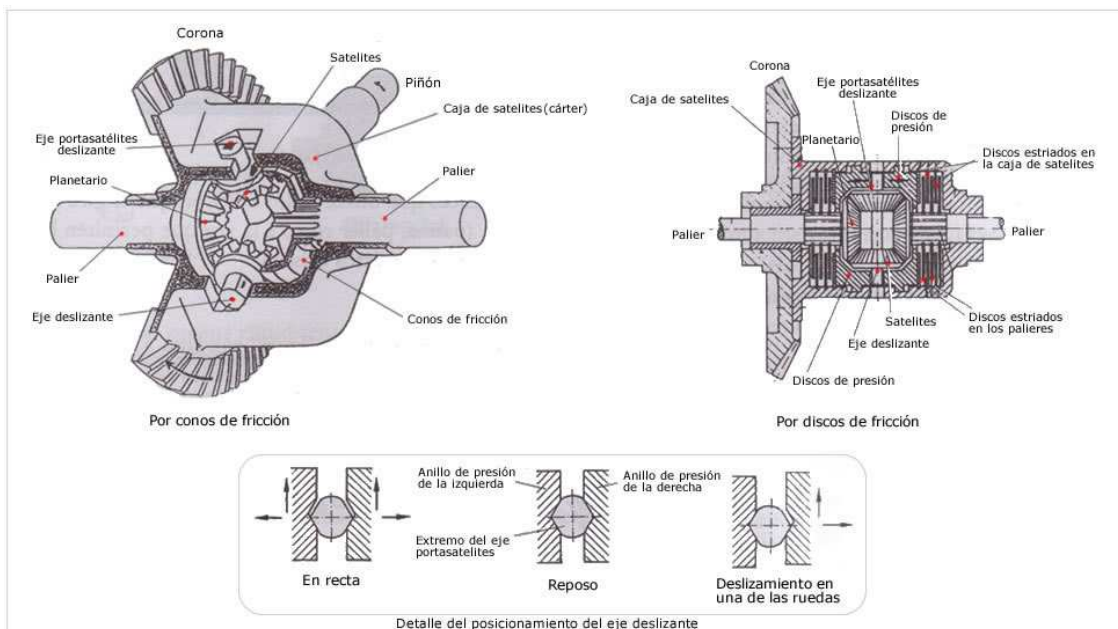


Fig. 3. 18: Diferencial de rampas

Los embragues de cada planetario están calculados para permitir un cierto deslizamiento mientras no se produzca la total pérdida de adherencia de una de las dos ruedas. Así, cuando el coche da una curva, este pequeño deslizamiento permite que la rueda exterior gire algo más rápido que la interior, comportándose el dispositivo como un diferencial convencional [9].

Cuando las dos ruedas disponen de similar adherencia, los ejes deslizantes de los satélites están sometidos a un esfuerzo que tiende a hacerlos subir por las rampas en “V”, pero sin embargo, como ambos se cruzan uno por delante del otro, el efecto de cada uno se contrapone, permaneciendo ambos equilibrados en el fondo de la “V” (ver detalle de posicionamiento del eje deslizante de la Fig. 3.18). En el momento en que una de las dos ruedas pierde adherencia, los satélites tienden a girar entre los planetarios y la tensión a que estaban sometidos los ejes de los primeros disminuye. La posición de equilibrio de los ejes de satélites se rompe y entonces el eje del lado de la rueda que todavía tiene adherencia sube por las rampas en “V”, ejerciendo un empuje sobre el piñón planetario que se aplica ahora con fuerza sobre su cubo estriado. Este movimiento aprieta el embrague de placas de este lado y el planetario se hace solidario de la caja diferencial, anulándose en parte, por tanto, el efecto diferencial [9].

3.2.3.3 Diferencial viscoso o Ferguson

Este diferencial autoblocante suele utilizarse como diferencial central en vehículos con tracción a las 4 ruedas (Fig. 3.19). Está constituido por una carcasa solidaria al árbol de transmisión que encierra unos discos, de los cuales, unos están unidos a la carcasa y otros al portadiscos solidario al eje de salida, los discos de ambas series van intercalados y con hendiduras y taladros, a través de los cuales puede pasar el aceite silicona mezclado con un 20% de aire, que llena todo el conjunto [9].

Una parte del conjunto es solidaria a las ruedas de un eje y la otra a las ruedas de otro eje. En marcha recta las ruedas traseras se ven arrastradas por las del tren que recibe la tracción a través de su contacto con el suelo, generándose una pequeña aportación de par a través del aceite silicona. Cuando uno de los ejes pierde tracción el deslizamiento que se genera entre los discos alternos hace aumentar la temperatura y presión en el

aceite silicona que los envuelve, aumentando las fuerzas de cizalladura, arrastrando los discos conductores a los conducidos, consiguiéndose un giro solidario entre ambos. El momento de actuación lo determina el número de discos, los taladros y el aire que tengan mezclado, no recibiendo en funcionamiento normal nada más que una pequeña parte del par a través de él, apenas un 10% [9].

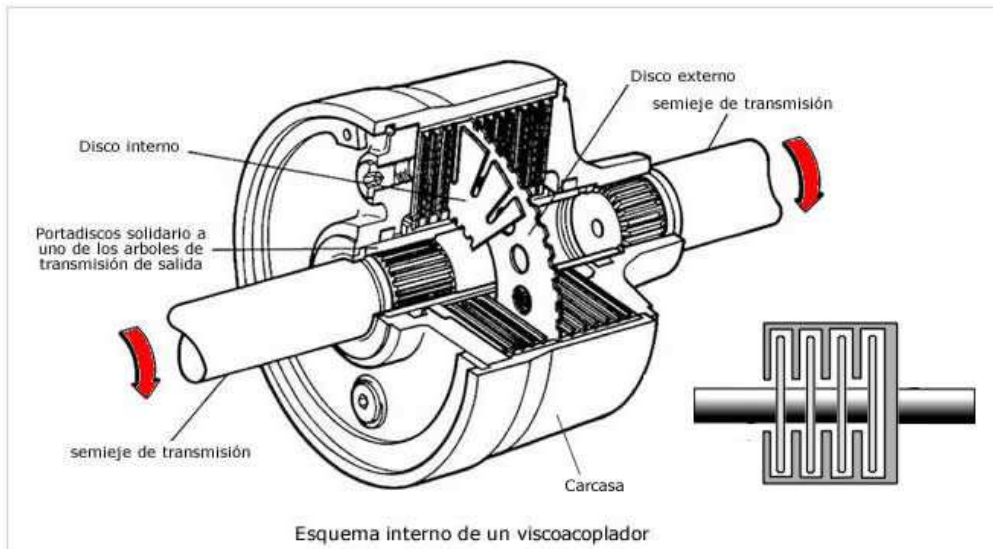


Fig. 3. 19: Esquema interno de un viscoacoplador

3.2.3.4 Diferencial Torsen

Su nombre procede de las palabras inglesas *Torque Sensitive*, que en español quieren decir “sensible al par”. Fue inventado por Vernon Gleasman y fabricado por Gleason Corporation [9].

Es un tipo de diferencial cuya peculiaridad radica en que reparte la fuerza que procede del motor a las ruedas de forma independiente a la velocidad rotatoria de cada uno de los dos árboles o semiejes de transmisión que parten de él. En cualquier diferencial autoblocante, ya sea convencional o viscoso, el reparto de fuerza entre los dos semiejes se realiza siempre de forma proporcional a su velocidad de giro, sin embargo, el diferencial Torsen (Fig. 3.20) puede repartir la fuerza del motor a cada semieje en función de la resistencia que oponga cada rueda al giro, pero al mismo

tiempo permite que la rueda interior en una curva gire menos que la exterior, aunque esta última reciba menos par [9].

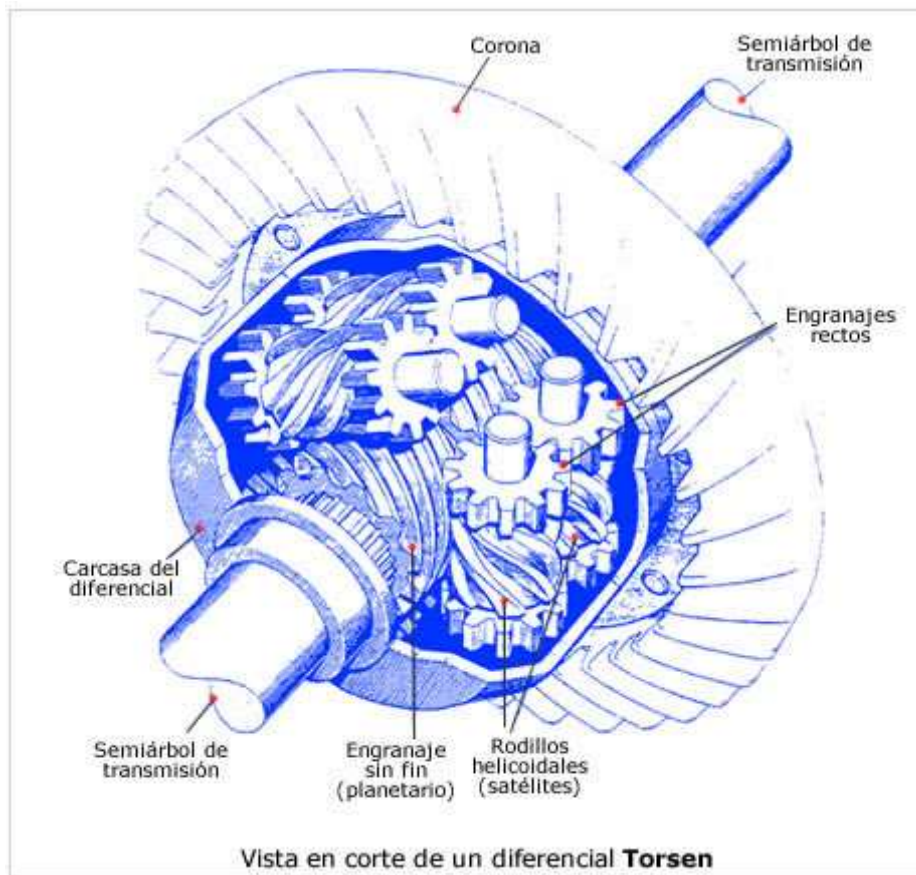


Fig. 3. 20: Vista en corte de un diferencial Torsen

El diferencial Torsen basa su funcionamiento en la combinación de una serie de engranajes convencionales y helicoidales. En concreto, se utilizan tres pares de ruedas helicoidales que engranan a través de dientes rectos situados en sus extremos. Si se compara con un diferencial convencional, en un Torsen se sustituyen los satélites convencionales por tres pares de engranajes helicoidales, engranados dos a dos por piñones de dientes rectos en sus extremos. Los planetarios en este caso son tornillos sin fin, con los cuales engrana cada uno de los engranajes helicoidales.

En curva los satélites giran sobre sus ejes acelerándose uno y frenándose otro para permitir la diferente velocidad de cada rueda. Si se genera el deslizamiento de una

rueda los satélites helicoidales no pueden hacer girar más rápido al planetario, dada la disposición de tornillo sin fin. Como los satélites forman parejas, la reacción de uno frente al otro impide el giro del planetario cuando hay deslizamiento (Fig. 3.21).

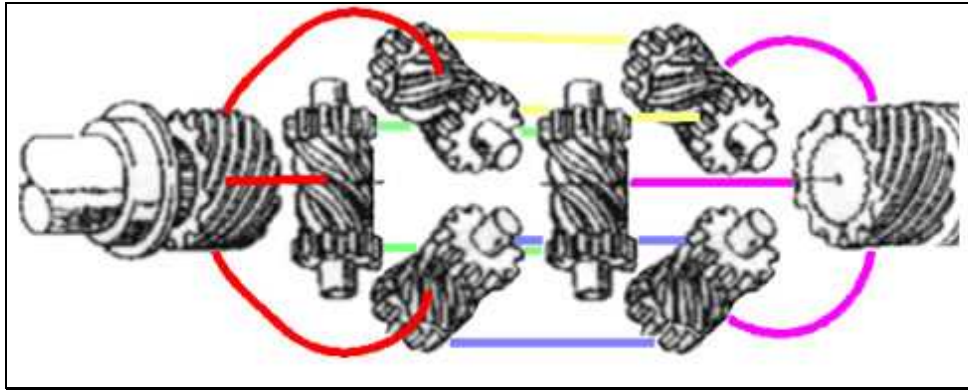


Fig. 3. 21: Engranajes internos del Torsen

El tarado a partir de cual manda el par a la rueda que tiene mejor agarre se determina con el ángulo de la hélice helicoidal. Esto permite, disponer siempre del máximo par en la rueda que mas agarre tiene, sin tener que llegar al deslizamiento en la rueda de menor agarre, y que este propicie el bloqueo del diferencial. Además, no sólo actúan en aceleración sino que también lo hacen en retención, es decir, el diferencial aporta más par de frenado (de retención) a la rueda izquierda o derecha dependiendo de cuál tenga más agarre y cuando el reparto de par a uno u otro lado llega a un cierto límite el diferencial se bloquea por completo. Dicho porcentaje de bloqueo (lo que habitualmente se denomina "tarado" del diferencial) suele ser distinto para aceleración y para retención [9].

En conclusión se pueden destacar como ventajas sus altos niveles de confort, porque el efecto autoblocante del diferencial Torsen se genera de forma netamente mecánica, sin escalonamientos y de manera imperceptible. No puede haber manejos equivocados, porque trabaja de forma automática y además funciona ampliamente exento de desgaste.

CAPÍTULO 4: MODELADO DEL DIFERENCIAL CON SOLID EDGE

En este capítulo se aborda uno de los dos núcleos más importantes de este proyecto, el modelado de cada pieza y la generación del conjunto de un diferencial mecánico real del Departamento de Ingeniería Mecánica de la Universidad Carlos III de Madrid con el programa de diseño Solid Edge V.18. Su importancia se debe a que es capaz de asistir en el diseño, facilitar la creación de planos y generar simulaciones de piezas y conjuntos mecánicos en entornos virtuales sin necesidad de fabricar prototipos reales. Por ello es necesario describir a grandes rasgos dicho programa y encuadrarlo dentro de los sistemas de asistencia por ordenador aplicados hoy en día a la industria.

4.1. Sistemas CAD-CAE-CAM-CAPP

El término Diseño asistido por ordenador fue acuñado por Douglas Ross y Dwight Baumann en 1959, y aparece por primera vez en 1960, en un anteproyecto del MIT, titulado “Computer-Aided Design Project” [11]. En aquella época ya se había comenzado a trabajar en la utilización de sistemas informáticos en el diseño, fundamentalmente de curvas y superficies.

4.1.1. CAD (Computer Aided Design)

CAD significa Diseño Asistido por Computador (del inglés Computer Aided Design) y representa el conjunto de aplicaciones informáticas que permiten a un diseñador “definir” el producto a fabricar [11]. Permite al diseñador crear imágenes de partes, circuitos integrados, ensamblajes y modelos de prácticamente todo lo que se le ocurra en una estación gráfica conectada a un computador. Estas imágenes se transforman en la base de un nuevo diseño, o en la modificación de uno previamente existente. A éstas se le asignan propiedades geométricas, cinéticas, químicas y físicas, entre otras, mejorando el diseño sobre papel. De esta manera, se consigue una mayor velocidad en el diseño, al existir la posibilidad de corregir, encargándose el computador de recalcular el dibujo. Existen sistemas CAD especiales para aplicaciones mecánicas, electrónicas y de arquitectura, los cuales permiten una mejor interrelación con sus respectivos sistemas CAE.

A continuación se enumeran algunas de las características de las aplicaciones CAD que permiten especificar y formalizar la representación inequívoca de una pieza o sistema:

- CAD 2D: sustitutivo básicamente del tablero de dibujo, la representación de los objetos es bidimensional; la información geométrica de que dispone el ordenador es bidimensional, es decir, está contenida en un plano. Las vistas son generadas de forma independiente y no existe capacidad de asociación entre las mismas. Su ámbito de aplicación es muy amplio.
- Modelado geométrico 3D: descripción analítica de la volumetría, contorno y dimensiones del objeto o sistema, incluyendo relaciones geométricas e incluso algebraicas entre los distintos componentes. Solid Edge forma parte de este tipo de modelado, que además permite obtener automáticamente diseños en 2-D a partir de un modelado en 3-D.

El uso de la tecnología CAD supone para el diseñador un cambio en el medio de plasmar los diseños industriales. Esta revolución en el campo del diseño ha venido de la mano de la revolución informática. Las mejoras que se alcanzan son:

- Mejora en la representación gráfica del objeto diseñado: con el CAD el modelo puede aparecer en la pantalla como una imagen realista, en movimiento, y observable desde distintos puntos de vista.
- Mejora en el proceso de diseño: se pueden visualizar detalles del modelo, comprobar colisiones entre piezas, interrogar sobre distancias, pesos, inercias, etc. En conclusión, se optimiza el proceso de creación de un nuevo producto reduciendo costes, ganando calidad y disminuyendo el tiempo de diseño.

En resumen, se consigue una mayor productividad en el trazado de planos, integración con otras etapas del diseño, mayor flexibilidad, mayor facilidad de modificación del diseño, ayuda a la estandarización, disminución de revisiones y mayor control del proceso de diseño.

4.1.2. CAE (Computer Aided Engineering)

Una herramienta CAE se traduce como Ingeniería Asistida por Computador y es un conjunto de programas informáticos que permiten analizar y simular el comportamiento físico de un producto diseñado previamente mediante CAD.

Hoy en día, CAE se aplica a dos ramas separadas: una es la aplicada a la mecánica y otra a la electrónica. Ambas realizan extensos análisis respecto de las leyes físicas, así como de los estándares de la industria. El CAE mecánico, en particular, incluye un análisis por elementos finitos (F.E.A., *finite element analysis*) para evaluar las características estructurales de una parte y programas avanzados de cinemática para estudiar los complejos movimientos de algunos mecanismos. El CAE electrónico, asimismo, permite verificar los diseños antes de fabricarlos, simular su uso y otros análisis técnicos para ahorrar costes [11].

Seguidamente se mencionan algunas de las principales aplicaciones CAE:

- Cálculo de propiedades físicas: volumen, masa, centro de gravedad, momentos de inercia, etc.
- Análisis tensional y cálculo mecánico y estructural: lineal y no lineal.

- Análisis de vibraciones.
- Simulación del proceso de inyección de un molde: análisis dinámico y térmico del fluido inyectado.
- Simulación de procesos de fabricación: mecanizado, conformado de chapas metálicas, soldaduras, análisis de fijaciones.
- Simulación gráfica del funcionamiento del sistema: cálculo de interferencias, estudios aerodinámicos, acústicos, ergonómicos, etc.

La reducción del tiempo y del costo, el descubrimiento de errores en el diseño, y la mejora de la calidad y cualidades del producto son objetivos muy importantes en el desarrollo del mismo dentro de un ambiente de competitividad industrial. Esto permitirá seleccionar el material más adecuado para la pieza, así como efectuar las modificaciones necesarias para mejorar el rendimiento de la misma [11].

Antes del desarrollo del CAE un cambio de material suponía la construcción de un nuevo prototipo, en lo cual se empleaban varios días; con el CAE supone alterar una serie de parámetros, operación que dura escasos segundos. Aunque esta técnica no elimina por completo la necesidad de construir prototipos, sí reduce drásticamente el número de pruebas a realizar con dichos prototipos y constituye una ayuda para poder identificar en una fase temprana la fiabilidad, el rendimiento, determinados problemas de coste, etc.

La Ingeniería Asistida por Ordenador también es conocida como Elaboración Virtual de Prototipos, debido a que permite simular el comportamiento de la pieza de forma virtual.

4.1.3. CAM (Computer Aided Manufacturing)

El sistema que agrupa las aplicaciones encargadas de traducir las especificaciones de diseño a especificaciones de producción se denomina CAM o Fabricación Asistida por Computador. Entre las posibilidades de aplicación se encuentran [11]:

- Generación de programas de Control Numérico.
- Simulación de estrategias y trayectorias de herramientas para mecanizado del producto diseñado (partiendo de un modelo CAD).
- Programación de soldaduras y ensamblajes robotizados.
- Inspección asistida por computadora. (CAI – *Computer Aided Inspection*).
- Ensayo asistido por computadora. (CAT – *Computer Aided Testing*).

La utilidad del CAM es proporcionar una serie de herramientas que permitan fabricar la pieza diseñada. El sistema CAM también se encarga de simular el recorrido físico de cada herramienta, con el fin de prevenir posibles interferencias entre herramientas y materiales.

4.1.4. CAPP (Computer Aided Process Planning)

Un sistema CAPP (Planificación de Procesos de Fabricación Asistidos por Computador) es un conjunto de programas que capturan las capacidades de un ambiente manufacturero específico y principios manufactureros ingenieriles, con el fin de crear un plan para la manufactura física de una pieza previamente diseñada.

Este plan especifica la maquinaria que se ocupará en la producción de la pieza, la secuencia de operaciones a realizar, las herramientas, velocidades de corte y avances, y cualquier otro dato necesario para llevar la pieza del diseño al producto terminado. Para usar el CAPP más efectivamente, el diseño debería provenir electrónicamente de un ambiente CAD.

El CAPP tiene dos tipos básicos: el variante y el generativo [12]. El variante es el más comúnmente usado y desarrolla un plan modificando un plan previamente existente, eligiendo éste usando criterios de tecnología de grupos y de clasificación. El generativo incorpora el concepto de inteligencia artificial, usando sus conocimientos sobre las capacidades de la planta. Basado en la descripción de la pieza (geometría y material) y sus especificaciones, el computador elige el método óptimo para producir la pieza y genera automáticamente el plan.

4.2. Descripción del programa Solid Edge V.18

Solid Edge V.18 es un software de diseño asistido por ordenador para empresas de diseño industrial, ingenierías y fabricantes. Este programa de diseño mecánico permite la representación de piezas y conjuntos en 3D, así como el modelado en chapa, representación de soldaduras y generación de planos en 2D.

Dependiendo del tipo de trabajo que se necesite realizar, se debe elegir el entorno de trabajo adecuado (Fig. 4.1). El entorno “pieza sólida” permite diseñar la mayoría de las piezas. El entorno “pieza de chapa” se especializa en elementos sólidos con poco espesor. El entorno conjunto permite ensamblar piezas sólidas para formar grupos de elementos, y el entorno plano ayuda a representar en un plano vistas, cotas, etc., de piezas modeladas en tres dimensiones.



Fig. 4. 1: Entornos de trabajo

Para este proyecto los entornos que se utilizan principalmente son el de pieza sólida y el de conjunto.

4.2.1 El interfaz Solid Edge V.18 en entorno “pieza sólida”

Si se ejecuta el entorno de pieza sólida, el programa muestra un menú (Fig. 4.2.) con diferentes herramientas de modelado [13]:

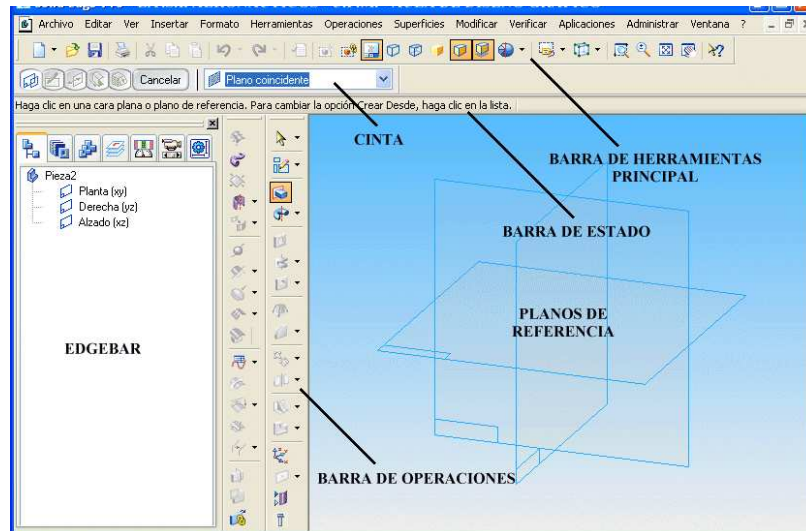


Fig. 4. 2: Interfaz entorno pieza sólida

- **Barra de herramientas principal.** Es la barra de herramientas común en todo programa que trabaje bajo el sistema operativo Windows; en ella se reconocen las herramientas más comunes (Fig. 4.3) como pueden ser: abrir, guardar, copiar, imprimir, hacer, deshacer, etc.



Fig. 4. 3: Barra de herramientas de archivo

Por otro lado, el programa muestra herramientas más específicas del programa (Fig. 4.4), que sirven para rotar la pieza, mover el campo de visión, zoom, y opciones de representación de las superficies sólidas.



Fig. 4. 4: Barra de opciones de representación

- **Menú Cinta.** La cinta es un menú desplegable que sitúa al usuario en el plano que va a trabajar. Una vez elegido el plano, el programa se simplifica a un entorno 2D, en el que se pueden modelar elementos geométricos simples que darán forma a la pieza (Fig. 4.5).

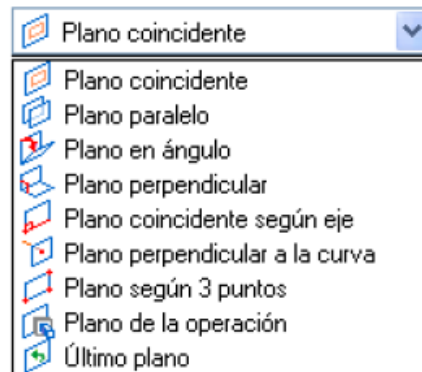


Fig. 4. 5: Desplegable para selección de plano de trabajo

- **Barra de Operaciones.** La barra de operaciones (Fig. 4.6) muestra todas las operaciones que se pueden realizar sobre la pieza. Más adelante se explicarán las más utilizadas en el modelado.

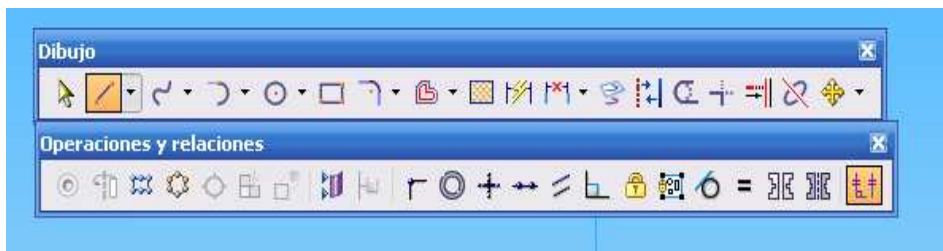


Fig. 4. 6: Barra de operaciones de modelado

- **Edgebar.** Es una ventana de ayuda para la gestión de tareas (Fig. 4.7). Esquematiza las operaciones que se realizan sobre la pieza en orden de ejecución con la posibilidad de modificarlas fácilmente.

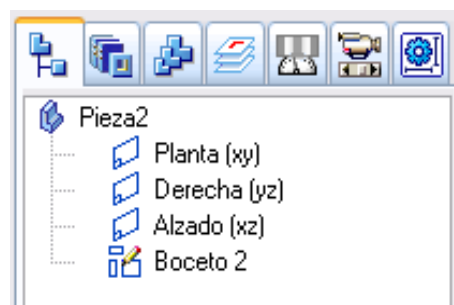


Fig. 4. 7: Lista de operaciones

- **Planos de referencia.** Son los planos principales que tiene el programa por defecto (Fig. 4.8). A partir de estos planos de referencia se pueden generar otros para modelar la pieza.

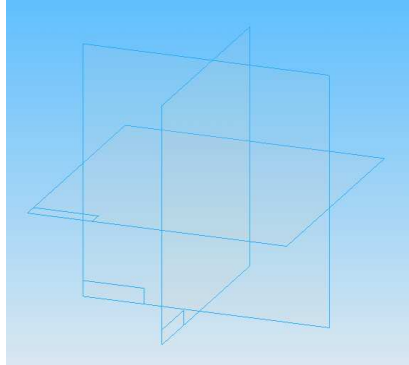


Fig. 4. 8: Planos de referencia

- **Barra de Estado o Smartstep.** Constituye una ayuda en línea. Identifica la herramienta (operación), y ofrece una pequeña ayuda sobre el proceso de utilización de la misma, cuando se pulsa o se señala el ratón sobre algún icono.

4.2.2 El interfaz SOLID EDGE V.18 en entorno “conjunto”

El entorno conjunto es donde Solid Edge V.18 permite unir piezas a través de relaciones de unión en función de la geometría de cada una de ellas.

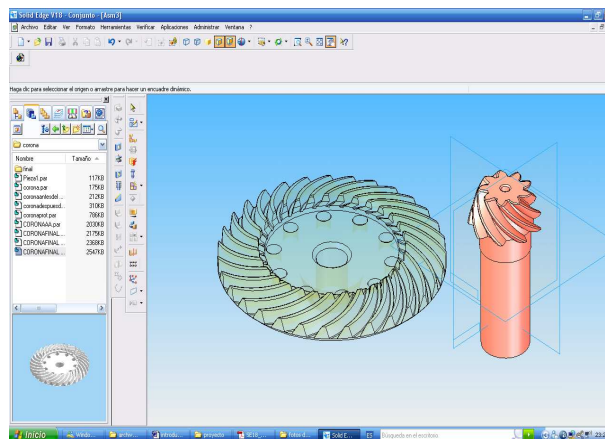


Fig. 4. 9: Interfaz del entorno conjunto

Se pueden introducir tantas piezas como sean precisas unir, e incluso, previsualizar la pieza a introducir (Fig. 4.9). Otra peculiaridad de este entorno es su barra de herramientas (Fig. 4.10). En ella hay disponibles opciones para desplazar las piezas, generar colocaciones simétricas, patrones de situación, etc. Pero la más importante es la herramienta “ensamblar”.



Fig. 4. 10: Barra de operaciones del entorno conjunto

Con la herramienta “ensamblar” se ejecuta la unión de las piezas. Para determinar el tipo de unión, existe un desplegable en la parte superior de la pantalla (Fig. 4.11) que permite elegir el tipo de relación geométrica entre las piezas que se visualizan.

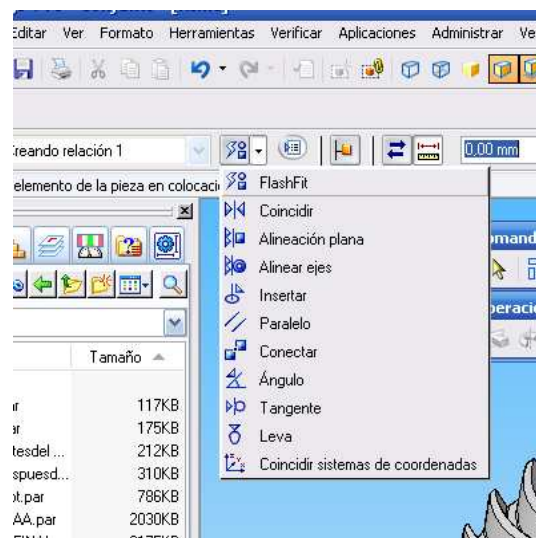


Fig. 4. 11: Desplegable opciones de ensamblaje

Una de las más útiles es la opción *Flash Fit* que permite colocar piezas tocando simplemente una cara o borde de cada una de ellas.

Frecuentemente es necesario aplicar más de una relación de ensamblaje debido a que pueden existir múltiples posibilidades.

4.2.3 Operaciones principales de modelado

En este apartado se describe brevemente las operaciones principales de Solid Edge, para que posteriormente el lector entienda adecuadamente los pasos de modelado que se han ejecutado:

- **Boceto.** Esta operación (Fig. 4.12) permite crear un boceto o perfil, que sirve de base para cualquier operación que se desee ejecutar. Este boceto será permanente y podrá ser utilizado en cualquier momento.

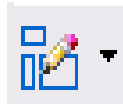


Fig. 4. 12: Icono Boceto

- **Protusión por Proyección.** Extruye un perfil perpendicular al plano seleccionado a partir de una silueta dibujada en dicho plano (Fig. 4.13).



Fig. 4. 13: Icono Protusión por Proyección

- **Protusión por revolución.** Extruye un perfil o boceto en torno a un eje (Fig. 4.14).



Fig. 4. 14: Protusión por revolución

- **Protusión por barrido.** Extruye un perfil en base a un perfil cerrado y una línea perpendicular, curva o recta, que lo conduce (Fig. 4.15).



Fig. 4. 15: Protusión por barrido.

- **Protusión helicoidal.** Extruye un perfil de forma helicoidal, a partir de una silueta y un eje (Fig. 4.16).



Fig. 4. 16: Protusión helicoidal

- **Vaciado por revolución.** Extruye un perfil haciendo vacío dentro de la pieza en torno a un eje, por barrido o por secciones análogo a la protusión (Fig. 4.17).

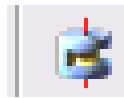


Fig. 4. 17: Vaciado por revolución

- **Patrón.** Multiplica una operación sencilla o compleja en la pieza, seleccionando la operación e indicando en número de repeticiones a lo largo del tramo de pieza a modelar (Fig. 4.18).



Fig. 4. 18: Patrón

- **Plano Coincidente.** Esta herramienta sirve para generar planos en cualquier parte de la pieza y en cualquier orientación (Fig. 4.19).



Fig. 4. 19: Plano Coincidente

4.3 Modelado de las piezas del diferencial

A continuación se describe el modelado paso a paso de cada una de las piezas del diferencial real cedido por la Universidad Carlos III de Madrid (Fig. 4.20 y Fig. 4.21).



Fig. 4. 20: Diferencial real a modelar



Fig. 4. 21: Diferencial real a modelar

4.3.1 Modelado de la corona hipoidal

Para el modelado de la corona se ejecuta el entorno “pieza sólida”. Posteriormente, para obtener un sólido que sirva de base para la corona hipoidal, se genera una protusión por revolución a partir de la sección bruta de la pieza, es decir, sin considerar el dentado del engranaje (Fig. 4.22).

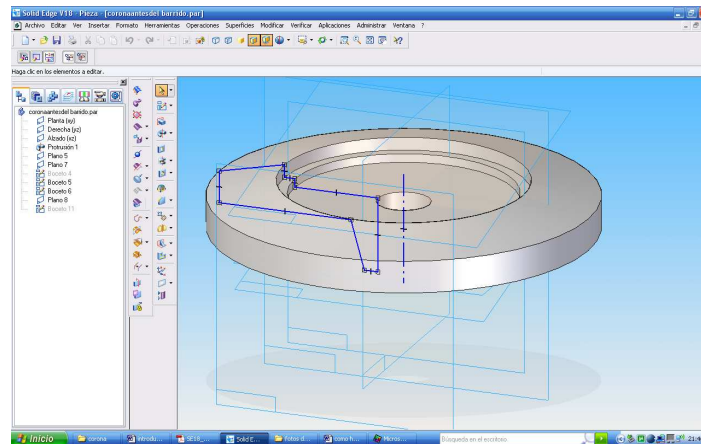


Fig. 4. 22: Protusión por revolución de la corona

Una vez generada la pieza bruta, se procede a hacer un vaciado para generar el hueco helicoidal entre dientes. Esto es quizás lo más complejo, ya que el hueco es de sección variable, y para conseguirlo es preciso hacer un boceto del hueco y una línea que guíe el barrido (Fig. 4.23).

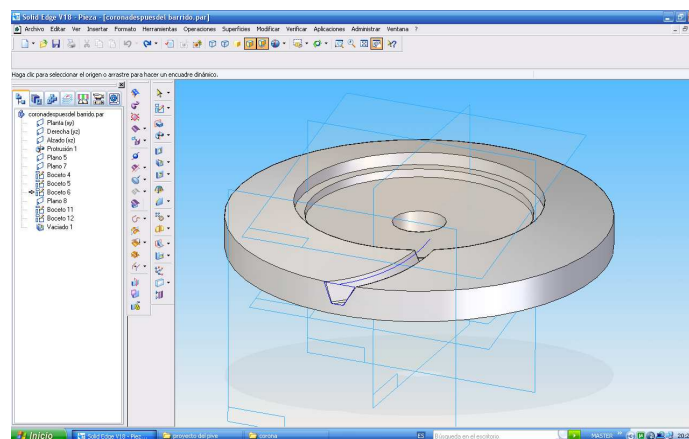


Fig. 4. 23: Vaciado por barrido del hueco interdental

Cuando se ha ejecutado el hueco, el siguiente paso es utilizar la herramienta *patrón*, anteriormente descrita, que permite copiar una operación a lo largo de la pieza con exactitud cuantas veces sea necesaria, de una vez y automáticamente (Fig. 4.24).

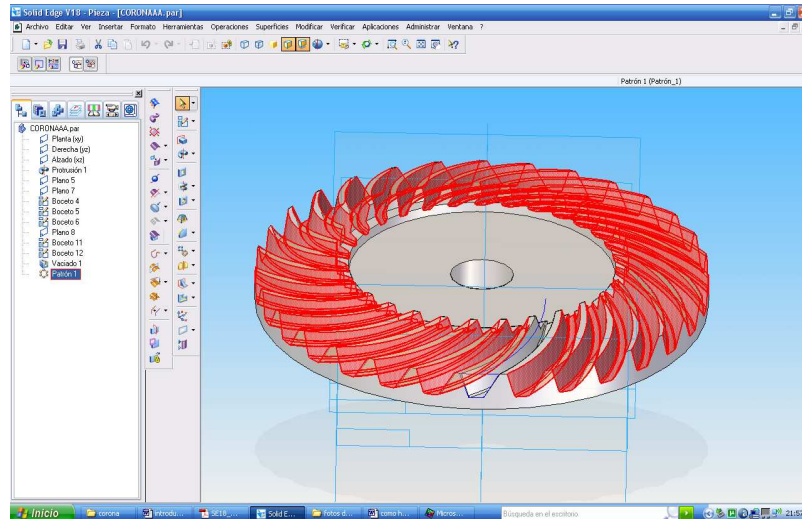


Fig. 4. 24: Repetición del vaciado con la herramienta patrón

Por último, se remata la pieza creando los agujeros para que los tornillos unan la corona al casquete. La manera de hacerlo es usar la herramienta *agujero*, que funciona indicándole el radio, la profundidad, y el tipo de rosca. Y después, se completan los demás utilizando la herramienta *patrón* (Fig. 4.25).

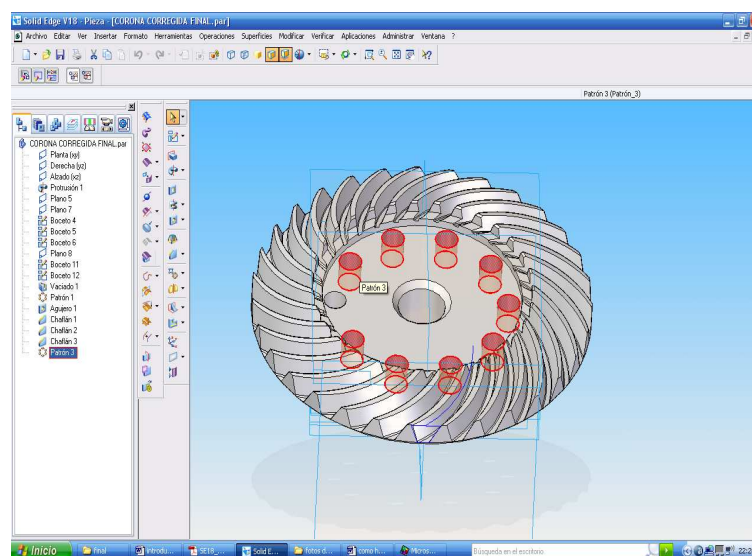


Fig. 4. 25: Repetición de la operación agujero

Finalmente se puede comparar el resultado del modelado (Fig. 4.26) frente a la corona real en la (Fig. 4.27).

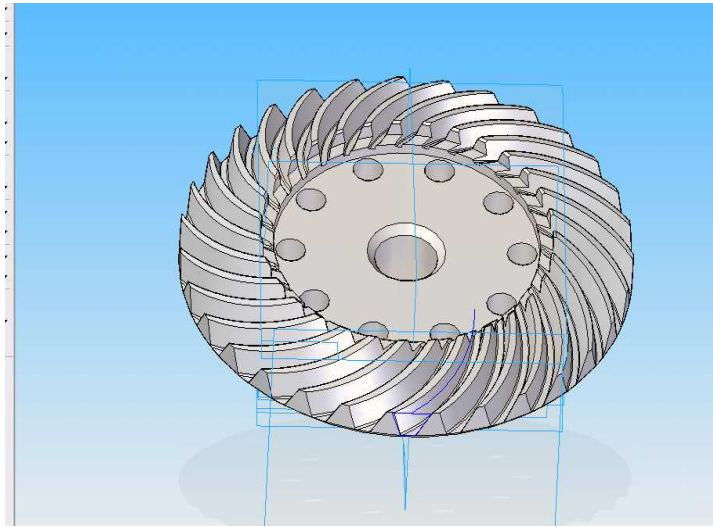


Fig. 4. 26: Corona modelada



Fig. 4. 27: corona real

4.3.2 Modelado del casquete

El casquete es la parte del diferencial que alberga y protege la transmisión entre satélites y planetarios. Esta pieza es muy complicada de modelar porque es acero forjado con geometría ambigua. Para empezar a modelarlo se realiza una protusión por revolución (Fig. 4.28).

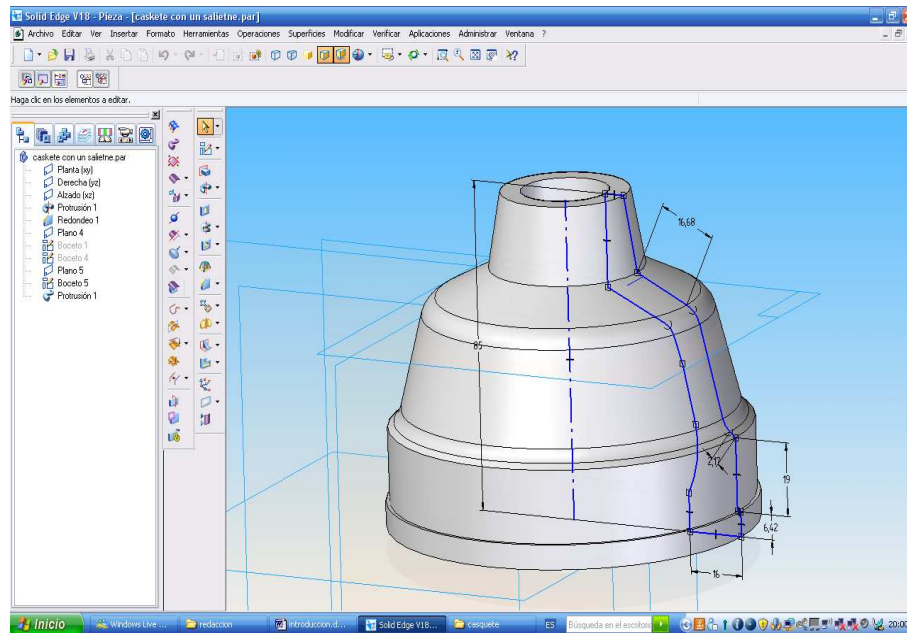


Fig. 4. 28: Protusión por revolución para el casquete

Para conseguir una reproducción fiel al modelo real, es necesario hacer dos salientes laterales con protusión por proyección y vaciado con contorno circular (Fig. 4.29).

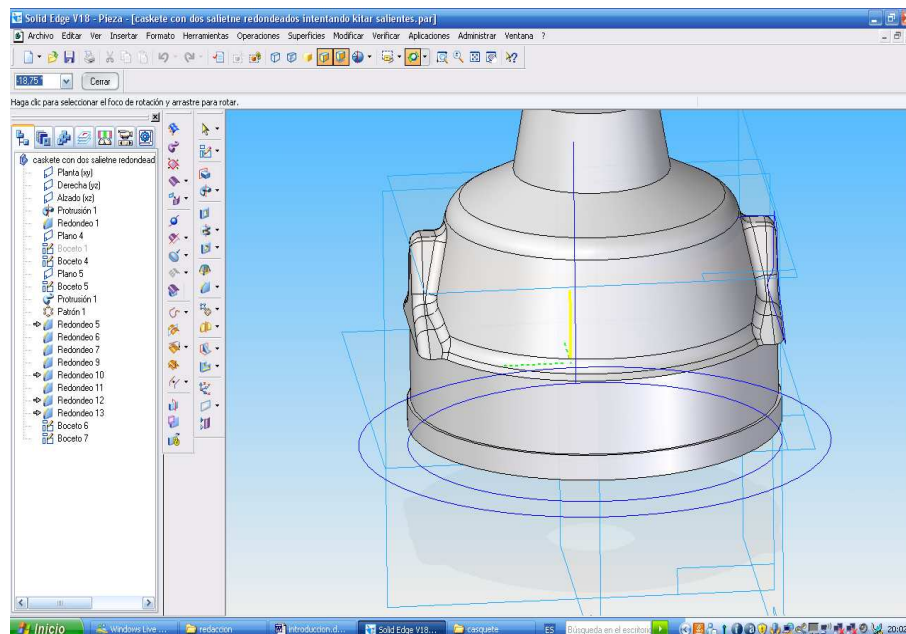


Fig. 4. 29: Protusión de salientes laterales

Por último, se ejecuta la herramienta “agujero” análoga al modelado de la corona (Fig.4.30). En la Fig. 4.31 se muestra el casquete real.

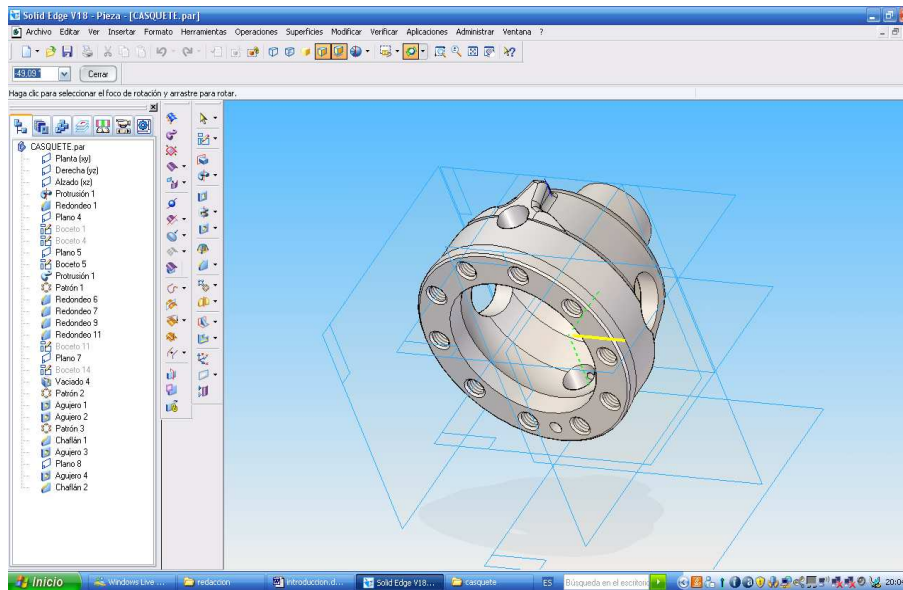


Fig. 4. 30: Generación de agujeros



Fig. 4. 31: Casquete real

4.3.3 Modelado del planetario

El planetario es la pieza que engrana con los satélites y transmite el movimiento a los palieres. Para modelarla se utiliza la protusión por revolución (Fig. 4.32).

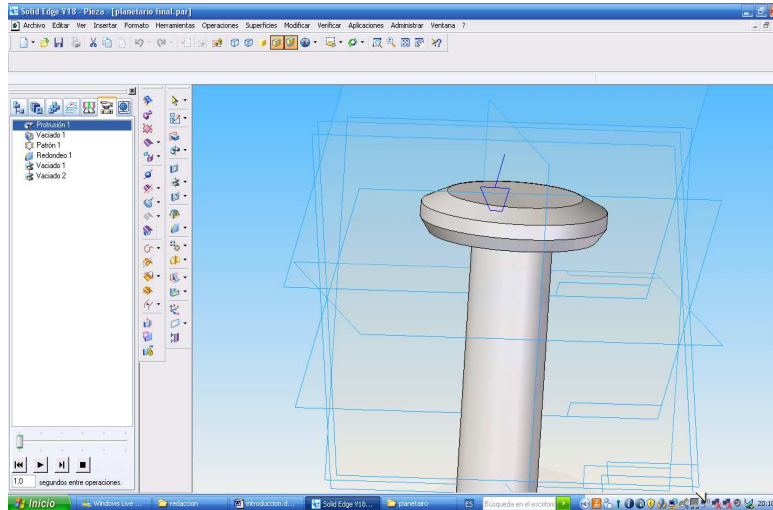


Fig. 4. 32: Protusión por revolución de los planetarios

Y para hacer el dentado recto, se hace un vaciado por barrido, con una sección tipo del hueco interdental y un línea perpendicular oblicua que dejará los espacios adecuados para un engrane perpendicular (Fig. 4.33 y Fig. 4.34).



Fig. 4. 33: Planetario modelado



Fig. 4. 34: Planetario real

4.3.4 Modelado del satélite

Los satélites son los engranes que junto a los planetarios “distribuyen” el movimiento cuando el vehículo traza una curva. Su modelado es bastante parecido al de los planetarios. Es suficiente con hacer una protusión por revolución (Fig. 4.35).

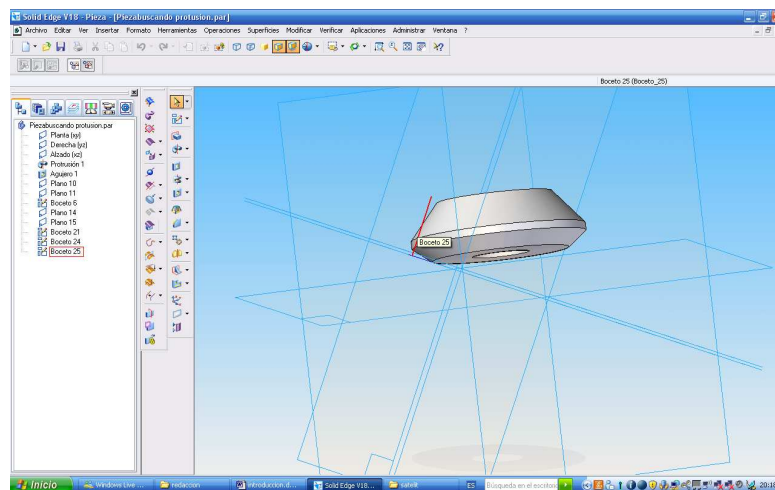


Fig. 4. 35: Modelado del satélite

Y, después, generar huecos interdentales con un vaciado por barrido y un patrón. Lo realmente complicado en el modelado de esta pieza es encontrar la orientación adecuada del plano auxiliar que permitirá hacer el boceto para el vaciado por barrido (Fig. 4.36).

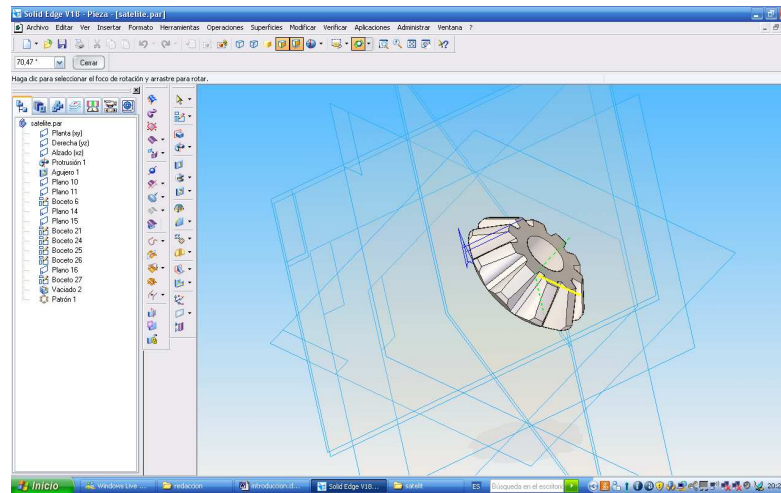


Fig. 4. 36: Vaciado interdental de los satélites

4.3.5 Modelado del piñón de ataque

El piñón de ataque es la pieza que transmite el movimiento a todo el mecanismo, por ello es una pieza robusta, que necesita soportar grandes esfuerzos de torsión, y además para llevar a cabo un movimiento suave, necesita que el engrane con la corona sea progresivo, y lo hace gracias a su dentado helicoidal. Para ello, también se comienza con un protusión por revolución (Fig. 4.37). Y, a diferencia del satélite y el planetario, es necesario crear el diente mediante una protusión helicoidal cónica, y un patrón (Fig. 4.38).

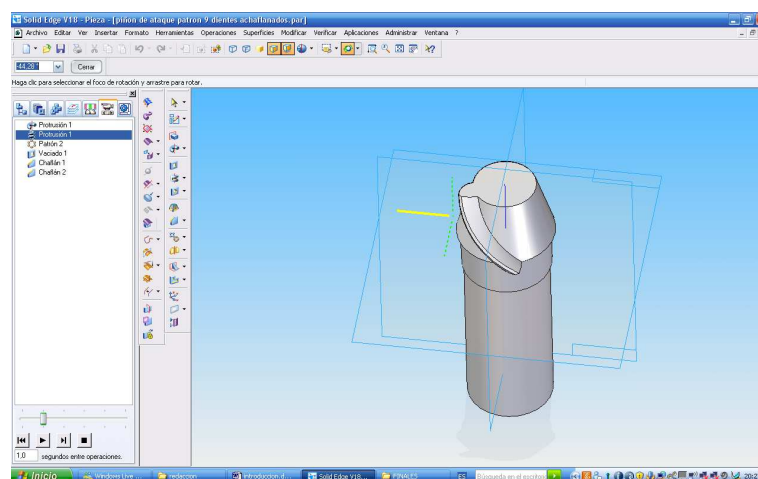


Fig. 4. 37: Protusión por revolución y protusión helicoidal cónica

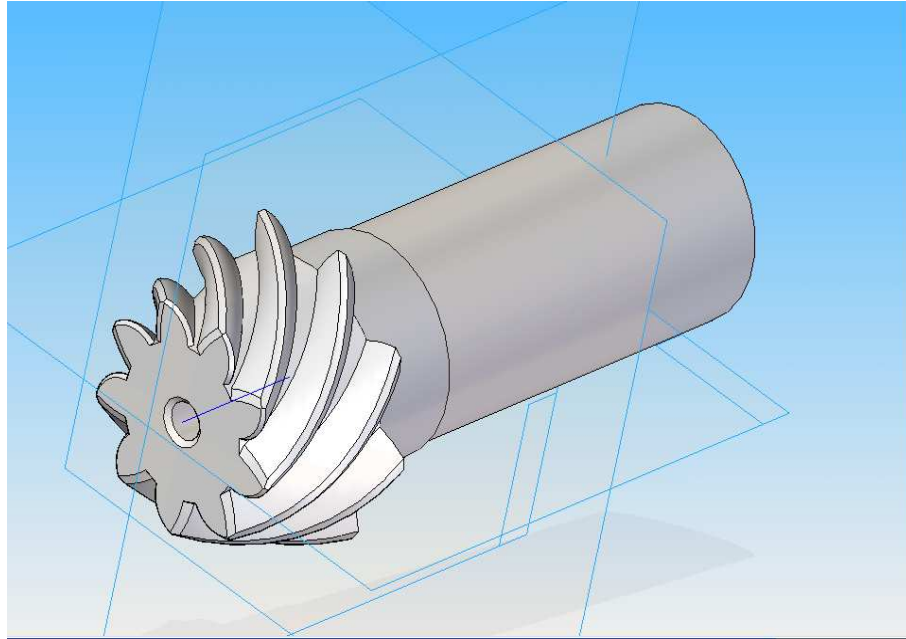


Fig. 4. 38: Piñón de ataque modelado

En la figura 4.39 se muestra el piñón de ataque real.



Fig. 4. 39: Piñón de ataque real

4.4 Ensamblaje del conjunto mecánico diferencial

Una vez diseñadas todas las piezas, es necesario abrir el entorno conjunto para unir todas las piezas con la relación geométrica adecuada.

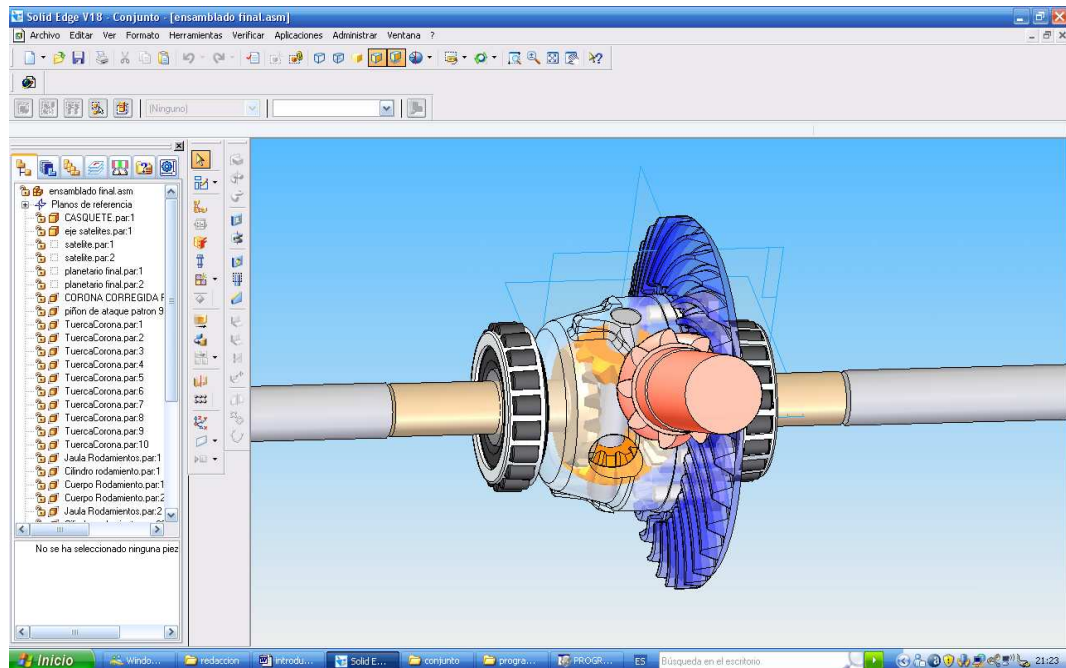


Fig. 4. 40: Ensamblaje final del diferencial

Para ensamblarlo, se parte de la corona hipoidal y del casquete. Al ser piezas con grandes superficies planas en común, se unen fácilmente con la herramienta de ensamblaje *coincidir*, de esta manera ya se tiene una referencia para todas las demás piezas. A continuación, con la herramienta *alinear ejes* se consigue introducir los satélites en el eje conductor del casquete. Y una vez colocados los satélites se deduce el ensamblaje de los planetarios para que sus dientes coincidan; esto se ejecuta mediante otra alineación de ejes con el hueco del casquete y a continuación se ajustan distancias con la herramienta *mover pieza* (Fig. 4.40). De la misma manera, se pueden colocar los rodamientos, palieres y ruedas. Y por último, el piñon de ataque, que al estar desplazado ligeramente de la perpendicularidad de todo el mecanismo, lo más conveniente es usar la herramienta de ensamble *Flash Fit* que permite situar la pieza en el lugar adecuado indicando las superficies que se van a tocar.

Finalmente se obtiene un conjunto mecánico diferencial al que se le ha añadido, de forma esquemática, los palieres y las ruedas con el fin de que el lector sitúe, de manera rápida, los ejes de entrada y de salida de movimiento (Fig. 4.41).

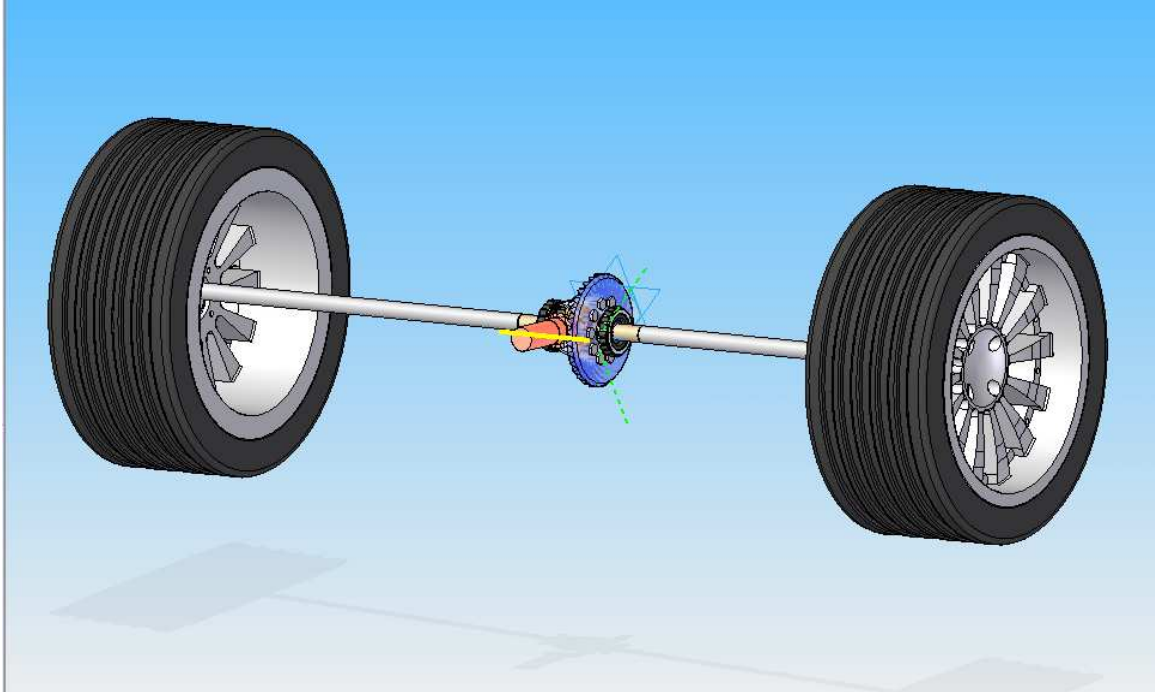


Fig. 4. 41: Vista general del conjunto

CAPÍTULO 5: ANIMACIÓN EN ENTORNO VRML

En este capítulo se completa el núcleo del proyecto aplicando movimiento al conjunto mecánico diferencial con un lenguaje de modelado de realidad virtual llamado VRML. Dicho programa es un lenguaje de definición de escenas cuyo objetivo es la descripción de entornos virtuales 3D con posibilidad de transmitirse e interrelacionarse a través del *World Wide Web*. Aunque VRML es totalmente capaz de generar cualquier objeto en realidad virtual con independencia, su proceso es mucho más complejo, por ello se han modelado las piezas con un programa especializado como Solid Edge V.18 para exportar posteriormente las piezas a un entorno VRML. Por tanto, a continuación se describe qué es VRML, su historia a grandes rasgos, su utilidad, su funcionamiento, y su aplicación a este proyecto creando tres escenas de movimiento posibles a las que se puede enfrentar un diferencial mecánico libre en la realidad.

5.1 Inicios del VRML y su evolución

En 1989, Rikk Carey y Paul Strauss de Silicon Graphics Inc., iniciaron un nuevo proyecto con el fin crear una infraestructura para aplicaciones interactivas con gráficos tridimensionales. Los dos objetivos originales eran [14]:

1. Construir un ambiente de desarrollo que permitiera la creación de una extensa variedad de aplicaciones interactivas con gráficos tridimensionales.
2. Utilizar este ambiente de desarrollo para construir una nueva interfaz de usuario tridimensional.

La primera fase del proyecto se concentraba en generar la semántica y los mecanismos para la plataforma de trabajo. En 1992 se liberó el Iris Inventor 3D, herramienta la cual definía gran parte de la semántica que hoy en día conforma a VRML. Una parte importante del Iris Inventor era que el formato del archivo utilizado para guardar los objetos de la aplicación era de poco tamaño y fácil utilizar. En 1994 se liberó la segunda gran versión de Inventor llamada Open Inventor. Está era compatible para diferentes plataformas y basada en OpenGL de Silicon Graphics. El manual de referencia que describe los objetos y el formato de archivo de Open Inventor fueron después utilizados por Gavin Bell para escribir la primera propuesta para la especificación de VRML 1.0 [14].

Después de varias propuestas se escogió la sintaxis de OpenInventor de Silicon Graphics como base de un formato de descripción de objetos geométricos texturizados, agregando la posibilidad de combinar objetos guardados remotamente en la red. De esta manera nació VRML 1.0 que aunque solo era una solución parcial, era una muestra de lo que VRML podría llegar a ser.

VRML 1.0 se caracterizó por ser un lenguaje para la descripción de mundos virtuales estáticos, que cumplía tres requisitos fundamentales [15]:

- Era independiente de la plataforma donde se ejecute el visualizador.
- Tenía capacidad para trabajar de un modo eficiente con conexiones lentas.
- Extensibilidad, o capacidad de actualizarse.

El principal problema de esta versión de VRML se encontraba en la ausencia de semántica. Se trataba simplemente de un lenguaje de especificación de geometrías.

Debido a los problemas de la versión 1.0, surgió la versión 1.1 del lenguaje que añadía algunas extensiones para la incorporación de sonido y animaciones simples. Paralelamente, la totalidad de las empresas que desarrollaban *browsers* VRML

incorporaban sus propias extensiones. Esto hizo peligrar la existencia de un estándar por lo que inmediatamente se comenzó el trabajo para la especificación de la versión 2.0 de VRML. Así, se creó el V. A. G., VRML Architecture Group [14].

Seis propuestas fueron presentadas para la versión 2.0 en las que se pretendía eliminar el problema de la semántica. Tras el correspondiente proceso de estudio y revisión de las mismas se decidió aceptar la propuesta Moving Worlds de SGI, Sony y Mitra [14]. Esta versión aportó grandes mejoras sobre su predecesora, que se reflejan en las siguientes características:

- **Semántica:** Esta es la principal mejora de la versión 2.0. que se consigue mediante la incorporación de nodos de comportamiento que pueden ejecutar *scripts* realizados en lenguajes como C++, Java, JavaScript y VRMLScript. Para ello es necesaria una comunicación entre nodos, comunicación que se realiza mediante eventos que se propagan a través de rutas (conexiones entre nodos generadores y receptores de eventos).
- **Mayor poder descriptivo:** La nueva versión 2.0 de VRML permite la definición de mundos más reales, al incorporar superficies irregulares, condiciones atmosféricas como niebla y nodos de sonido espacial.
- **Mayor interactividad:** Mediante la incorporación de nodos sensores de proximidad, contacto, visibilidad y tiempo.
- **Animación:** Mediante la incorporación de interpoladores.
- **Modularidad:** Mediante la generación de prototipos que agrupan conjuntos de nodos predefinidos.

De cara al futuro, la razón de ser de esta herramienta será la socialización. Es necesario definir protocolos para seguir y sincronizar los comportamientos de objetos programados y de usuarios interactuando en tiempo real. Se está trabajando sobre una propuesta de Silicon Graphics llamada *Living Worlds* [14] cuya principal característica es la creación de un entorno para múltiples usuarios, representados cada uno de ellos por su *avatar* (representación humana).

El mundo de VRML y los lenguajes de navegación virtual se consideran el futuro y la evolución al texto y las imágenes que ofrece HTML. Dentro de un tiempo se podrá navegar no por webs sino por verdaderos mundos en 3D.

5.2 Metodología de trabajo con VRML

Un proyecto VRML, de cualquier envergadura, puede ser dividido en las siguientes etapas:

1.- Especificación. En este paso se procede a realizar la descripción del proyecto, puntualizar el perfil del usuario, indicar los recursos necesarios y establecer el modo de interacción con el usuario. Es decir, en función del problema definido, se establece con precisión qué se propone lograr con el entorno virtual de enseñanza-aprendizaje.

2.- Planificación. La complejidad del proceso de elaboración de los materiales depende de la calidad pedagógica y académica deseada para el producto final, y de la variedad, alcance y complejidad en la integración de los medios empleados. En esta etapa se decide cuándo y cómo construir.

3.- Muestreo. En esta etapa se recaban los antecedentes acerca del objeto a modelar, es decir, toda la información necesaria para un adecuado modelado de los objetos que poblarán el mundo virtual a construir. Esta etapa incluye a su vez:

- Elaboración de contenidos: Se seleccionan los contenidos y se recopila información específica.
- Preproducción del material: En este paso se procede a darle al material su estructura definitiva, donde, de acuerdo a las características propias del medio utilizado, se presentarán los contenidos y actividades.
- Supervisión: En esta etapa se lleva a cabo una nueva revisión de lo realizado. El objetivo fundamental de este paso es comprobar que no se hayan producido distorsiones en los contenidos.

4.- Diseño. Una vez conseguidos todos los antecedentes acerca de los diversos objetos a modelar, por medio del proceso de muestreo, se procede al diseño del mundo virtual. Esta etapa incluye a su vez las siguientes etapas:

- Introducción al diseño: Antes de iniciar el proceso de diseño, se revisa la sintaxis y estructura básica de un archivo con extensión “.wrl”.

- **Identificación de objetos:** Para ello, se sigue un estudio similar al análisis orientado a objetos, identificando los mismos a partir de la evaluación de las herramientas que se utilizarán y los antecedentes obtenidos a partir del muestreo realizado. También se definen los sucesos más significativos para los componentes de cada sistema.
- **Especificación de atributos:** En este caso, se entiende por atributos aquellos datos que le otorgan cualidades físicas relevantes a un objeto, para su modelado posterior. Con los objetos ya identificados, se procede a realizar una lista adecuada para la finalización del proyecto a realizar.
- **Identificación de eventos:** Un evento involucra cambios en los valores de uno o más de sus atributos. Algunos de los cambios pueden ser de posición, color, tamaño, sonido, entre otros.
- **Comunicación entre objetos:** Los objetos del mundo virtual se comunicarán, por medio de mensajes, quiénes deben ir encaminados por una ruta. Se identifica la existencia de estas interacciones, teniendo en cuenta la información obtenida y, además, se indicarán objetos que presenten variaciones en su movimiento o características físicas.

5.- Construcción. Existen muchos métodos usados para crear VRML. Cada compañía tiene sus propios conjuntos de herramientas. Para el presente proyecto, las piezas modeladas en Solid Edge se han guardado con extensión “.wrl”, para exportarse posteriormente al entorno VRML.

6.- Pruebas. Debido a que VRML no es un lenguaje de programación (no se compila antes de lanzarlo) cualquier detección de errores en la sintaxis de los archivos se conoce en el momento en que estos son cargados en memoria. Esto permite conocer los posibles errores cometidos antes de cargar la simulación, lo que disminuye el tiempo transcurrido entre el momento en el que se detecta el problema, y el momento en el que se soluciona.

7.- Publicación. En esta etapa simplemente se coloca el archivo en un servidor Web, si fuera oportuno

5.3 Características del lenguaje

El funcionamiento de VRML se basa en nodos, eventos. Los nodos permiten construir objetos 3D, crear luces, aplicar texturas a los objetos, asociar ficheros de audio a determinados sucesos o utilizarlos como sonido de fondo, detectar la proximidad de un objeto a otro, etc. Y para animar un mundo virtual se recurre a la conexión de unos nodos a otros, de manera que puedan intercambiar información por medio de eventos. Estos eventos pueden estar producidos por un contador o como respuesta de un objeto a otros eventos que ha recibido. Por ejemplo, pulsando con el ratón sobre un interruptor se puede hacer que una luz se encienda o se apague; o arrastrando el ratón pulsado sobre un objeto puede hacerse que éste se desplace sobre una superficie o gire sobre sí mismo.

5.3.1. Software necesario para trabajar con VRML

Los programas necesarios para poder editar y visualizar imágenes son:

- **Un editor de textos.** Cualquier editor de texto plano. Los editores de textos "planos" se distinguen de los procesadores de textos en que se usan para escribir sólo texto, sin formato y sin imágenes, es decir sin diagramación. Aunque existen editores que asisten, corrigen, esquematizan y facilitan la programación y son muy útiles cuando se manejan grandes cantidades de información, o bien no se conocen correctamente los nodos y los campos. Para este proyecto se dispone del editor VrmIpad de ParallelGraphics.
- **Un visualizador VRML.** Es necesario para ver las imágenes e interactuar en un entorno en tres dimensiones. El visualizador se instala en el navegador y pone a disposición del usuario una interfaz para moverse dentro del mundo virtual e incluso accionar animaciones. El visualizador que se usa en este proyecto es el Cortona.

5.3.2. Normas básicas de sintaxis en VRML

Todos los programas de VRML tienen que respetar ciertas normas de notación y de metodología [25]:

- VRML es un lenguaje sensible a mayúsculas y minúsculas, lo cual ha de ser tenido en cuenta a la hora de asignar nombres.
- Todos los nodos han de comenzar siempre con letra mayúscula.
- Los campos de los nodos deben comenzar siempre con letra minúscula.
- Los números se escriben en punto flotante.
- Utilizar una línea distinta para cada nodo, para cada campo y para cada valor de cada campo.
- Tabular cada línea, según su jerarquía
- Colocar cada símbolo de cierre en el nivel de tabulación que le corresponda.
- Poner las líneas de comentario necesarias al mismo nivel que lo que se comenta.

5.3.3. Estructura básica

A continuación se muestra la estructura básica de un programa tipo:

- **Cabecera:** indica al navegador que está recibiendo un archivo de VRML, y la versión correspondiente del lenguaje. La parte del “utf8” dice al navegador qué código de texto debe emplear, VRML 1.0 usaba “ascii” o “utf8”, pero la versión 2 tiene sólo “utf8”. Por tanto en cabecera siempre se escribe:

#VRML V2.0 utf8

- **Nodos:** son los elementos básicos del lenguaje. Cada nodo define una característica de la escena (un objeto 3D, un color, una fuente luminosa, un fichero de audio, un sensor), y a su vez puede ser parte de otro nodo, de este modo los nodos pueden anidarse entre sí creando un tipo de jerarquía de nodos.

Cada nodo poseerá una serie de atributos, que lo caracterizan. En la tabla 5.1 y 5.2 se enumeran los más importantes [16].

Nodos comunes	Apariencia	Nodos de agrupamiento	Sensores
<i>AudioClip</i>	<i>Appearance</i>	<i>Anchor</i>	<i>CylinderSensor</i>
<i>DirectionalLight</i>	<i>FonStyle</i>	<i>Billboard</i>	<i>PlaneSensor</i>
<i>Pointlight</i>	<i>ImageTexture</i>	<i>Collision</i>	<i>ProximitySensor</i>
<i>Script</i>	<i>Material</i>	<i>Group</i>	<i>SphereSensor</i>
<i>Shape</i>	<i>MovieTexture</i>	<i>Transform</i>	<i>TimeSensor</i>
<i>Sound</i>	<i>PixelTexture</i>		<i>TouchSensor</i>
<i>Spotlight</i>	<i>TextureTransform</i>		<i>VisibilitySensor</i>
<i>Worldinfo</i>			

Tabla 5. 1: Nodos 1

Geometría	Interpoladores	Agrupamientos especiales	Nodos de activación excluyente
<i>Box</i>	<i>ColorInterpolator</i>	<i>Inline</i>	<i>Background</i>
<i>Cone</i>	<i>CoordinateInterpolator</i>	<i>LOD</i>	<i>Fog</i>
<i>Cylinder</i>	<i>NormalInterpolator</i>	<i>Switch</i>	<i>NavigationInfo</i>
<i>ElevationGrid</i>	<i>OrientationInterpolator</i>		<i>Viewpoint</i>
<i>Extrusion</i>	<i>PositionInterpolator</i>	Propiedades geométricas	
<i>IndexedFaceSet</i>	<i>ScalarInterpolator</i>	<i>Color</i>	
<i>IndexedLineSet</i>		<i>Coordinate</i>	
<i>PointSet</i>		<i>Normal</i>	
<i>Sphere</i>		<i>TextureCoordinate</i>	

Tabla 5. 2: Nodos 2

- **Campos:** Los nodos a su vez, contienen campos que describen propiedades. Cada tipo de nodo tiene una serie de valores predeterminados para todos sus campos, de forma que cuando se utilizan en una escena, únicamente se deben

indicar aquellos campos que se quieran modificar [16]. Este tipo de información puede ser de carácter numérico, alfabético, color, etc. y puede ser un valor único (SF, *Single File*) o pueden ser múltiples valores (MF, *Multiple file*). En la tabla 5.3 se muestran todos los campos con una descripción de cada uno de ellos.

SFBool	Es un valor <i>booleano</i> simple que puede tomar el valor “TRUE” o “FALSE”.
SFColor & MFFloat	Este campo contiene un color simple, conformado por tres números de punto flotante entre 0 y 1 que corresponden a los valores rojos, verdes y azules de ese color.
SFFloat & MFFloat	SFFloat es un valor simple de punto flotante. MFFloat son varios valores de punto flotante. Puede tomar el valor de “TRUE” o “FALSE”.
SFImage	SFImage es una imagen bidimensional.
SFInt32 & MFInt32	Uno solo o una lista de números enteros de 32 bits. Éstas pueden estar en forma decimal o hexadecimal. El valor se define ente $(+\infty, -\infty)$.
SFNode & MFNode	SFNode es un solo nodo, y MFNode es una lista de nodos.
SFRotation & MFRotation	Estos campos especifican una rotación sobre un eje. Esto es realizado por cuatro números de punto flotante (X Y Z α).
SFString & MFString	Contiene una lista de caracteres en el set de caracteres utf-8, debiendo estar entre comillas.
SFTime & MFTime	Un valor o una lista de valores de tiempo.
SFVec2f & MFVec2f	Uno solo o una lista de vectores 2D (XY).
SFVec3f &	Uno solo o una lista de vectores 3D (X Y Z).

Tabla 5. 3: Campos de propiedades

5.3.4. Descripción de los nodos

En este apartado se describe la estructura de los nodos más comunes.

5.3.4.1 Nodo *Shape*

Para poder visualizar figuras geométricas primitivas es necesario utilizar el nodo *Shape*. Este nodo tiene dos campos: *geometry* y *appearance*.

El campo *geometry* es donde se especifica la geometría de algún objeto 3D. Estas primitivas son nodos. A continuación, se muestra cómo definir cada una de las primitivas existentes:

- ***Box***: Se define una caja de X unidades de ancho, Y de alto y Z de fondo. Este objeto y los definidos posteriormente quedan definido de forma que su centro coincide con el origen de coordenadas locales (Fig. 5.1).

```
Shape {  
  Geometry Box size {X Y Z}  
}
```

Fig. 5. 1: Código de generación de un cubo

- ***Sphere***: Se define una esfera de X unidades de radio (Fig. 5.2).

```
Shape {  
  Geometry Sphere {radius X}  
}
```

Fig. 5. 2: Código de generación de una esfera

- **Cone:** Se define un cono con una base de X unidades de radio y una altura de Y unidades (Fig. 5.3).

```
Shape {  
    Geometry Cone {  
        Bottom Radius X  
        Height Y  
    }  
}
```

Fig. 5. 3: Código de generación de un cono

- **Cylinder:** Se define un cilindro con una base de X unidades de radio y una altura de Y unidades (Fig. 5.4).

```
Shape {  
    Geometry Cylinder {  
        Radius X  
        Height Y  
    }  
}
```

Fig. 5. 4: Código de generación de un cilindro

Para definir la apariencia de las figuras primitivas (y en general de todos los objetos) se debe definir un color o una textura a aplicar a su superficie. El nodo *Shape* dispone de un campo llamando *appearance* donde se definen las características del material del objeto. Para hacerlo, se utiliza el nodo *Appearance* de VRML. Este nodo tiene diversos campos entre los que se encuentra material. Para definir este campo se

debe utilizar el nodo *Material*, que a su vez posee un campo llamado *diffuseColor*, que hace referencia al color que refleja el objeto, es decir, el color del objeto. Los colores en VRML se definen en RGB (*Red-Green-Blue* o Rojo-Verde-Azul). Los valores de R, G y B deben estar dentro del rango [0,1].

5.3.4.2 Nodo *Group*

El nodo *Group* permite unir un conjunto de nodos de forma que actúen como una entidad única, pero sin efectuar ninguna transformación en ellos. La principal característica de este tipo de grupo es que los objetos son todos creados en el mismo punto (en el centro del escenario de realidad virtual). El campo *children* dentro del nodo *Group* puede contener otros nodos. De este modo los nodos pueden anidarse entre sí, creando una jerarquía de nodos. En la Fig. 5.5 se muestra un ejemplo de cómo se puede definir este nodo.

```

Group {
    eventIn      MFNode addChildren
    eventIn      MFNode removeChildren
    exposedField MFNode children []
    field        SFVec3f bbox Center 000 # (-∞,∞)
    field        SFVec3f bbox Size -1 -1 -1 # (0, ) ó -1,-1,-1
}

```

Fig. 5. 5: Código del nodo genérico *Group*

5.3.4.3 Nodo *Inline*

El nodo *Inline* permite incluir elementos, conjuntos o mundos ya creados dentro del que se está construyendo, importando ese fichero desde su ubicación en el ordenador o “URL”.

En la Fig. 5.6 se muestra el código de VRML necesario para importar alguno de los mundos ya creados (en el caso de este proyecto, las piezas modeladas en *Solid Edge*).

```

Inline {
  exposedField MFString url []
  field SFVec3f bboxCenter 000 # (-∞,∞)
  field SFVec3f bboxSize -1 -1 -1 # (0, ) ó -1,-1,-1
}

```

Fig. 5. 6: Código del nodo genérico **Inline**

5.3.4.4 Nodo *Transform*

Sirve para situar, orientar y escalar los objetos según las necesidades dentro del entorno que se esté construyendo. Este nodo aplica las transformaciones geométricas básicas, que son: traslación, rotación y escalado (Fig. 5.7).

```

Transform {
  eventIn MFNode addChildren
  eventIn MFNode removeChildren
  exposedField SFVec3f center 000 # (-∞,∞)
  exposedField MFNode children []
  exposedField SFRotation rotation 0010# [-1,1],(-∞,∞)
  exposedField SFVec3f scale 111# (0,∞)
  exposedField SFRotation scale Orientation 0010# [-1,1],(-∞,∞)
  exposedField SFVec3f translation 000 # (-∞,∞)
  field SFVec3f bboxCenter 000 # (-∞,∞)
  field SFVec3f bboxSize -1-1-1# (0,∞) o -1,-1,-1
}

```

Fig. 5. 7: Código del nodo genérico **Transform**

El campo *translation* define el desplazamiento que sufre el objeto que se desea trasladar. Así pues, se define un desplazamiento a lo largo de los ejes (X Y Z). Las traslaciones son relativas y, por lo tanto, se suman a la posición actual de los objetos.

El campo *rotation* define la rotación que sufre el objeto a modificar. Este campo tiene en realidad dos partes. Los tres primeros valores determinan un vector en el espacio 3D que corresponde a la orientación del eje de rotación de la transformación. El eje de rotación coincide con los ejes coordenados cuando:

- Eje rotación X: Se define de la siguiente forma: (1 0 0).
- Eje rotación Y: Se define de la siguiente forma: (0 1 0).
- Eje rotación Z: Se define de la siguiente forma: (0 0 1).

Este sistema resulta complejo cuando el eje de rotación no es paralelo a un eje coordenado.

La segunda parte es el ángulo de la rotación que se quiere aplicar (expresado en radianes) respecto al eje definido en la primera parte. Al estar en un sistema de ejes a derechas, los ángulos positivos resultan en rotaciones en sentido antihorario (y los ángulos negativos en sentido horario).

El campo *scale* define el escalado no uniforme que sufre el objeto que se desea modificar.

5.3.4.5 Nodo *TimeSensor*

VRML dispone de una herramienta muy potente, el sensor de tiempo designado como *Timesensor* (Fig. 5.8). Este es un reloj que se puede utilizar para aprovechar el paso del tiempo como motor para mover objetos, cambiar su color, variar orientaciones, etc. Este se basa en el reloj real del sistema, midiendo el tiempo actual en segundos transcurridos a partir de las 00:00 horas del 1º de enero de 1970 [16].

Los campos más característicos son el *cycleInterval*, en el que se pone el ciclo de tiempo elegido para mover los objetos, y *loop*, que sirve para que ese ciclo de tiempo elegido se repita una vez termine.

```

TimeSensor {
    exposedField SFTime cycleInterval 1 # (0,∞)
    exposedField SFBool enabled TRUE
    exposedField SFBool loop FALSE
    exposedField SFTime startTime 0 # (-∞,∞)
    exposedField SFTime stopTime 0 # (-∞,∞)
    eventOut SFTime cycleTime
    eventOut SFFloat fraction_changed 0 # [0,1]
    eventOut SFBool isActive
    eventOut SFTime time
}

```

Fig. 5. 8: Código del nodo genérico Timesensor

5.3.4.6 Nodo *TouchSensor*

El sensor de tacto permite detectar cuándo el observador apunta a un objeto con el cursor y cuándo oprime el botón del ratón. Este nodo es útil para activar o desactivar nodos o eventos. El código del nodo *TouchSensor* se indica en la Fig. 5.9.

```

TouchSensor {
    exposedField SFBool enable TRUE
    eventOut SFVec3 hitNormal_changed
    eventOut SFVec3 hitPoint_changed
    eventOut SFVec2 hitTexCoord_changed
    eventOut SFBool isActive
    eventOut SFBool isOver
    eventOut SFTime touchTime
}

```

Fig. 5. 9: Código del nodo genérico Touchsensor

5.3.4.7 Interpoladores

La interpolación lineal (que es la que utiliza VRML) es un concepto matemático que permite definir dos puntos (en cualquier dimensión) y calcular un punto intermedio sobre la recta que los une a partir de especificar el tanto por ciento del recorrido entre los dos puntos. Aunque existen diversos tipos, en la Fig. 5.10 se describe el interpolador de orientación *OrientatorInterpolator*, que define la orientación de cada engranaje en el diferencial cuando está en movimiento.

```

OrientatorInterpolator {
    eventIn      SFFloat set_fraction      #(-∞,∞)
    exposedField MFFloat key []           #(-∞,∞)
    exposedField MFRotation keyValue []  #[-1,1],(-∞,∞)
    eventOut     SFRotation value_changed
}

```

Fig. 5. 10: Código del nodo *OrientatorInterpolator*

Los dos campos principales son *key* y *keyvalue*. El campo *key* indica los estados por los que va pasar la variable y el campo *keyvalue* indica el valor que va a alcanzar en cada uno de esos estados.

5.3.4.8 Nodo *Viewpoint*

En VRML existe la posibilidad de definir puntos de vista diversos. De esta forma, se pueden mostrar al usuario aquellas partes que son más importantes del mundo que se ha definido. El usuario puede navegar con total libertad por el mundo creado. Pero si se define una serie de puntos de interés, puede ir directamente a estos.

La forma de definir estos puntos de vista o puntos de interés es mediante el nodo *Viewpoint*. Para escoger cualquier punto de vista, se utiliza el comando que el visualizador normalmente tiene disponible. El primer punto de vista que se define en el código, pasa a ser el punto de vista de arranque del mundo.

El primer nodo *Viewpoint* define un punto de vista idéntico al que ya se establece por defecto al entrar al visualizador. Esto sirve para poder volver a la posición inicial sin tener que volver a cargar el entorno. Los valores por defecto son pues, situar el observador sobre el eje Z positivo a 10 unidades del origen y mirando en dirección del eje Z negativo. En la Fig. 5.11 se indica el código para la programación del nodo *Viewpoint*.

```

Viewpoint {
    eventIn      SFBool      set_bin
    exposedField SFFloat      fieldOfView 0.785398 # (0, $\pi$ )
    exposedField SFBool      jump      TRUE
    exposedField SFRotation orientation 0 0 1 0 #[-1,1],(- $\infty$ , $\infty$ )
    exposedField SFVec3f    position    0 0 1 0 #(- $\infty$ , $\infty$ )
    field        SFString    description ""
    eventOut     SFTime      bindTime
    eventOut     SFBool      isBound
}

```

Fig. 5. 11: Código del nodo genérico *Viewpoint*

5.3.4.9 Nodos *Event* y *Route*

Con VRML es posible crear escenas dinámicas, objetos que cambian de forma, color o en movimiento. Para cambiar el estado de un objeto es preciso indicar cómo y cuándo éste debe efectuar el cambio. Esto requiere un modelo interno de ejecución, necesario para gobernar cómo cambian las cosas y en qué orden lo hacen. Esto se hace conectando los nodos entre sí, proporcionando “rutas” a través de las que puedan enviar y recibir “eventos”. Los eventos son mensajes que ligan los elementos de la escena y todo lo que se mueve o interactúa en VRML es debido a los eventos.

Casi todos los nodos poseen *eventIns* y *eventOuts* (eventos de entrada y de salida); de esta forma es como los nodos se comunican con el exterior. Los *eventIns* son

semejantes a receptores que escuchan los mensajes (eventos) del exterior y los toman para procesarlos. Los *eventOuts* son transmisores que envían los eventos producidos por el nodo al exterior. A continuación se describe el código tipo para ejecuta una ruta:

```
ROUTE NodoSalida.EventoSalida To NodoEntrada.EventoEntrada
```

5.3.4.10 Nodos *DEF* y *USE*

Cuando se quiere definir un objeto que se utiliza en repetidas ocasiones y se satura el código por el hecho de repetir un mismo bloque de código, entonces se puede utilizar *DEF* y *USE*. Estos dos comandos permiten dar un nombre a un cierto nodo que, posteriormente, se utilizará con la misma estructura y definición en diversos sitios.

5.4 Programación en VRML del movimiento del diferencial

En este apartado se muestra, mediante la programación en VRML, el montaje del conjunto mecánico diferencial y los regímenes de movimiento más significativos a los que puede someterse.

Antes de programar en VRML se debe hacer un boceto o tener al menos algunas ideas claras en cuanto a lo que se quiere mostrar y lo que se necesita mover, con el fin de crear un programa lo más sencillo y eficaz posible, ya que, un mismo programa puede realizarse de diferentes maneras, más o menos complicadas, y el resultado puede ser el mismo.

5.4.1 Identificación de grupos fijos

Es muy importante saber que al importar elementos individuales a VRML el programa lo sitúa en el punto O (0, 0, 0) del eje de coordenadas tridimensional. Si se

importan piezas de Solid Edge o cualquier otro programa que ya tenga sistemas de coordenadas, planos de referencia, etc., VRML lo situará igualmente que el programa de origen, provocando problemas si no coinciden los planos de referencia del programa de origen con los ejes locales del elemento. Por tanto, es trabajo imprescindible situar cada conjunto fijo en el centro de los planos de referencia de Solid Edge.

A continuación, se identifican los grupos solidarios independientes, sus componentes y una imagen importada desde Solid Edge a VRML con el visualizador *Cortona*:

1. **Grupo Casquete-Corona.** En la Fig. 5.12 se muestra el conjunto formado por la corona hipoidal de 31 dientes, los tornillos que unen el casquete a la corona, los rodamientos que unirían este conjunto a un bastidor portante no representado en este proyecto, y el eje que soporta a los satélites.

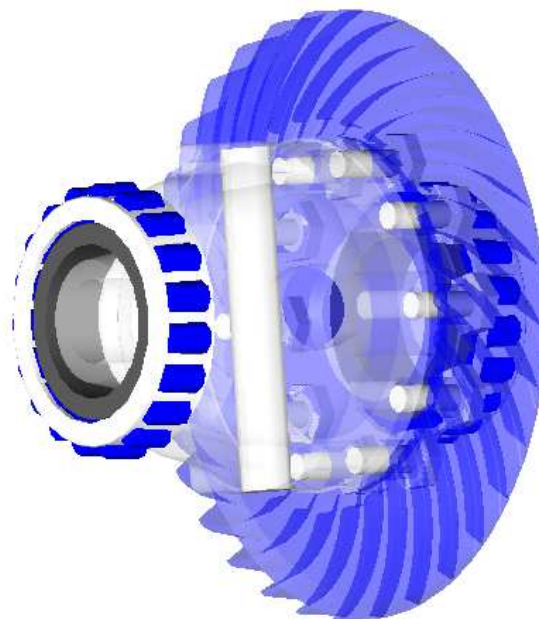


Fig. 5. 12: Grupo Casquete-Corona importado a VRML

2. **Grupo Eje-Planetario.** Agrupa la unión entre la rueda, el neumático, palier y el planetario de 16 dientes. Este grupo representa el planetario y su unión esquemática con la rueda (Fig.5.13).

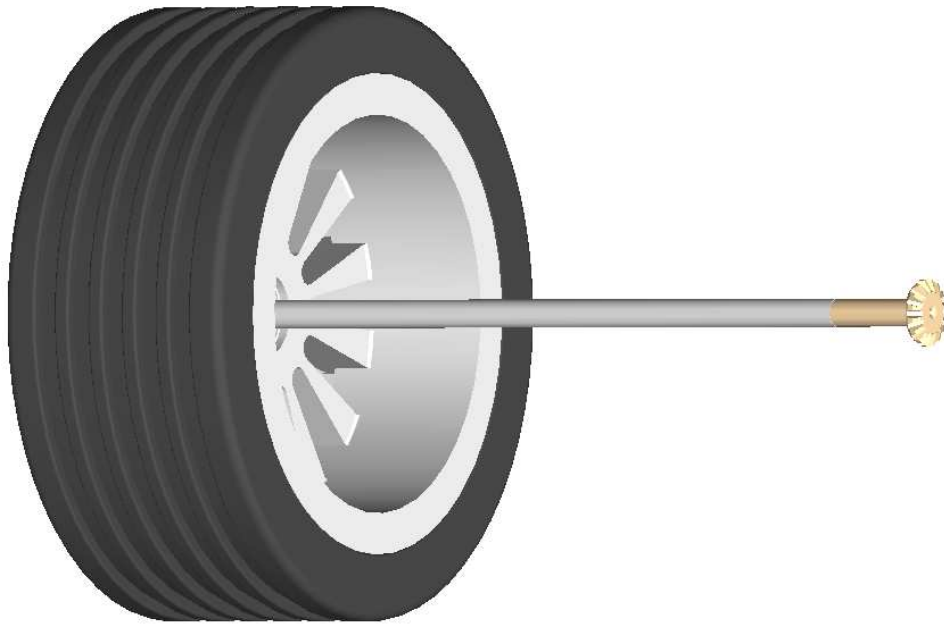


Fig. 5. 13: Grupo eje-planetario importado a VRML

3. **Piñón de ataque.** Es la pieza (Fig. 5.14) que inicia el evento del movimiento y por tanto se debe definir su movimiento de manera independiente a las demás piezas.

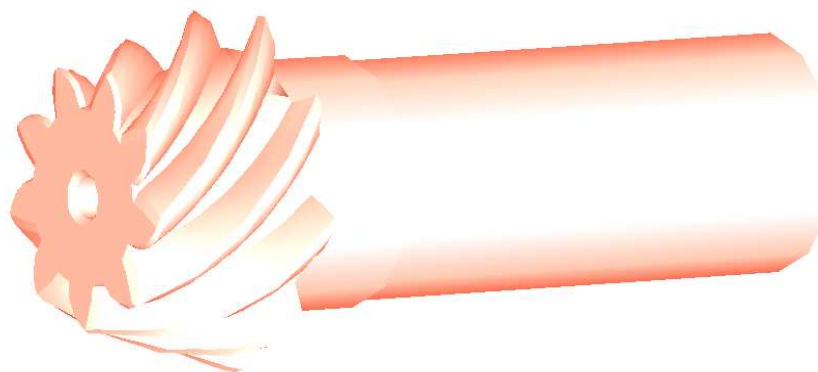


Fig. 5. 14: Piñón de ataque importado a VRML

4. **Grupo Satélite.** Van introducidos en el eje del Grupo Casquete-Corona y transmiten el movimiento de compensación entre planetarios (Fig. 5.15).

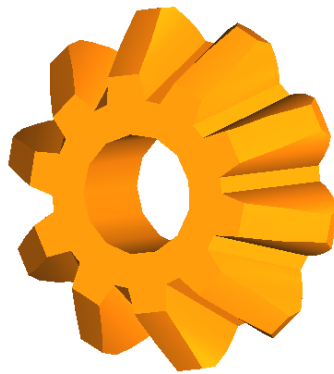


Fig. 5. 15: Satélite importado a VRML

5.4.2 Ensamblaje del conjunto en VRML

Una vez reconocidos los grupos independientes, el siguiente paso es ensamblarlos en el espacio de VRML.

5.4.2.1 Inserción al entorno VRML del Grupo Casquete-Corona

Se define el grupo *Casquete-Corona* (Fig. 5.16) con el nodo DEF, y dentro de él se define el *SATELITE1* y *SATELITE2* porque su movimiento es solidario al del Casquete-Corona, independientemente de cómo se muevan estos. Para incluirlos se ha utilizado el nodo *Children*, que da la posibilidad de indicar un conjunto de varios elementos a “transformar”. Después se utiliza el nodo *Inline*, que tiene como misión importar piezas guardadas con la extensión “.wrl” propia de VRML. Como el Grupo Casquete-Corona sitúa su centro en el centro de nuestro espacio virtual (0, 0, 0), no se le aplica ninguna transformación. Por el contrario los satélites deben ser colocados con el nodo *Transform*.

En la Fig. 5.16 se describe el código para integrar en el entorno VRML el Grupo Casquete-Corona. En la Fig. 5.17 se ve el resultado de la integración, con el acoplamiento de los satélites dentro del casquete.

5.4.2.2 Inserción al entorno VRML de los Grupos Eje-Planetario

En este caso se introducen dos grupos exactamente iguales pero que se sitúan en lugares diferentes. Para ello, se definen ambos ejes con un nombre diferente (Fig. 5.18) para posteriormente poder hacer referencia a ellos de manera independiente, y se usa el nodo *Inline* para importarlos y el nodo *Transform* para colocarlos en el lugar adecuado. En el Grupo *Ejederecho* es necesario rotar el conjunto para obtener la posición simétrica respecto al diferencial (Fig.5.19).

```

DEF Ejeizquierdo Transform {
  translation      0 0 0.01581
  children         [ Inline {
    url "EjePlanetario.wrl"
  }
  ]
}

DEF Ejederecho Transform {
  translation      0 0 0.04554
  rotation 0 1 0 3.14
  children         [ Inline {
    url "EjePlanetario.wrl"
  }
  ]
}

```

Fig. 5. 18: Código de integración del Grupo Eje-Planetario en VRML

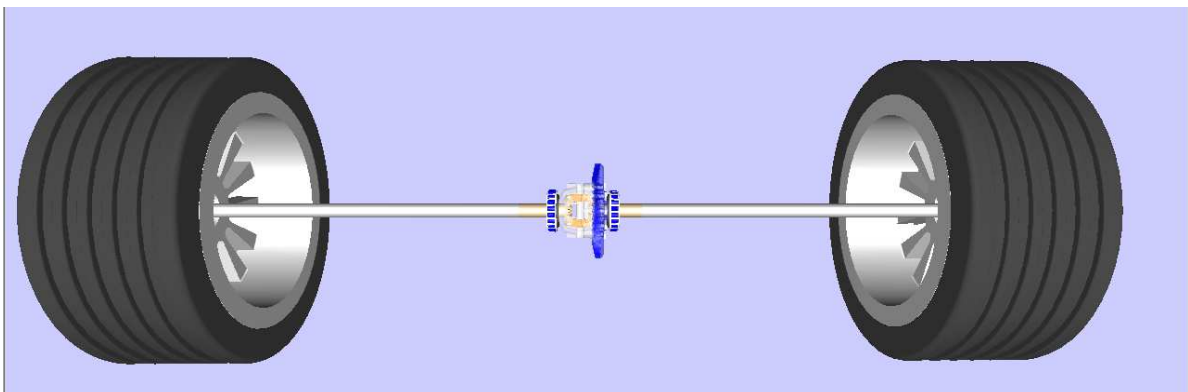


Fig. 5. 19: Grupo Eje-planetario ensamblado en VRML

5.4.2.3 Inserción al entorno VRML del piñón de ataque

Por último se introduce el piñón de ataque que engrana con la corona y es la pieza que transmite movimiento al diferencial. Debido a la posición que tenía en Solid Edge, ha sido necesario trasladar la pieza y rotarla 3,14 radianes respecto al eje Y con el fin de obtener la posición adecuada (Cuadro 5.20). Finalmente se obtiene el piñón posicionado (Fig. 5.21).

```
DEF PINON_DE_ATAQUE Transform{  
  translation      0.051 0.03622 0.02662  
  rotation 0 1 0 3.14  
  children        [ Inline {  
                    url "pinonejex.wrl"  
                  }  
                ]  
}
```

Fig. 5. 20: Integración del piñón de ataque en el entorno VRML

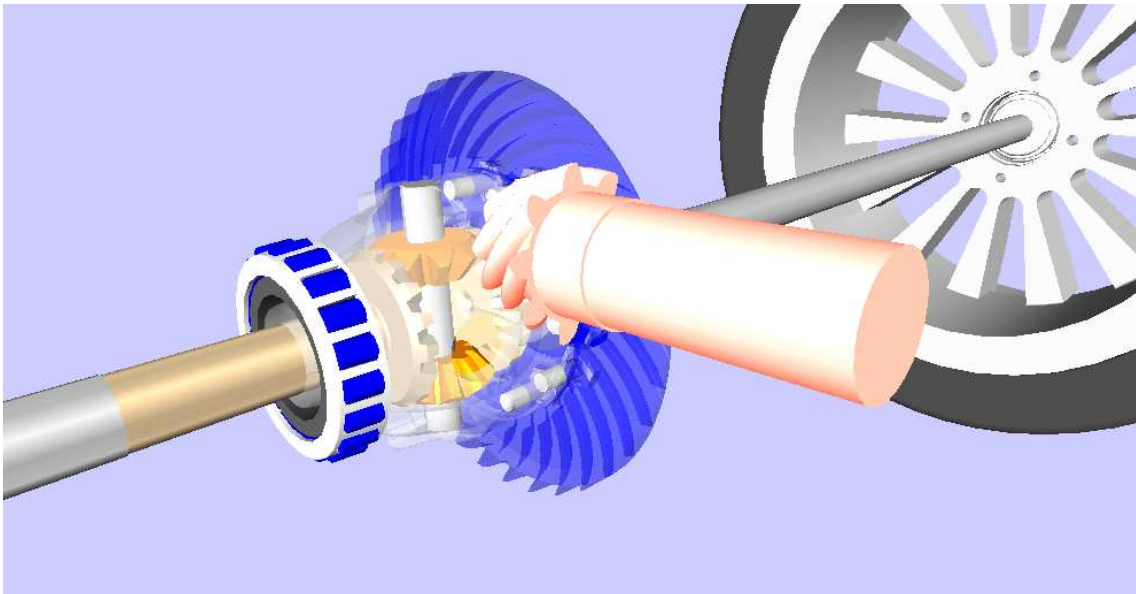


Fig. 5. 21: Piñón de ataque ensamblado en VRML

5.4.3 Animación del diferencial mecánico en VRML

La simulación del diferencial muestra tres posibles regímenes de movimiento:

- Diferencial libre en trazada recta.
- Diferencial libre en trazada curva.
- Diferencial libre bloqueado.

Para generar el movimiento se emplean rutas que envían y reciben eventos. Normalmente las rutas comienzan cuando se activa un sensor que ordene el movimiento a los conjuntos independientes, de manera coordinada, para dar sensación de dependencia dinámica real. Esta coordinación se puede controlar con los interpoladores que rigen el movimiento y “transforman” el estado de los grupos de piezas, trasladándolos o rotándolos en función del tiempo.

5.4.3.1 Programación de los sensores táctiles

Para iniciar el movimiento se necesitan tres botones virtuales que representen el régimen que van a activar. En la Fig. 5.22 se describe la programación del botón virtual que activa el diferencial en trazada recta. Los demás activadores se programan de manera análoga.

```
DEF ActivadorRecta Transform {  
  
  translation0 0.15 0  
  
  children [  
    Shape {  
      geometry Box { size 0.05 0.05 0.05}  
      appearance Appearance  
      {material Material {diffuseColor 0 0 1}  
      texture ImageTexture {url "recta.gif"}  
      }  
    }  
  ]  
  DEF SensorRecta TouchSensor {}  
}
```

Fig. 5. 22: Código para la inserción de sensores táctiles

En el código de texto mostrado (Fig. 5.22) se genera un cubo con una imagen que representa el régimen asociado y posteriormente es asignado como *Touchsensor* (Fig. 5.23), de esta manera cada vez que se señale con el puntero este área, dicho puntero se transforma en una mano para indicar al usuario que está preparado para activar el sensor.

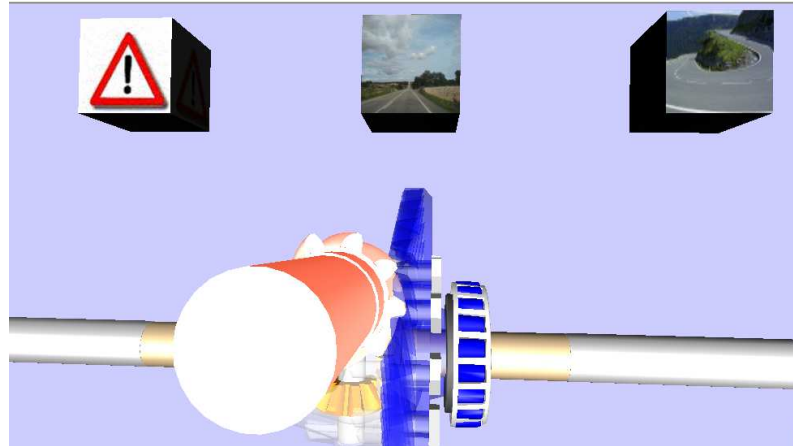


Fig. 5. 23: Sensores táctiles de activación de regímenes.

5.4.3.2 Programación de los cronómetros de movimiento

Para generar movimiento, VRML cambia de posición las piezas en función de un intervalo de tiempo determinado. Para conseguirlo se necesita un cronómetro o *timer*. El nodo *Timesensor* es muy importante en la animación de VRML. Puede usarse para generar eventos regulares, proporcionar eventos cronometrados, o manejar los nodos interpoladores.

En la Fig. 5.24 se muestra la programación de los cronómetros de movimiento necesarios para cada régimen de funcionamiento. En primer lugar está el campo *cycleInterval* que indica el tiempo que el *timer* correrá antes de restablecerse. El campo *loop* especifica si el timer se ejecutará continuamente o si simplemente lo hará una vez. Si le indicamos *true*, se generará un evento cada *cycleInterval*. De otra manera sólo se generará uno, después del *cycleInterval*. Los campos *startTime* y *stopTime* poseen valores que especifican cuándo comienza a contar el cronómetro y cuándo se detiene. En el presente proyecto, el diferencial no se moverá continuamente y cada pieza tendrá 11 intervalos de tiempo, a los cuales se deben asignar la orientación y la posición adecuada a cada régimen.


```

DEF MotorDiferencialRecta TimeSensor {
loop FALSE
cycleInterval 11.0
startTime -1
stopTime -1
}

DEF MotorDiferencialBloqueo TimeSensor {
loop FALSE
cycleInterval 11.0
startTime -1
stopTime -1
}

DEF MotorDiferencialCurva TimeSensor {
loop FALSE
cycleInterval 11.0
startTime -1
stopTime -1
}

```

Fig. 5. 24: Código para la programación de los cronómetros de movimiento

5.4.3.3 Programación del movimiento del diferencial en recta

Para definir la orientación correcta de cada grupo de piezas independiente se deben aplicar las fórmulas de la cinemática del diferencial. En la figura 5.25 se indica la notación a seguir.

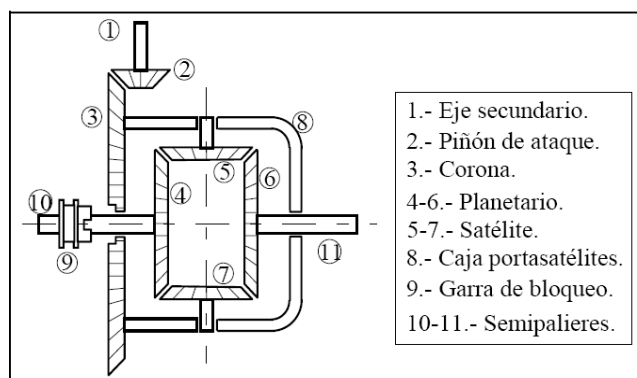


Fig. 5. 25: Notación

En la tabla 5.4 se muestra el valor de los parámetros geométricos de cada elemento del diferencial ya que es un dato conocido.

Descripción	Parámetro	Radio (mm)
Radio del piñón de ataque	R_2	22
Radio de la corona	R_3	72,5
Radio de los planetarios	$R_4 = R_6$	29
Radio de los satélites	$R_5 = R_7$	18

Tabla 5. 4: Radios de las piezas del diferencial

Y en la tabla 5.5 se indican los parámetros cinemáticos de cada uno de los elementos del diferencial.

Descripción	Parámetro
Velocidad angular del piñón de ataque	ω_2
Velocidad angular de la corona	ω_3
Velocidad angular del planetario izquierdo	ω_4
Velocidad angular del satélite superior	ω_5
Velocidad angular del planetario derecho	ω_6
Velocidad angular del satélite inferior	ω_7

Tabla 5. 5: Notación de los parámetros cinemáticos

A continuación se exponen las fórmulas básicas a partir de las cuales se obtienen las velocidades angulares de cada elemento del diferencial en cada uno de los casos particulares.

Según la Ley Fundamental de Engrane [18] el piñón de ataque transmite el movimiento a la corona cumpliendo la igualdad descrita en la expresión 1 (ver notación en tablas 5.4 y 5.5):

$$\omega_2 \cdot R_2 = \omega_3 \cdot R_3 \quad (1)$$

Despejando la velocidad angular de la corona ω_3 se obtiene la expresión 2:

$$\boxed{\omega_3 = \frac{\omega_2 \cdot R_2}{R_3}} \quad (2)$$

Por otro lado, como la corona es solidaria al casquete y a su vez este contiene a los satélites, el movimiento del planetario ω_4 depende del movimiento de la corona ω_3 y del satélite ω_5 cumpliendo la expresión 3:

$$\omega_3 \cdot R_4 - \omega_5 \cdot R_5 = \omega_4 \cdot R_4 \quad (3)$$

Sacando factor común R_4 se obtiene la expresión 4:

$$\boxed{(\omega_3 - \omega_4) \cdot R_4 = \omega_5 \cdot R_5} \quad (4)$$

De forma análoga se opera con el satélite y el planetario opuesto (ver notación en tablas 5.4 y 5.5) resultando las expresiones 5 y 6:

$$\omega_3 \cdot R_6 - \omega_5 \cdot R_5 = \omega_6 \cdot R_6 \quad (5)$$

$$\boxed{(\omega_3 - \omega_6) \cdot R_6 = \omega_5 \cdot R_5} \quad (6)$$

Dividiendo la expresión 4 entre la expresión 6 se obtiene una relación de velocidades angulares (expresión 7):

$$\frac{(\omega_3 - \omega_4)}{(\omega_3 - \omega_6)} = -1 \quad (7)$$

Y despejando la velocidad angular de la corona ω_3 resulta que es igual a la semisuma de los planetarios (expresión 8):

$$\boxed{\omega_3 = \frac{(\omega_4 + \omega_6)}{2}} \quad (8)$$

Un diferencial mecánico puede funcionar dentro de un amplio rango de velocidades, por tanto se debe establecer como condición de contorno una velocidad angular para la pieza que inicia el movimiento, es decir, para el piñón de ataque.

Por razones de diseño y para que se pueda apreciar el movimiento de las piezas en la simulación virtual, se establece que el piñón de ataque gira a π radianes por segundo. De la expresión 2 se obtiene la velocidad angular de la corona ω_3 en radianes por segundo:

$$\omega_3 = \frac{\omega_2 \cdot R_2}{R_3} = \frac{\pi \cdot 22}{72,5} = 0,953 \text{ rad/s}$$

Para el caso particular del diferencial en recta se sabe que los planetarios giran a la misma velocidad angular, a dicha velocidad se designa como ω :

$$\omega_4 = \omega_6 = \omega$$

De la expresión 8 y de la hipótesis anterior se obtiene:

$$\omega_3 = \frac{(\omega_4 + \omega_6)}{2} = \frac{(\omega + \omega)}{2} = \frac{2\omega}{2} = \omega$$

Se llega a que los planetarios giran a la misma velocidad angular que la corona:

$$\omega_3 = \omega_4 = \omega_6 = 0,953 \text{ rad/s}$$

Como el diferencial en recta se comporta como un eje solidario, los satélites no se mueven, es decir ω_5 y ω_7 son nulos:

$$\omega_5 = \omega_7 = 0 \text{ rad/s}$$

En la Fig. 5.26 se muestra el código de programación en VRML donde se indica la posición de los elementos en cuaternios para cada uno de los 11 intervalos o segundos establecidos en los que el diferencial mecánico está en movimiento.

En aquellos elementos virtuales diseñados en Solid Edge que han sido importados con una posición y orientación que no coincide con la que ocupan finalmente en VRML ha sido necesario utilizar un software de apoyo [19] para poder calcular los cuaternios.

```

#a 0,953 rad/seg en z
DEF GiroCasqueteCoronaRecta OrientationInterpolator {
  Key [0, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1]
  keyValue [0 0 1 0,0 0 1 0.953,0 0 1 1.906,0 0 1 2.859,0 0 1 3.812,0 0 1 4.765,0 0 1 5.718,0 0 1 6.671,0
0 1 7.624,0 0 1 8.577,0 0 1 9.53]
}

#a 0,953 rad/seg en z
DEF GiroEjeDerechoRecta OrientationInterpolator {
  Key [0, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1]
  keyValue [0 1 0 3.14,-0.4587 0.8886 0.0004 3.1402,-0.8152 0.5792 0.0006 3.1407,-0.99 0.1408 0.0008
3.1414,-0.9443 -0.329 0.0008 3.1421,-0.6883 -0.7255 0.0005 3.1427,-0.2788 -0.9603 0.0002
3.1431,0.1827 -0.9813 -0.0002 3.1432,0.6213 -0.7836 -0.0005 3.1428,0.9115 -0.4113 -0.0007
3.1422,0.9986 0.0526 -0.0008 3.1415]
}

#a 0,953 rad/seg en z
DEF GiroEjeIzquierdoRecta OrientationInterpolator {
  Key [0, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1]
  keyValue [0 0 1 0,0 0 1 0.953,0 0 1 1.906,0 0 1 2.859,0 0 1 3.812,0 0 1 4.765,0 0 1 5.718,0 0 1 6.671,0
0 1 7.624,0 0 1 8.577,0 0 1 9.53]
}

#a -3.14 radianes por segundo en z
DEF GiroPinondeataque OrientationInterpolator {
  Key [0, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1]
  keyValue [0 1 0 3.14,-0.0008 0.0008 1 3.1416,0 -1 0.0016 3.1432,0.0008 -0.0024 -1 3.1416,0 1 -
0.0032 3.14,-0.0008 0.004 1 3.1415,0 -1 0.0048 3.1432, 0.0008 -0.0056 -1 3.1416,0 1 -0.0064 3.14,-
0.0008 0.0072 1 3.1416,0 -1 0.008 3.1432,0.0008 -0.0088 -1 3.1416]
}

```

Fig. 5. 26: Posición de cada grupo en función de los intervalos de tiempo en recta

La posición que deben adoptar cada una de las piezas en cada intervalo de tiempo se deben conectar a través del nodo *route* (rutas). Este nodo funciona como conexiones que encauzan los eventos de un *eventOut* hacia un *eventIn*. Se pueden conectar varios *eventIns* a un *eventOut* para hacer un efecto causado por muchos eventos distintos.

En el Fig. 5.27 se describe el código que define las rutas para coordinar el movimiento del diferencial en recta. La primera ruta enlaza el nodo *Touchsensor* denominado “SensorRecta” con el nodo *Timesensor* “MotorDiferencialRecta” para que comience a funcionar el cronómetro. Una vez que este comienza a funcionar, en las siguientes rutas llama a cada una de las piezas para que adopten la posición en el espacio en función del intervalo correspondiente definido mediante el *OrientatorInterpolator*.

```

# rutas diferencial en recta

ROUTE SensorRecta.touchTime TO MotorDiferencialRecta.startTime

ROUTE MotorDiferencialRecta.fraction_changed TO GiroPinondeataque.set_fraction
ROUTE GiroPinondeataque.value_changed TO PINON_DE_ATAQUE.rotation

ROUTE MotorDiferencialRecta.fraction_changed TO GiroCasqueteCoronaRecta.set_fraction
ROUTE GiroCasqueteCoronaRecta.value_changed TO CASQUETE_CORONA.rotation

ROUTE MotorDiferencialRecta.fraction_changed TO GiroEjeDerechoRecta.set_fraction
ROUTE GiroEjeDerechoRecta.value_changed TO Ejederecho.rotation

ROUTE MotorDiferencialRecta.fraction_changed TO GiroEjeIzquierdoRecta.set_fraction
ROUTE GiroEjeIzquierdoRecta.value_changed TO Ejeizquierdo.rotation

```

Fig. 5. 27: Rutas caso diferencial en recta

5.4.3.4 Definición de movimiento del diferencial en bloqueo

En este caso se somete al diferencial mecánico libre a una situación de bloqueo, provocando que todo el movimiento se “escape” por uno de sus ejes, y las causas pueden ser diversas: pérdida de adherencia de una rueda, atasco en uno de los ejes, etc.

Al igual que el caso del diferencial en recta se supone que el piñón de ataque gira a π radianes por segundo. De la expresión 2 se obtiene la velocidad angular de la corona ω_3 :

$$\omega_3 = \frac{\omega_2 \cdot R_2}{R_3} = \frac{\pi \cdot 22}{72,5} = 0,953 \text{ rad/s}$$

Se sabe que uno de los planetarios no gira:

$$\omega_6 = 0$$

Y que los satélites giran a la misma velocidad angular:

$$\omega_5 = \omega_7$$

De la expresión 6 se despeja la velocidad angular de uno de los satélites ω_5 :

$$\omega_5 = \omega_7 = \frac{(\omega_6 - \omega_3) \cdot R_6}{R_5} = \frac{-0,953 \cdot 29}{18} = -1,53 \text{ rad/s}$$

Y la velocidad angular del planetario ω_4 se obtiene de la expresión 7:

$$\frac{(\omega_3 - \omega_4)}{(\omega_3 - \omega_6)} = -1$$

La velocidad angular del planetario derecho ω_6 es nula, entonces:

$$\frac{(\omega_3 - \omega_4)}{\omega_3} = -1$$

Despejando ω_4 resulta:

$$\omega_4 = 2\omega_3$$

Sustituyendo los parámetros, se obtiene que el planetario izquierdo gira a 1,906 rad/s:

$$\omega_4 = 2 \cdot 0,953 = 1,906 \text{ rad/s}$$

En la Fig. 5.28 y Fig. 5.29 se definen los giros que deben adoptar las piezas en este régimen.

```
#a 0,953 rad/seg en z
DEF GiroCasqueteCoronaRecta OrientationInterpolator {
  Key [0, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1]
  keyValue [0 0 1 0,0 0 1 0.953,0 0 1 1.906,0 0 1 2.859,0 0 1 3.812,0 0 1 4.765,0 0 1 5.718,0 0 1 6.671,0
0 1 7.624,0 0 1 8.577,0 0 1 9.53]
}

#a -3.14 radianes por segundo
DEF GiroPinondeataque OrientationInterpolator {
  Key [0, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1]
  keyValue [0 1 0 3.14,-0.0008 0.0008 1 3.1416,0 -1 0.0016 3.1432,0.0008 -0.0024 -1 3.1416,0 1 -
0.0032 3.14,-0.0008 0.004 1 3.1415,0 -1 0.0048 3.1432, 0.0008 -0.0056 -1 3.1416,0 1 -0.0064 3.14,-
0.0008 0.0072 1 3.1416,0 -1 0.008 3.1432,0.0008 -0.0088 -1 3.1416]
}
```

Fig. 5. 28: Posición de cada grupo en función de los intervalos de tiempo en bloqueo

```

DEF GiroSatelite1 OrientationInterpolator {
  Key [0, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1]
  keyValue [0 -1 0 0,0 -1 0 1.57,0 -1 0 3.14,0 -1 0 4.71,0 -1 0 6.28,0 -1 0 7.85,0 -1 0 9.42,0 -1 0 10.99,0 -
1 0 12.56,0 -1 0 14.13,0 -1 0 15.7]
}

DEF GiroEjeDerechoBloqueo OrientationInterpolator {
  Key [0, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1]
  keyValue [0 1 0 3.14,-0.8152 0.5792 0.0006 3.1407,-0.9443 -0.329 0.0008 3.1421,-0.2788 -0.9603
0.0002 3.1431,0.6213 -0.7836 -0.0005 3.1428,0.9986 0.0526 -0.0008 3.1415,0.5356 0.8445 -0.0004
3.1402,-0.3782 0.9257 0.0003 3.1401,-0.9737 0.228 0.0008 3.1412,-0.7498 -0.6617 0.0006 3.1426,0.105
-0.9945 -0.0001 3.1432]
}

DEF GiroSatelite2 OrientationInterpolator {
  Key [0, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1]
  keyValue [0 1 0 0,0 1 0 1.57,0 1 0 3.14,0 1 0 4.71,0 1 0 6.28,0 1 0 7.85,0 1 0 9.42,0 1 0 10.99,0 1 0
12.56,0 1 0 14.13,0 1 0 15.7]
}

```

Fig. 5. 29: Posición de cada grupo en función de los intervalos de tiempo en bloqueo

Análogo al régimen del diferencial en recta se deben definir las rutas para enlazar los eventos (Fig. 5.30).

```

ROUTE SensorBloqueo.touchTime TO MotorDiferencialBloqueo.startTime

ROUTE MotorDiferencialBloqueo.fraction_changed TO GiroPinondeataque.set_fraction
ROUTE GiroPinondeataque.value_changed TO PINON_DE_ATAQUE.rotation

ROUTE MotorDiferencialBloqueo.fraction_changed TO GiroCasqueteCoronaRecta.set_fraction
ROUTE GiroCasqueteCoronaRecta.value_changed TO CASQUETE_CORONA.rotation

ROUTE MotorDiferencialBloqueo.fraction_changed TO GiroEjeDerechoBloqueo.set_fraction
ROUTE GiroEjeDerechoBloqueo.value_changed TO Ejederecho.rotation

ROUTE MotorDiferencialBloqueo.fraction_changed TO GiroSatelite1.set_fraction
ROUTE GiroSatelite1.value_changed TO SATELITE1.rotation

ROUTE MotorDiferencialBloqueo.fraction_changed TO GiroSatelite2.set_fraction
ROUTE GiroSatelite2.value_changed TO SATELITE2.rotation

```

Fig. 5. 30: Rutas de enlace del diferencial en bloqueo

5.4.3.5 Definición de movimiento del diferencial en curva

Por último se describe la simulación del régimen de trabajo más habitual para un diferencial, la trazada de una curva. En este caso, para obtener la velocidad angular de los elementos, no basta con los datos de los que se dispone inicialmente ya que depende de otros factores como el radio de giro de la curva. En consecuencia, se fijan dos velocidades angulares para los planetarios y, en función a estas, se calculan las velocidades de los satélites.

Se toma ω_4 y ω_6 tal que cumplan la expresión 8:

$$\omega_4 = 0,600 \text{ rad/s}$$

$$\omega_6 = 1,305 \text{ rad/s}$$

Entonces las velocidades angulares de los satélites son:

$$\omega_5 = \omega_7 = \frac{(\omega_3 - \omega_4) \cdot R_4}{R_5} = \frac{(0,953 - 0,6) \cdot 29}{18} = 0,5684 \text{ rad/s}$$

En esta simulación se mueven todos los grupos de piezas y elementos del diferencial. Para simular este caso, el piñón de ataque transmite movimiento al Grupo Eje-Corona que a su vez traslada a los satélites que balancean el movimiento relativo entre los Grupos Eje-planetario. Uno de los Grupos Eje-Planetario se mueve más rápido que el otro debido a que cada rueda traza un arco diferente (Fig. 5.31 y Fig. 5.32).

```

DEF GiroEjeIzquierdoCurva OrientationInterpolator {
  Key [0, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1]
  keyValue [0 0 1 0,0 0 1 1.306,0 0 1 2.611,0 0 1 3.917,0 0 1 5.222,0 0 1 6.528,0 0 1 7.834,0 0 1 9.139,0
0 1 10.445,0 0 1 11.750,0 0 1 13.056]
}

DEF GiroEjeDerechoCurva OrientationInterpolator {
  Key [0, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1]
  keyValue [0 1 0 3.14,-0.2955 0.9553 0.0002 3.1401,-0.5646 0.8253 0.0004 3.1403,-0.7833 0.6216
0.0006 3.1406,-0.932 0.3624 0.0007 3.141,-0.9975 0.0707 0.0008 3.1415,-0.9738 -0.2272 0.0008 3.142,-
0.8632 -0.5048 0.0007 3.1424,-0.6755 -0.7374 0.0005 3.1428,-0.4274 -0.9041 0.0003 3.143,-0.1411 -0.99
0.0001 3.1432]
}

```

Fig. 5. 31: Posición de cada grupo en función de los intervalos de tiempo en curva

```

DEF GiroSatelite1curva OrientationInterpolator {
  Key [0, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1]
  keyValue [0 -1 0 0,0 -1 0 -0.568,0 -1 0 -1.137,0 -1 0 -1.705,0 -1 0 -2.274,0 -1 0 -2.842,0 -1 0 -3.410,0 -
1 0 -3.979,0 -1 0 -4.547,0 -1 0 -5.116,0 -1 0 -5.684]
}

DEF GiroSatelite2curva OrientationInterpolator {
  Key [0, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1]
  keyValue [0 1 0 0,0 1 0 -0.568,0 1 0 -1.137,0 1 0 -1.705,0 1 0 -2.274,0 1 0 -2.842,0 1 0 -3.410,0 1 0 -
3.979,0 1 0 -4.547,0 1 0 -5.116,0 1 0 -5.684]
}

```

Fig. 5. 32: Posición de cada grupo en función de los intervalos de tiempo en curva

Finalmente se establecen las rutas entre eventos (Fig. 5.33) para el caso del diferencial en curva.

```

ROUTE SensorCurva.touchTime TO MotorDiferencialCurva.startTime

ROUTE MotorDiferencialCurva.fraction_changed TO GiroPinondeataque.set_fraction

ROUTE GiroPinondeataque.value_changed TO PINON_DE_ATAQUE.rotation

ROUTE MotorDiferencialCurva.fraction_changed TO GiroCasqueteCoronaRecta.set_fraction
ROUTE GiroCasqueteCoronaRecta.value_changed TO CASQUETE_CORONA.rotation

ROUTE MotorDiferencialCurva.fraction_changed TO GiroEjeDerechoCurva.set_fraction
ROUTE GiroEjeDerechoCurva.value_changed TO Ejederecho.rotation

ROUTE MotorDiferencialCurva.fraction_changed TO GiroEjeIzquierdoCurva.set_fraction
ROUTE GiroEjeIzquierdoCurva.value_changed TO Ejeizquierdo.rotation

ROUTE MotorDiferencialCurva.fraction_changed TO GiroSatelite1curva.set_fraction
ROUTE GiroSatelite1curva.value_changed TO SATELITE1.rotation

ROUTE MotorDiferencialCurva.fraction_changed TO GiroSatelite2curva.set_fraction
ROUTE GiroSatelite2curva.value_changed TO SATELITE2.rotation

```

Fig. 5. 33: Rutas de enlace del diferencial en curva

5.5 Resultados obtenidos

Desde el punto de vista descriptivo, se ha conseguido obtener una reproducción aceptable del diferencial real mediante un sistema mecánico semi-esquemático que

representa con exactitud el diferencial y de forma menos detallada los demás elementos poco relevantes como los palieres, las ruedas o sus uniones (Fig. 5.34)

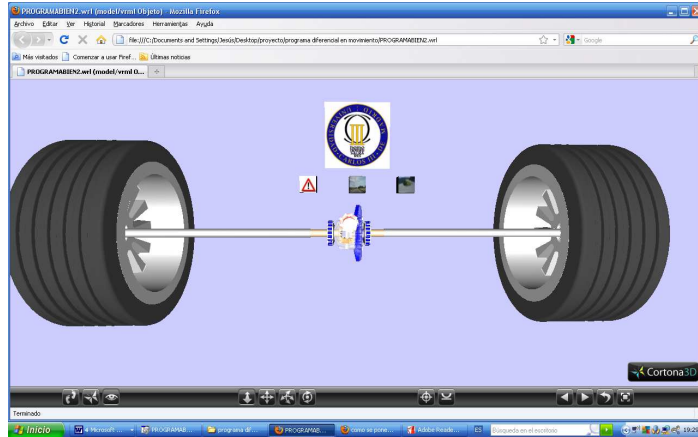


Fig. 5.34: Presentación del simulador

Se puede examinar el diferencial con los cursores del visualizador (Fig. 5.35).



Fig. 5.35: Cursores del visualizador Cortona

De esta manera es posible obtener una visión muy detallada de las piezas del diferencial en su posición estática, la situación dentro del casquete de cada una de las piezas (ya que el casquete queda transparente), la relación que se guardan entre ellas, también puede intuirse a que se debe la forma de cada uno de los elementos y la forma del dentado de las mismas (Fig. 5.36, Fig. 5.37, Fig. 5.38 y Fig. 5.39).

En cuanto al resultado dinámico del diferencial, se han obtenido tres simulaciones coherentes ya que están basadas en el análisis cinemático del diferencial. Y en las simulaciones no se detectan fallos que puedan generar errores de comprensión en el usuario.

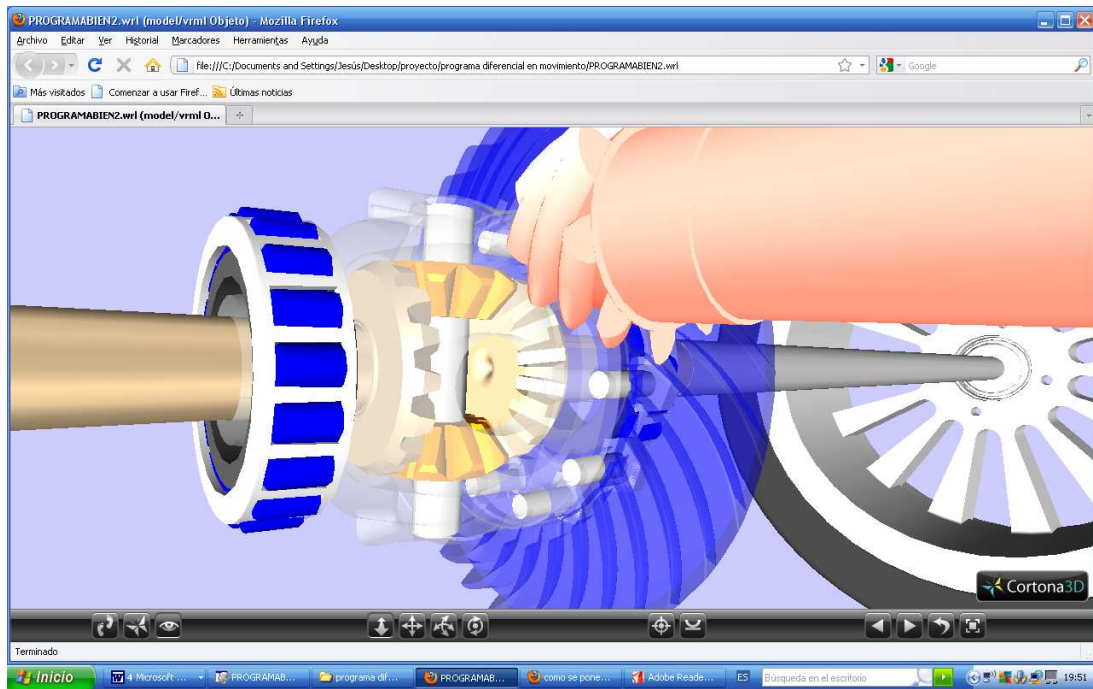


Fig. 5. 36: Visión de detalle1

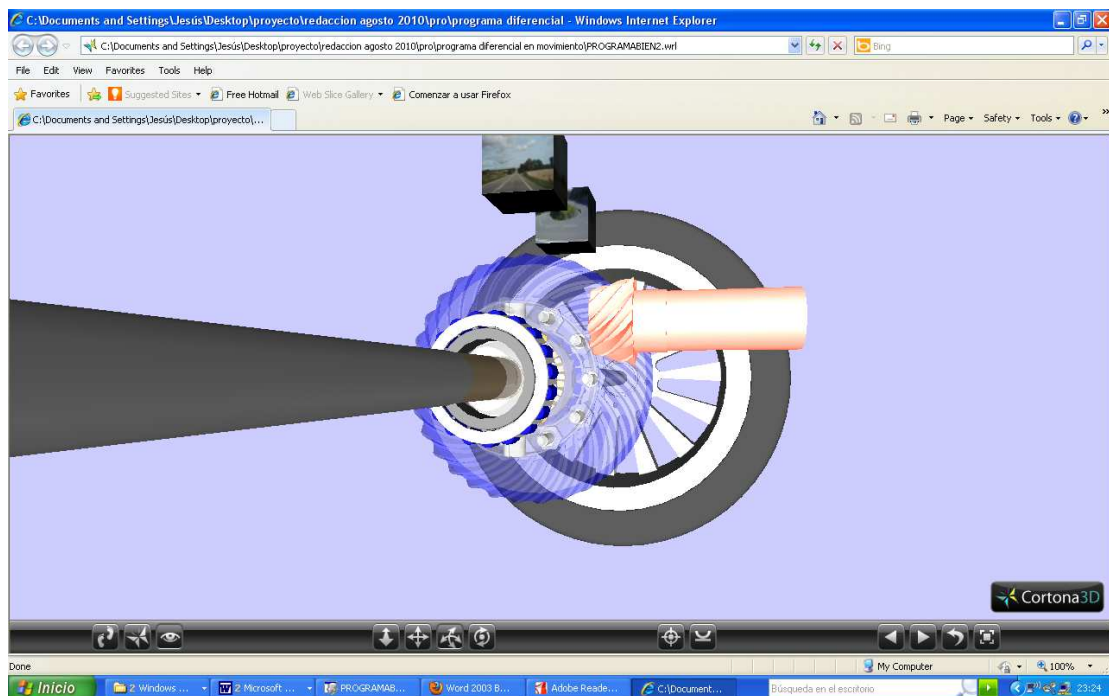


Fig. 5. 37: Visión de detalle 2

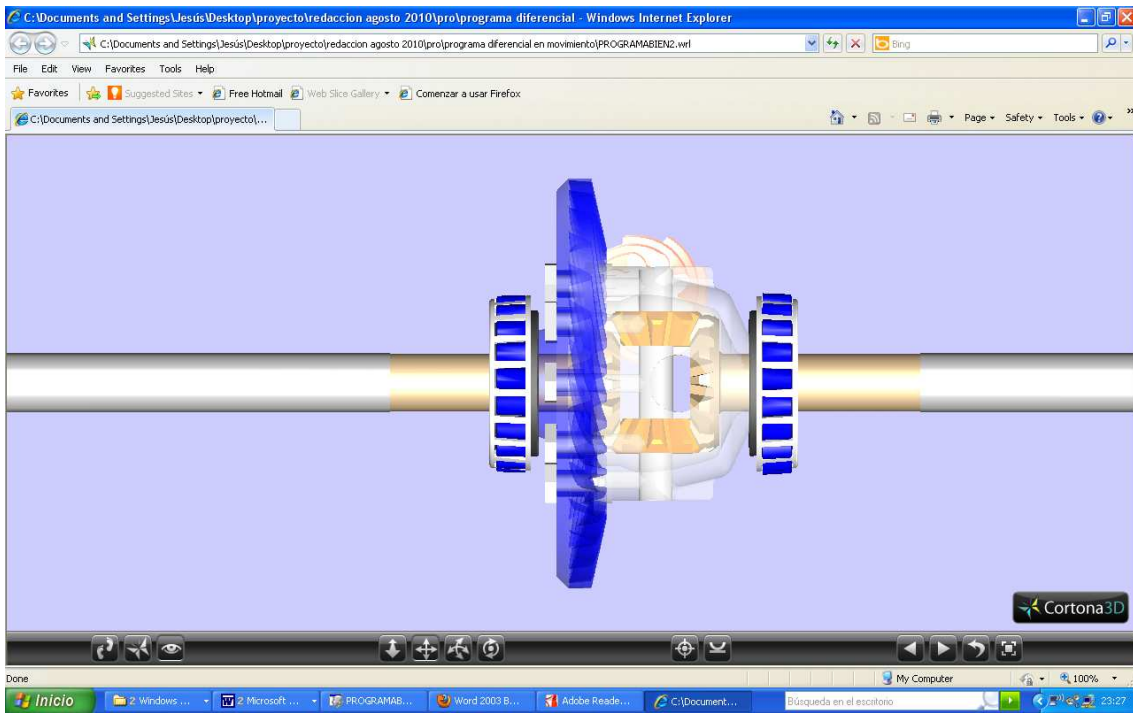


Fig. 5. 38: Visión de detalle 3

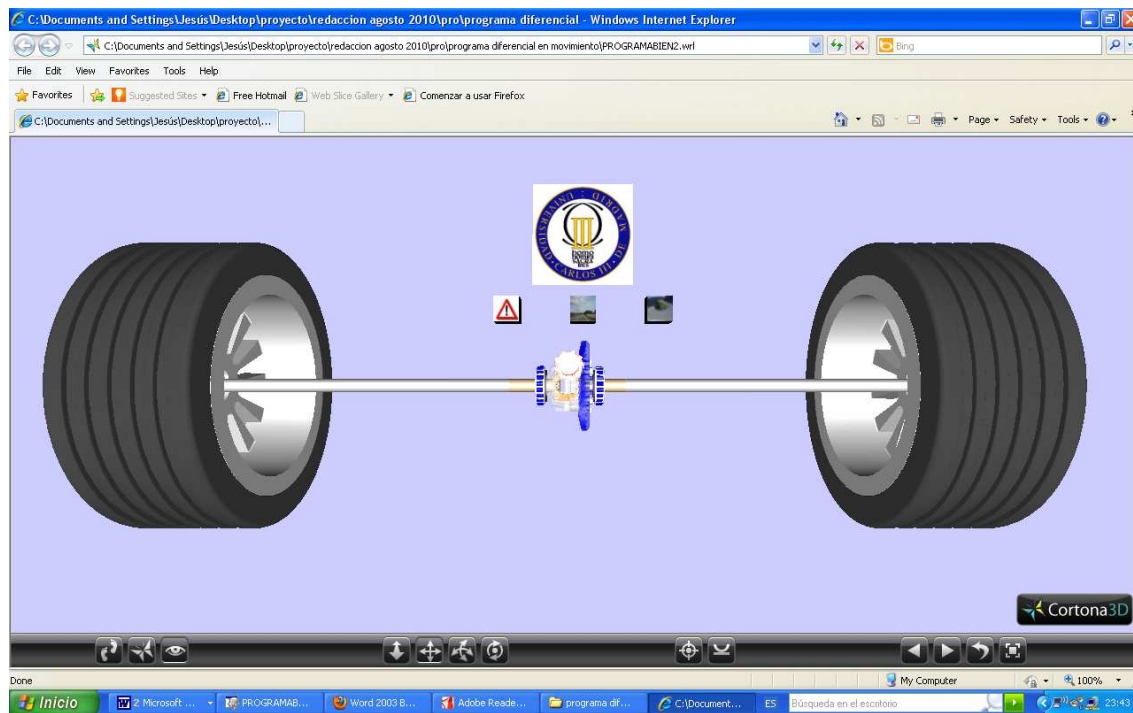


Fig. 5. 39: Visión de detalle 4

CAPÍTULO 6: CONCLUSIONES Y TRABAJOS FUTUROS

6.1 Conclusiones

Como conclusiones finales se puede afirmar que:

- Se ha conseguido crear un diferencial mediante simulación virtual a partir de un diferencial real, manteniendo una gran similitud tanto en las piezas individuales como en el conjunto ensamblado.
- Se ha logrado simular el comportamiento dinámico del diferencial en los tres casos más comunes y generar una aplicación en un entorno virtual con capacidad de movimiento, y muy intuitivo, que puede ser de gran ayuda tanto al alumno como al profesor, en la comprensión de sistemas mecánicos poco accesibles en el mundo real.
- Se ha demostrado que el entorno VRML proporciona una gran facilidad para la introducción de piezas en el mundo virtual, modeladas en 3D, mediante programas de diseño asistido por ordenador. Esto es una ventaja, ya que no son necesarias demasiadas herramientas para la creación de una simulación virtual. Lo único que se necesita es un programa de diseño 3D (*Solid Edge*), un editor de texto (*Word*, *WordPad*, etc.) y un visualizador o navegador de Internet.

- La programación en VRML es relativamente sencilla. Esto supone otra ventaja para la enseñanza y el estudio de cualquier materia a través de esta herramienta: no son necesarios unos estudios previos o una excesiva especialización en el mundo de la programación para poder crear estas simulaciones. Un proceso complicado en el aprendizaje de una herramienta puede suponer un rechazo por parte del estudiante.
- La simulación por ordenador en un entorno virtual facilita la comprensión acerca del funcionamiento de un mecanismo, ya que permite observar la misión y el lugar que ocupa cada una de las piezas que lo forman.

6.2 Trabajos futuros

Este proyecto puede mejorarse y completarse de formas diversas, a continuación se enumeran las líneas más inmediatas e interesantes:

- Mejora del conjunto mecánico diferencial añadiendo las piezas necesarias para desarrollar uno de los sistemas autoblocantes descritos, como el sistema de discos o de rampas, simular su movimiento y compararlos con el sistema diferencial libre primario. Para ello puede reutilizarse la mayoría de las piezas modeladas y el programa de simulación virtual en VRML.
- Otra línea de trabajo a seguir, es la diversificación del sistema mecánico, es decir, integrar el diferencial con otros sistemas tipo que formen parte de un vehículo de tracción mecánica a cuatro ruedas, como por ejemplo el modelado del sistema de frenado, integrando el diferencial como elemento transmisor de potencia, con el fin de ver cómo trabajan ambos sistemas a la vez en diferentes casos.
- No es desdeñable, la posibilidad de aprovechar las piezas modeladas para hacer análisis de elementos finitos, importando dichas piezas a programas especializados.

Todas estas propuestas de trabajo son interesantes y pueden ser provechosas tanto en el ámbito docente como en el de la investigación, ya que las piezas han sido reproducidas fielmente a un modelo real.

REFERENCIAS

[1] Adell Segura, Jordi. (1995). Educación en Internet. Disponible [Internet]: <<http://www.ucm.es/info/multidoc/multidoc/revista/cuad6-7/index6-7.htm>> [Consultado en Diciembre 2011].

[2] The size of the World Wide Web. De Kunder Internet Media. Disponible [Internet]: <<http://www.worldwidewebsize.com>> [Consultado en Diciembre 2011].

[3] CABERO, Julio (2006). «Bases pedagógicas del *e-learning*». *Revista de Universidad y Sociedad del Conocimiento (RUSC)* Vol. 3, n.º 1. Disponible [Internet]: <<http://www.uoc.edu/rusc/3/1/dt/esp/cabero.pdf>> [Consultado en Diciembre 2011]

[4] PARRA, J.C., GARCIA, R. y SANTELICES, I.: “Introducción Práctica a la Realidad Virtual”, ediciones universidad del Bio-Bio, Chile, 2001.

[5] SANCHEZ, V. y PIERART, P.: “Realidad Virtual... ¿de qué se trata?”, Universidad San Sebastián, 1997.

[6] TULIO PIOVAN, Marcelo (2004). «Trenes de engranajes, reductores planetarios y diferenciales». (Facultad Regional de Bahía Blanca): Cátedra de Elementos de Máquinas. Disponible [Internet]: <<http://www.frbb.utn.edu.ar/carreras/materias/elementosdemaquinas/cap09-01.pdf>> [Consultado en Diciembre 2011]

[7] Martell Pérez, J; Rodríguez de Torres, A. Ramón Moliner, P. (1976). “Elementos de Máquinas”. Universidad Nacional de Educación a Distancia.

[8] “Definición de diferencial mecánico” Gabinete Informático de Galicia S.L. Disponible [Internet]:
<[http://diccionario.motorgiga.com/diccionario/diferencial-definicion significado/gmx-niv15-con193856.htm](http://diccionario.motorgiga.com/diccionario/diferencial-definicion_significado/gmx-niv15-con193856.htm)> [Consultado en Diciembre 2011]

[9] Web MECANICAVirtual. Disponible [Internet]:
<<http://www.mecanicavirtual.org/diferencial-autoblocante.htm>> [Consultado en septiembre de 2011]

[10] “Diferenciales de deslizamiento limitado”. Autor: Juan Lasheras. Disponible [Internet]:
<<http://www.8000vueltas.com>> [Consultado en diciembre de 2011]

[11] Bruño Brunet P.: "Diseño gráfico y modelado geométrico". Mompín J. (Ed.): "Sistemas CAD/CAM/CAE. Diseño y fabricación por ordenador". Marcombo 1986. Disponible [Internet]:
<www.ilustrados.com/publicaciones> [Consultado en diciembre de 2011]

[12] Alexandre Samper: “La integración de las diversas técnicas de CAD/CAM/CAE”. Disponible [Internet]:
<<http://www.interempresas.net-La-integracion-de-las-diversas-tecnicas-de-CAD-CAM-CAE.html>> [Consultado en diciembre de 2011]

[13] Rafael Gutiérrez Olivar. “Solid Edge v.18 Guía de Referencia”. Unidad Docente de expresión gráfica en ingeniería. Universidad Politécnica de Madrid.

[14] Ernesto Jiménez Ruiz “Introducción al VRML” Disponible [Internet]:
<<http://www.jimenezruiz.es>> [Consultado en diciembre de 2011]

[15] Andrea L. Ames, David R. Nadeau, John L. Moreland. VRML 2.0 Sourcebook Ediciones John Wiley & Sons. 1996.

[16] Carlos L. González Cordero. Tutorial de VRML 2.0. 2009. Disponible [Internet]:

<<http://www.di.ujaen.es/~rsegura/igai/vrml/documentos/tema0.html>>

[17] Ilde Gámez, Martín Martín “VRML, curso de iniciación: aprenda a crear mundos virtuales en 3D para Internet”. Barcelona: Inforbooks, [2001]

[18] Adelardo Lamadrid Martínez, Antonio de Corral Saiz “Cinemática y dinámica de máquinas” Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingenieros Industriales, Sección de Publicaciones, 1992

[19] “XYZ to Angle-Axis Orientation Converter”. *Matthew Lewis*. ACCAD, Ohio State University, Columbus, OH, USA mlewis@accad.osu.edu.