

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

**Facultad de Ingeniería en Mecánica y Ciencias de la
Producción**

Desarrollo del modelo virtual de una máquina herramienta con objetivo de
ser conectada a IoT

PROYECTO INTEGRADOR

Previo la obtención del Título de:

Ingeniero en Mecatrónica

Presentado por:

Jhon Marco Navas Sánchez

GUAYAQUIL - ECUADOR

Año: 2020

DEDICATORIA

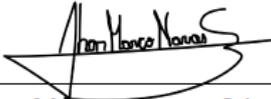
Dedico este trabajo principalmente a Dios, mi mejor maestro, que me ha permitido por gracia llegar a esta instancia para honrarle. A mis padres que siempre me han apoyado y brindado la mejor educación que estuvo a su alcance. A mi abuelita Angela que desde muy pequeño se preocupó por inculcarme valores fundamentales para ser una persona de bien. A mi tío Juan Carlos por servirme de inspiración profesional y demostrarme que surgir de la nada es posible. A mi tía Menni por siempre extender su mano de apoyo desinteresado e incondicional cuando la he necesitado. A toda mi familia por el cariño que me brindan y el aliento que me dan para alcanzar mis metas. A mis amigos que me han acompañado en esta aventura, llena de anécdotas que vivirán por siempre en nuestra memorias.

AGRADECIMIENTOS

Quiero primero agradecerle a Dios “Porque todas las cosas proceden de él, y existen por él y para él. ¡A él sea la gloria por siempre! Amén.” [1, rom 11:36]. Agradecerle de manera especial y sincera al Ph.D. Marcelo Fajardo por darme la oportunidad de trabajar en este proyecto. A mis tíos Marjorie Saltos y Joseph Rosales por sus palabras de apoyo y su importante contribución del computador necesario que me permitió desarrollar proyecto. A mis padres por su fe ciega en mis capacidades, mi más profundo agradecimiento.

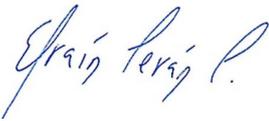
DECLARACIÓN EXPRESA

"Los derechos de titularidad y explotación, me corresponde conforme al reglamento de propiedad intelectual de la institución; Jhon Marco Navas Sánchez y doy mi consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"



Jhon Marco Navas Sánchez

EVALUADORES



Efraín Terán, M.Sc.

PROFESOR DE LA MATERIA

Marcelo Fajardo, Ph.D.

PROFESOR TUTOR

RESUMEN

La cuarta revolución industrial requiere que los equipos de los procesos trabajen en conjunto mediante la estrategia de sistemas cyberfísicos. El objetivo de este proyecto es contribuir en esta revolución, presentando una herramienta y una metodología que permita la creación de los sistemas cyberfísicos. Se utilizó un modelo V para el proceso de diseño, en el que se desarrolló el gemelo digital de una máquina herramienta 3PRS+XY+C. Se inició definiendo los requerimientos, que sirvieron de base para la creación de la estructura lógico-funcional. Esta estructura fue la guía para la creación de los modelos de comportamiento de cada elemento, el control en cascada y el diseño 3D de la máquina. Con el gemelo digital listo se realizó el proceso de simulación que permitió comprobar el correcto funcionamiento. Se realizó una prueba de posición y velocidad con arranque escalón y a una señal de segundo orden que evidenció el correcto control y el cumplimiento de la consigna. Las pruebas realizadas muestran que la estrategia utilizada permitió realizar el gemelo digital que conforma el sistema cyberfísico de la máquina, además debido al modo de diseño el modelo es altamente flexible y su estructura modular facilita los cambios. Es el momento correcto para aplicar esta estrategia en la industria ecuatoriana y así disminuir la brecha tecnológica y aumentar la competitividad internacional.

Palabras Clave: Industria 4.0, Sistemas cyberfísicos, Gemelo digital, Revolución industrial.

ABSTRACT

The fourth industrial revolution requires the process equipment to work together through the strategy of cyber-physical systems. The objective of this project is to contribute to this revolution by presenting a tool and a methodology that allows the creation of cyber-physical systems. A V-model was used for the design process, in which the digital twin of a 3PRS+XY+C machine tool was developed. It started by defining the requirements, which served as the basis for the creation of the logical-functional structure. This structure was the guide for the creation of the behavioral models of each element, the cascade control, and the 3D design of the machine. With the digital twin ready, the simulation process was carried out to verify the correct operation. A position and speed test were performed with step input and a second order signal that showed the correct control and compliance with the setpoint. The tests performed show that the strategy used allowed the realization of the digital twin that conforms the cyber-physical system of the machine, also due to the design mode the model is highly flexible, and its modular structure facilitates the changes. It is the right time to apply this strategy in the Ecuadorian industry and thus reduce the technological gap and increase international competitiveness.

Keywords: *Industry 4.0, Cyber-Physical Systems, Digital Twin, Industrial Revolution.*

ÍNDICE GENERAL

RESUMEN	I
ABSTRACT	II
ÍNDICE GENERAL	III
ABREVIATURAS	VI
SIMBOLOGÍA.....	VII
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS	X
CAPÍTULO 1	1
1. Introducción.....	1
1.1 Descripción del problema	3
1.2 Justificación del problema	4
1.3 Objetivos	5
1.3.1 Objetivo general.....	5
1.3.2 Objetivos específicos	5
1.4 Marco teórico	5
1.4.1 Máquina herramienta de configuración 3PRS+XY+C	6
1.4.2 Modelo cyberfísico	7
1.4.3 Internet Industrial de las cosas (IIoT)	8
1.4.4 Industria 4.0	9
1.4.5 Sistemas de manufactura inteligentes (IMS).....	10
1.4.6 Enovia.....	10
1.4.7 Catia	11
1.4.8 Modelica	11

CAPÍTULO 2	12
2. Metodología.....	12
2.1 Alternativas de solución	12
2.2 Proceso de diseño.....	15
2.2.1 Definición de requerimientos (R).....	16
2.2.2 Análisis funcional (F).....	18
2.2.3 Diseño lógico (L).....	20
2.2.4 Diseño físico (P)	23
2.3 Análisis Denavit-Hartenberg 3PRS.....	25
2.4 Análisis cinemático 3PRS.....	28
2.5 Análisis Dinámico 3PRS.....	31
2.6 Estructura de control	32
CAPÍTULO 3	33
3. Resultados Y Análisis.....	33
3.1 Mecanismo 3PRS+XY+C 3D.....	33
3.2 Modelo del sistema	37
3.2.1 Modelo del bloque Set_Point	37
3.2.2 Modelo del bloque Dinamic_Controller	38
3.2.3 Diagrama del actuador.....	39
3.2.4 Modelo del bloque 3PRS+XY+C_Machine.....	40
3.3 Simulación de funcionamiento.....	42
3.4 Análisis de costos.....	50
CAPÍTULO 4	53
4. Conclusiones Y Recomendaciones	53
4.1 Conclusiones.....	53
4.2 Recomendaciones.....	54

BIBLIOGRAFÍA	55
APÉNDICES	60

ABREVIATURAS

IloT	Industrial Internet of the Thing
IoT	Internet of the Things
IWN	Intelligent Wireless Networks
CNC	Computer Numerical Control
CPPS	Cyber Physical Production Systems
IMS	Intelligent Manufacturing Systems
FMS	Flexible Manufacturing System
CdG	Centro de Gravedad

SIMBOLOGÍA

Deg	Grado
s	Segundos
kg	Kilogramo
m	Metro
mm	Milímetro

ÍNDICE DE FIGURAS

Figura 1.1 Máquina herramienta CNC de configuración 3PRS + XY + C [4].....	1
Figura 1.2 Dassault Systèmes 3DEXperience [7].....	2
Figura 1.3 Revolución Industrial [14]	4
Figura 1.4 Sistema 3PRS + XY + C [4].....	7
Figura 1.5 Representación de un modelo cyberfísico[21]	8
Figura 1.6 Esquema del uso de Ilot[22]	8
Figura 1.7 Elementos de la industria 4.0[23]	9
Figura 1.8 ejemplo de un sistema de manufactura inteligente	10
Figura 2.1 Diseño de sistema cyberfísico	13
Figura 2.2 Diseño de sistema cyberfísico integral	13
Figura 2.3 Diseño de sistema cyberfísico sinérgico [28]	14
Figura 2.4 Matriz de decisión	15
Figura 2.5 Modelo V	16
Figura 2.6 Estructura del árbol de requerimientos	18
Figura 2.7 Estructura del árbol de requerimientos simplificado.....	18
Figura 2.8 Estructura del diagrama funcional	19
Figura 2.9 Estructura del diagrama funcional de la junta prismática uno	20
Figura 2.10 Estructura del diagrama lógico	21
Figura 2.11 Estructura del diagrama lógico del servomotor de la junta prismática uno	22
Figura 2.12 Estructura del diagrama lógico del actuador de la junta prismática uno.....	23
Figura 2.13 Vista superior del modelo 3D 3PRS+XY+C	24
Figura 2.14 Vista frontal del modelo 3D 3PRS+XY+C	24
Figura 2.15 Diagrama esquemático 3PRS	26
Figura 2.16 Estructura de control para un actuador.....	32
Figura 3.1 Configuración inicial del administrador del mecanismo	33
Figura 3.2 Configuración correcta del administrador del mecanismo.....	34
Figura 3.3 Interfaz de simulación del mecanismo	35
Figura 3.4 Estado de madurez del registro de pesos	36
Figura 3.5 Diagrama interno del bloque Set Point	38
Figura 3.6 Diagrama de control dinámico	39

Figura 3.7 Bloques del modelo del actuador	40
Figura 3.8 Bloque de conexión del modelo mecánico	41
Figura 3.9 Estructura del modelo mecánico	42
Figura 3.10 Señal de referencia de posición	43
Figura 3.11 Señal de referencia de velocidad	44
Figura 3.12 Respuesta de posición medida para el arranque	45
Figura 3.13 Respuesta de posición medida.....	45
Figura 3.14 Respuesta de velocidad medida.....	46
Figura 3.15 Respuesta de velocidad medida para $t > 100$ ms	47
Figura 3.16 Ventana de configuración del controlador de posición.....	48
Figura 3.17 Ventana de configuración del controlador de velocidad.....	48
Figura 3.18 Parámetros de control y relaciones externas.....	49
Figura 3.19 Parámetros de control definidos en los requerimientos	50
Figura 3.20 Roles utilizados en el proyecto	51

ÍNDICE DE TABLAS

Tabla 2.1 Elementos de masa no uniforme	25
Tabla 2.2 Parámetros D-H 3PRS	27
Tabla 3.1 Atributos del mecanismo 3PRS+XY+C.....	37
Tabla 3.2 Detalle de costos por hora para el cálculo del precio del servicio	52

CAPÍTULO 1

1. INTRODUCCIÓN

El presente trabajo se refiere al tema de modelos cyberfísicos diseñados para máquinas herramientas. Una máquina herramienta, como se muestra en la Figura 1.1, es una máquina cortadora de metal motorizada que asiste en la gestión del movimiento entre la herramienta de corte y el material de trabajo, cambiando su tamaño y forma [2]. La inclusión de estas máquinas en la industria potenció los procesos de manufactura y redujo los tiempos de trabajo. La nueva revolución de los procesos industriales implica un uso intensivo del internet [3] para la conexión entre sistemas cyberfísicos y máquinas físicas.



Figura 1.1 Máquina herramienta CNC de configuración 3PRS + XY + C [4]

Con el fin de conseguir este avance tecnológico, se necesita de modelos virtuales para las máquinas herramienta que permiten la simulación en tiempo real de los procesos desarrollados en sus pares reales. El internet de las cosas en la industria (IIoT) tiene como función: monitorear, recopilar, intercambiar y analizar información [5]. Estos datos son llevados a una plataforma colaborativa, (*3D Experience from Dassault Systèmes*) la cual posee herramientas que permiten la vinculación de esta información con su modelo virtual.

El interés que impulsa este trabajo es el de contribuir con un objetivo global que apunta a la consolidación de la industria 4.0, ya que en la actualidad este es un concepto visionario pero realista [6]. Uno de los retos que actualmente existe es el

de modelamiento y análisis de los sistemas, pues se busca reducir en lo posible las ecuaciones dinámicas y conseguir modelos de control apropiado. Éstos sistemas deben ser modelados como sistemas de manufactura auto organizados [6].

Dassault Systèmes ha desarrollado un ambiente colaborativo (Figura 1.2) mediante internet que ofrece los medios necesarios para el desarrollo completo de una máquina herramienta virtual. Esto es como motivación para participar en el desarrollo de modelos que puedan servir para trabajos futuros, como herramientas que faciliten el desarrollo de sistemas cyberfísicos.

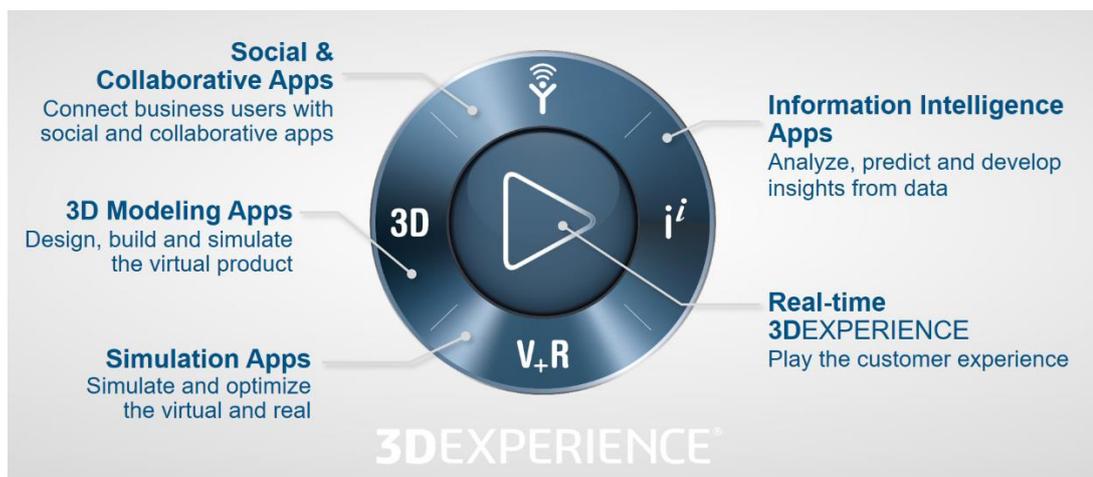


Figura 1.2 Dassault Systèmes 3DExperience [7]

Por otra parte, el desarrollo de un modelo virtual de una máquina micro cortadora para posteriormente ser conectada mediante IoT, permitirá realizar una simulación en tiempo real. Una vez conseguido esto, se deja listo una vía para un futuro aprovechamiento de esta información en integración de inteligencia artificial para la toma de decisiones autónomas [8].

En la actualidad, los modelos virtuales son muy utilizados en las industrias, tanto para procesos como para productos. El uso de modelos virtuales para prototipado permite una reducción de tiempos de diseño, pues se pueden corregir errores sin tener que esperar a producir un elemento físico [9].

Volkswagen es una de las empresas pioneras en el uso de prototipos virtuales para la temprana detección de errores y pruebas de uso para vehículos antes de ser construidos. En 1994 iniciaron la investigación en esta área para desarrollar un sistema de realidad virtual, pues representa un alto costo el producir un nuevo modelo y tomaba entre 4 a 5 años de cambios para que este modelo se fabrique en serie [10].

En cuanto a procesos industriales, existe una tendencia de digitalización de las cadenas de manufactura, para realizar análisis ingenieriles sobre estos. Se han desarrollado sistemas virtuales de procesos para mecanizado de piezas, que permiten el análisis de la interacción entre la herramienta de corte y la pieza de trabajo, este se usa para predecir: fuerzas de corte, torque, potencia y posibles vibraciones u otras eventualidades en la trayectoria de la herramienta, con el fin de realizar correcciones y optimizar los procesos [11].

1.1 Descripción del problema

La industria 4.0 busca la toma de decisiones de manera autónoma en los procesos industriales. Ésta es una visión realista que se proyecta en un futuro cercano y que actualmente presenta desafíos como: mecanismos inteligentes de tomas de decisiones y negociaciones, protocolos IWN de alta velocidad, Big Data de manufactura y análisis, modelamiento de sistemas autónomos y análisis, ciberseguridad, dispositivos físicos modulares y flexibles, y una significativa inversión [6].

Actualmente se está buscando mejoras para el modelamiento de sistemas, una universidad de Madrid realiza un amplio trabajo en conjunto con una reconocida empresa de Diseño, en el cual se ha desarrollado una máquina herramienta CNC de configuración 3PRS + XY + C [4]. Esta máquina además posee un sistema de visión artificial que permite la corrección del posicionamiento de la herramienta [12]. Con estos elementos se desea diseñar un par virtual de la máquina en el que se pueda simular las operaciones en tiempo real.

El desarrollo de un dispositivo virtual de simulación en tiempo real implica un diseño multidisciplinario, pues está compuesto de piezas y mecanismos, circuitos eléctricos y electrónicos, un sistema de control computacional y un ambiente de simulación. La integración de todos estos componentes implica el uso de una gran variedad de software.

1.2 Justificación del problema

La industria ha cambiado a lo largo de la historia, pasando por cuatro revoluciones, como se muestra en la Figura 1.3. La primera incluyó el uso de procesos mecánicos y el aprovechamiento de la hidráulica y las máquinas de vapor. En la segunda se ordenó los procesos para una distribución en masa eficiente, haciendo uso de la electricidad. La tercera integró las tecnologías de computación y automatización [13]. Actualmente la cuarta busca conseguir la integración de sistemas cyberfísicos, los cuales mediante el internet industrial de las cosas se constituyen en un sistema de manufactura inteligente.

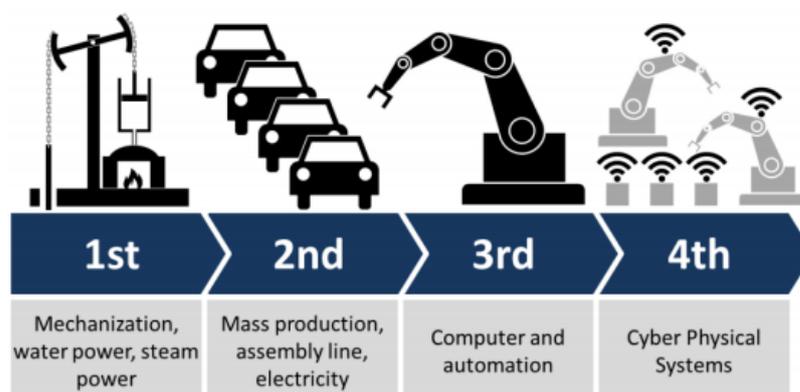


Figura 1.3 Revolución Industrial [14]

Como en todas las revoluciones, esta última busca la optimización de los procesos y los recursos para así aumentar la eficiencia de producción. Ésta se diferencia en el modo en que se optimizan los procesos, pues el objetivo es conseguir que los procesos se mejoren de manera autónoma, que tomen decisiones sobre los parámetros configurables, a partir de la información obtenida en sus operaciones.

Para hacer esto posible se requiere de modelos cyberfísicos que permitan la interpretación de los datos obtenidos en los procesos reales.

Los modelos cyberfísicos son una pieza fundamental de los sistemas de manufactura inteligente [15]. Debido a esto, se busca crear un modelo cyberfísico en una plataforma que brinde todas las herramientas para su integración, que sirva como referencia para el desarrollo industrial de esta tecnología.

1.3 Objetivos

1.3.1 Objetivo general

Desarrollar el modelo cyberfísico de una máquina herramienta CNC de configuración 3PRS + XY + C, para conectarse a sistemas de manufactura inteligente.

1.3.2 Objetivos específicos

1. Realizar un análisis de las herramientas existentes para la implementación de modelos cyberfísicos, utilizados en sistemas de manufactura inteligente.
2. Diseñar la estructura lógico-funcional para a creación del gemelo digital.
3. Configurar un ensamble digital en una plataforma virtual.
4. Simular el comportamiento mecánico y control de la máquina herramienta.
5. Validar las gráficas obtenidas en las simulaciones con datos del sistema físico.

1.4 Marco teórico

En el Ecuador se ha despertado un interés por la automatización de los procesos industriales, las empresas se interesan en hacer análisis sobre sus niveles de automatización, gracias a esto son conscientes de los beneficios que trae la misma a la producción, un ejemplo de estos es RistokCacao S.A. [16]. La empresa del cacao en Ecuador empieza a tomar en consideración los riesgos laborales que existen en sus procesos, debido a esto se está popularizando el uso de procesos autónomos [17].

De este modo nacen planes estratégicos como el de Holcim-Ecuador [18], o iniciativas como la de la empresa Intelware S.A. [19], que son los primeros pasos para el surgimiento de la Industria 4.0 en nuestro país, mejorando la calidad de producción y permitiéndonos competir en el mercado regional.

1.4.1 Máquina herramienta de configuración 3PRS+XY+C

Ésta es una máquina de corte de un solo filo, la cual posee un sistema de tres ejes paralelos para el posicionamiento de la herramienta, el sistema paralelo posee en cada eje una junta prismática, revoluta y esférica (PRS). Este mecanismo sirve para posicionar y orientar el eje rotacional C de la herramienta. Además, para ampliar el rango de trabajo, posee una cama con movimiento XY que mueve la pieza de trabajo [4]. La figura 1.4 es una representación sin carcasa de la máquina.

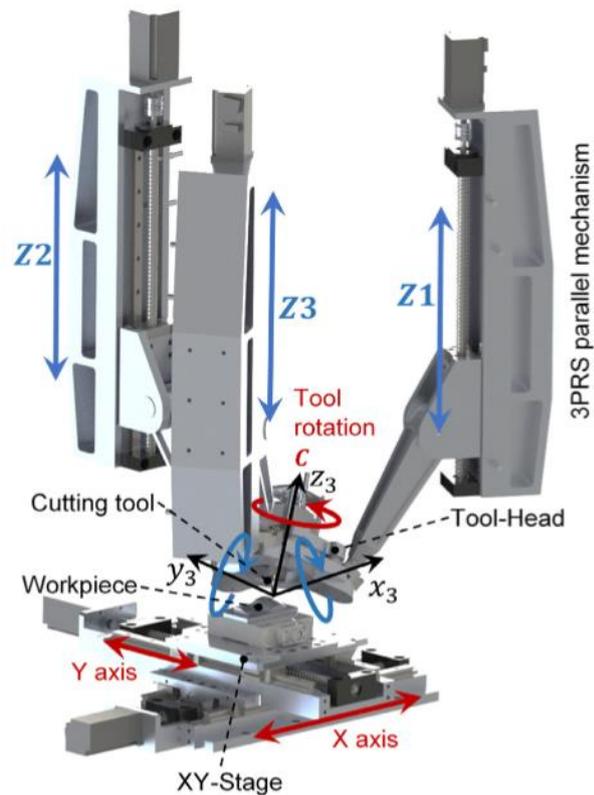


Figura 1.4 Sistema 3PRS + XY + C [4]

1.4.2 Modelo cyberfísico

Estos modelos son una herramienta para el diseño de sistemas cyberfísico, los cuales permiten a los sistemas tener una representación completa de la funcionalidad y las configuraciones de sus equipos [20]. Estos modelos deben tener una gran similitud de comportamiento entre sus partes física y virtual. La figura 1.5 presenta una representación de cómo se conforma un modelo cyberfísico.

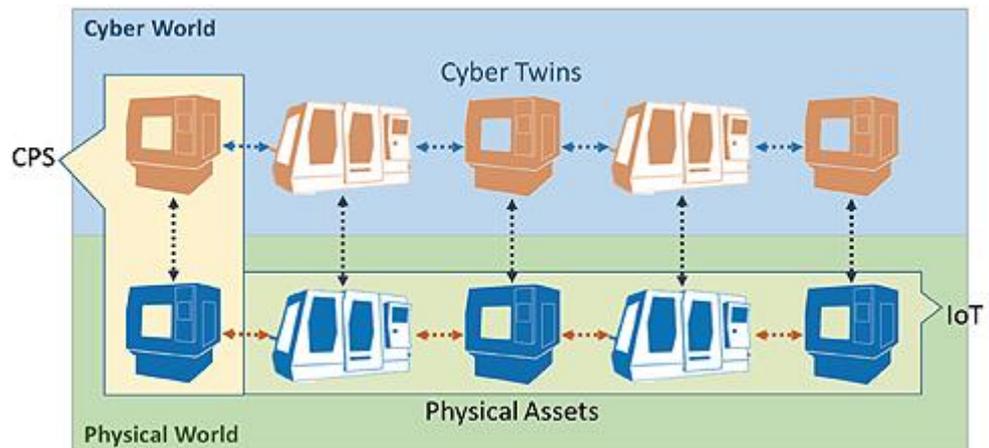


Figura 1.5 Representación de un modelo cyberfísico[21]

1.4.3 Internet Industrial de las cosas (IIoT)

Este nuevo concepto hace referencia a los procesos industriales que utilizan tecnologías del internet de las cosas (IoT) dentro de sistemas cyberfísicos con el propósito de potenciar el logro de los objetivos industriales [5]. La visión del IIoT es la de que todos los elementos de la industria trabajan como un solo sistema de sistemas, como se puede apreciar en la figura 1.6, conformando una empresa de fabricación inteligente.



Figura 1.6 Esquema del uso de IIoT[22]

1.4.4 Industria 4.0

Esta es una manera de llamar a la nueva revolución industrial, la idea consiste en una estructura emergente, la cual combina los procesos de fabricación y los sistemas logísticos en sistemas de producción cyberfísicos (CPPS), los cuales utilizan la información disponible de manera global y las redes de comunicación para un extensivo intercambio de información, en el que los procesos de producción y de negocios coinciden [6]. La figura 1.7 muestra un conjunto de elementos que conformarían este nuevo modelo de industria.



Figura 1.7 Elementos de la industria 4.0[23]

1.4.5 Sistemas de manufactura inteligentes (IMS)

Esta estrategia resulta de la dificultad que existe en la actualidad para el diseño de sistemas flexibles de manufactura (FMS). El diseño de un FMS consiste en la toma de decisiones a partir de la aplicación de criterios de negocios, sin embargo, el mercado actual y futuro se caracteriza por necesitar una respuesta rápida, bajos costos para producción de bajo volumen y calendarización en tiempo real. Para suplir este tipo de mercado los IMS proponen el uso de sistemas holónicos y autónomos [24].

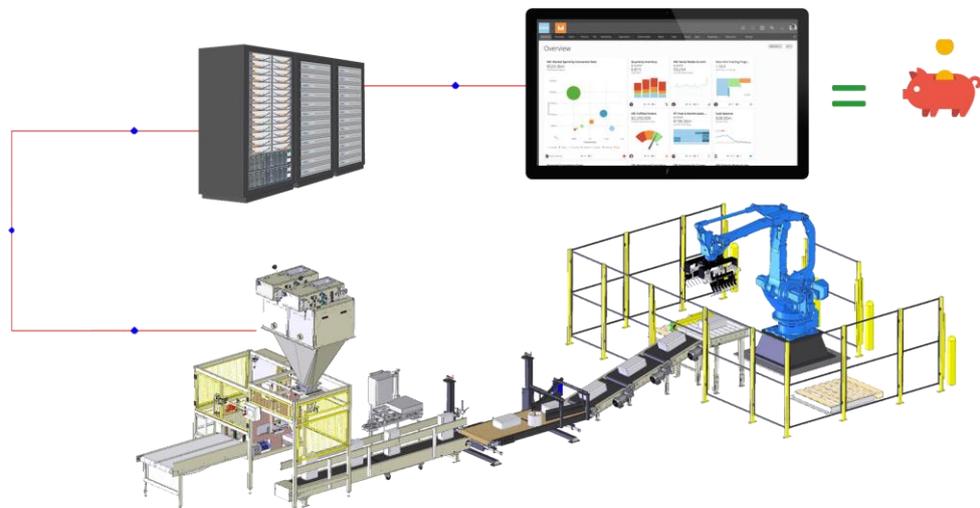


Figura 1.8 ejemplo de un sistema de manufactura inteligente

1.4.6 Enovia

Este software de la familia de *Dassault Systèmes* permite la planificación ordenada que permite la creación de versiones digitales de empresas y proyectos. Estos modelos inteligentes permiten elaborar planes más eficaces e inteligentes [25] gracias a la información en contexto. Su herramienta de requerimientos permite la organización para mejorar el manejo global de los requerimientos, permitiendo captar la voz del cliente.

1.4.7 Catia

Es una de las principales soluciones a nivel mundial para el diseño de productos. Catia permite modelar cualquier producto en contexto a su comportamiento en la vida real [26]. En este conjunto de aplicaciones se facilita la creación de prototipos digitales en conjunto con su simulación y análisis de datos, que permite crear y analizar virtualmente un producto.

1.4.8 Modelica

Es una organización sin fines de lucro instaurada en 1996. Esta organización permite el uso gratuito de una librería que contiene alrededor de 1600 componentes de modelos [27]. Esta organización estableció el lenguaje de modelamiento Modelica que se orienta a objetos en un amplio dominio de aplicaciones como sistemas mecánicos, eléctricos, electrónicos, entre otros. Sus sistemas de ecuaciones no describen asignaciones, sino igualdades por lo que no existe una causalidad hasta que sea interpretado el programa.

CAPÍTULO 2

2. METODOLOGÍA

En este capítulo se escogió la estrategia de desarrollo del proyecto. Se partió de las alternativas de solución, escogiendo la que permite una integración completa del sistema mecatrónico virtual. El proceso de diseño siguió un modelo V que partió de la definición de requerimientos. A continuación, se estableció las funciones y los elementos necesarios en los diagramas lógico y funcional respectivamente. Posterior a esto se realizó el modelo 3D con sus masas, grados de libertad, conexiones ingenieriles y representación mecánica. Finalmente se analizó el sistema mecánico en su enfoque cinemático y dinámico, para diseñar el modelo del sistema y su control.

2.1 Alternativas de solución

Al momento de escoger la solución que nos permita realizar el gemelo digital se consideraron tres alternativas posibles, de entre las herramientas existentes, las cuales poseen un distinto grado de dificultad al momento de implementarse:

- Diseño de un modelo cyberfísico a partir de datos extraídos del modelo físico mediante el uso de IoT y procesamiento en la nube de Big Data. Este genera respuestas a partir de los datos previos Figura 2.1.

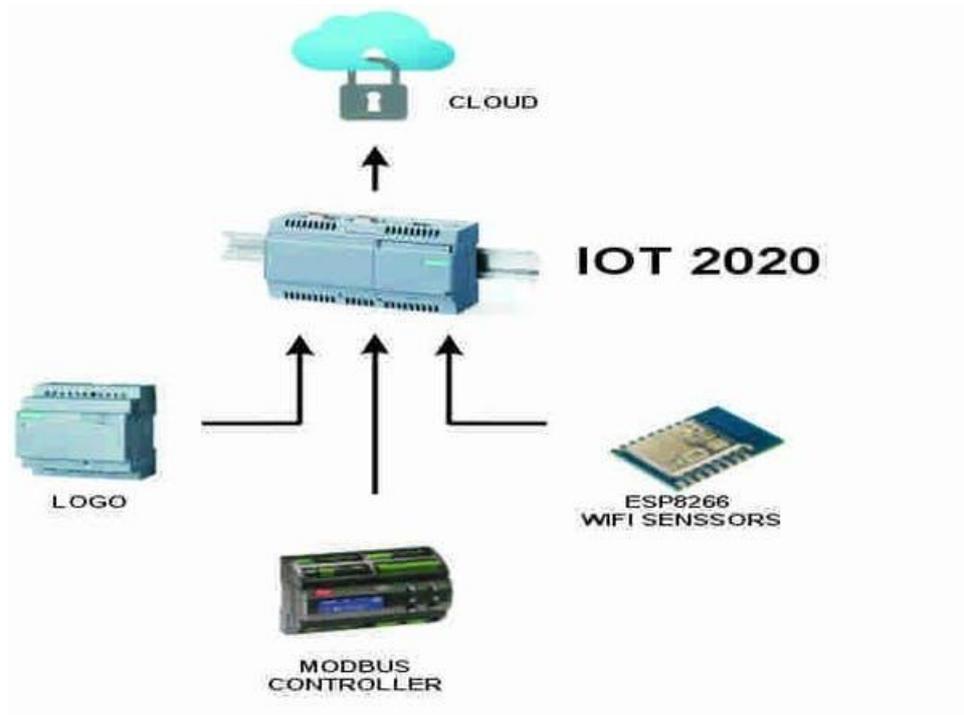


Figura 2.1 Diseño de sistema cyberfísico

- Diseño integral de un modelo cyberfísico, mediante el trabajo conjunto de distintas plataformas (LabVIEW, SolidWorks, Matlab). Esta opción genera sus propias respuestas a partir de los datos obtenidos Figura 2.2.

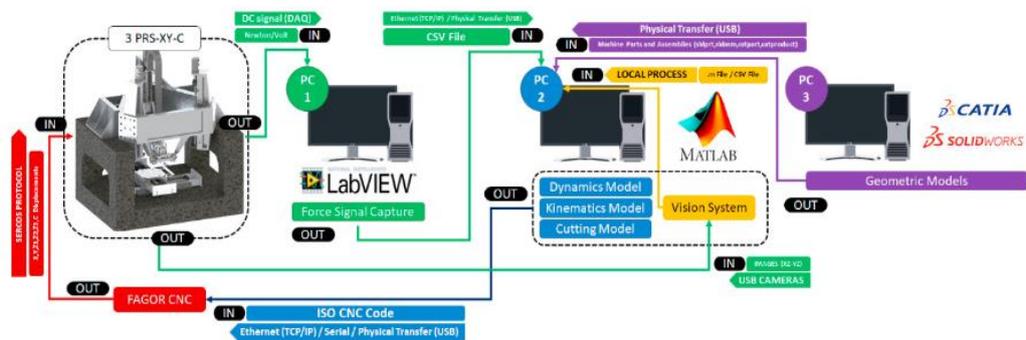


Figura 2.2 Diseño de sistema cyberfísico integral

- Diseño sinérgico de un modelo cyberfísico, dentro de una misma plataforma (3DEXPERIENCE). Esta opción genera respuestas tanto de simulaciones como de datos obtenidos del modelo físico Figura 2.3.

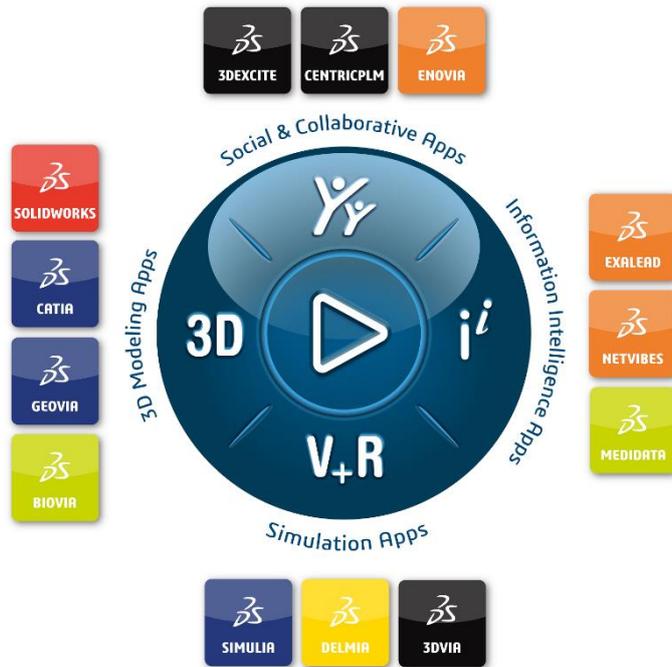


Figura 2.3 Diseño de sistema cyberfísico sinérgico [28]

Los criterios usados para la selección de la mejor solución se describen a continuación:

- **Desempeño:** El modelo virtual debe poseer una alta semejanza con el modelo real para producir respuestas válidas.
- **Extensivo:** Se puede implementar un sistema de toma de decisiones autónomas, para lo cual se debe poder enviar y recibir órdenes.
- **Flexible:** El diseño debe ser fácilmente modificable y extensible a modelos semejantes.
- **Costo Inicial:** Inversión a realizarse para la implementación de los dispositivos conectados a internet y el manejo de datos.
- **Compatibilidad:** Facilidad para el manejo de los datos y el envío de estos a través de los distintos softwares a usarse.
- **Análisis de datos:** Facilidad para la interpretación de los datos obtenidos por los modelos creados.

- **Costo de operación:** Costos por el manejo de los datos y el mantenimiento de los dispositivos necesarios para que el modelo virtual y el físico estén conectados.

De las soluciones propuestas, la que consiguió un mejor cumplimiento de los criterios es la de diseño sinérgico mediante 3DExperience. Esta opción se destaca al posibilitar la creación de modelos muy apegados al diseño real, fácilmente extensible para el control del modelo físico a partir del virtual, altamente flexible por su diseño en bloques y su control acausal. Además, el entorno se encarga de la comunicación de datos entre programas, evitando esta tarea y reduciendo pérdidas de información. En la Figura 2.4 se puede observar la matriz de decisión que permitió conseguir este resultado.

MATRIZ DE DECISIÓN

	CRITERIO 1	CRITERIO 2	CRITERIO 3	CRITERIO 4	CRITERIO 5	CRITERIO 6	CRITERIO 7	
Descripción	Desempeño (Semejanza física-virtual)	Extensivo (toma de decisiones autonomas)	Flexible	Costo Inicial	Compatible entre plataformas	Analisis de datos	Costo de operación	
Peso	7	4	3,5	3	2,5	1,5	1	Total 22,5
	31%	18%	16%	13%	11%	7%	4%	100%
Opciones	Calificación	Calificación	Calificación	Calificación	Calificación	Calificación	Calificación	
PLC con análisis de Data en la nube	4	2	1	8	3	4	7	3,1
LabVIEW, Matlab, Catia	6	6	7	7	6	6	7	5,3
Dassault Systemes 3DExperiences	9	7	8	4	10	7	7	6,1

Figura 2.4 Matriz de decisión

2.2 Proceso de diseño

Para el desarrollo del proceso de diseño se decidió trabajar con el modelo V, el cual es una herramienta útil para la fase de desarrollo. Existen cuatro etapas de definición en las que se establecen los requisitos, las operaciones, la estructura y finalmente la forma. Con la solución lista se procede a la ejecución y revisión de cada una de las

etapas, finalizando en la validación de los requerimientos. La figura 2.5 muestra el orden a seguir y las relaciones entre las etapas.

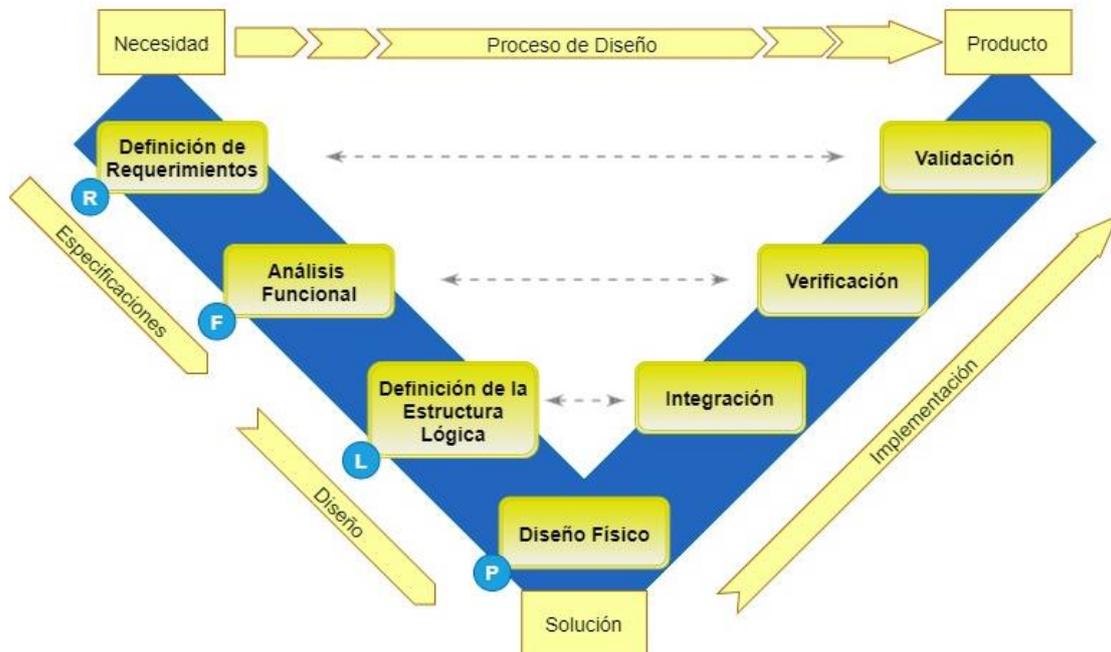


Figura 2.5 Modelo V

2.2.1 Definición de requerimientos (R)

Esta fue la primera y más importante etapa. En esta se establecieron las características de la solución de modo que se cumpliera con las necesidades del cliente. Los requerimientos pueden clasificarse en 6 tipos [29]. En base a estos se definieron los del presente proyecto.

2.2.1.1 *Requerimientos funcionales*

- El mecanismo tiene seis grados de libertad.
- El desplazamiento en Z se realiza mediante tres ejes paralelos.
- La herramienta de trabajo gira sobre su propio eje (C).
- Desplazamiento en XY para aumentar el área de trabajo.

- Diseño de modelo acausal.
- Alimentación trifásica.
- Articulaciones accionadas por servomotores.
- El control de los tres ejes Z permitirá el *roll* y *pitch*.
- El sistema debe evitar colisiones.
- Bajo error de posición por deformación.

2.2.1.2 *Requerimientos de desempeño*

- Desplazamiento lineal mínimo en cada eje de $1\mu m$.
- Desplazamiento angular mínimo del eje C: 0.001° .
- Área de corte en el rango de $4\mu m^2 - 2500\mu m^2$.

2.2.1.3 *Requerimientos de restricción*

- El ángulo entre el brazo y la cabeza de la herramienta en cada junta esférica nunca deberá ser mayor a 180° .
- Tiempo de desarrollo 4 meses.
- Diseñado mediante software de Dassault Systèmes.

2.2.1.4 *Requerimientos de interfaz*

- Recibe una señal de posición deseada y devuelve la posición conseguida.
- Tiene acceso a internet para transferencia de datos.

2.2.1.5 *Otros requerimientos*

- Accionamiento de parada que detiene el sistema en emergencias.
- Interfaz de comprensión intuitiva.

Los requerimientos listados se manejaron y desarrollaron en niveles mediante la herramienta de base de datos para requerimientos de Enovia. La figura 2.6 muestra la estructura completa de los requerimientos.

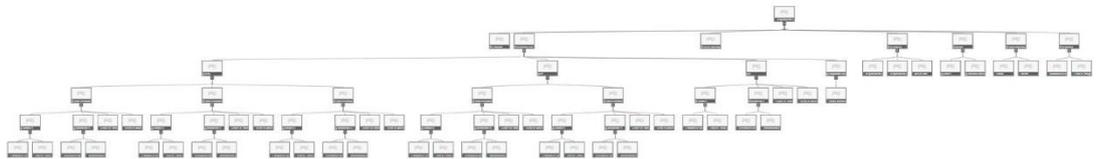


Figura 2.6 Estructura del árbol de requerimientos

La figura 2.7 muestra el árbol de requerimientos de manera simplificada junto al entorno de trabajo de Enovia.



Figura 2.7 Estructura del árbol de requerimientos simplificado

2.2.2 Análisis funcional (F)

Esta segunda etapa se estableció a partir de los requerimientos funcionales. El objetivo fue el de analizar y posteriormente representar las distintas funciones, que realiza el sistema, de manera abstracta. En esta etapa de diseño se buscó responder que es lo que realizaría el sistema [30]. El análisis partió de la representación funcional de un servomotor, expandiéndose en este caso para un total de seis actuadores. La figura 2.8 muestra el diagrama funcional obtenido y sus relaciones correspondientes.

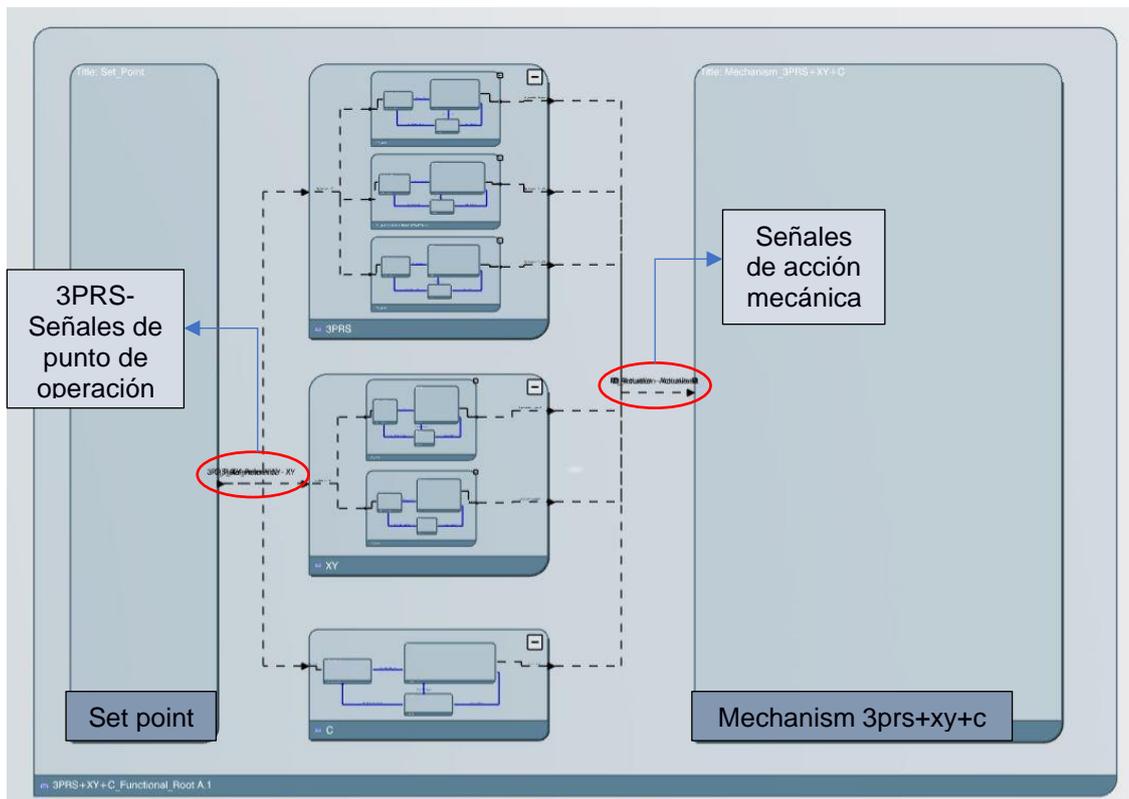


Figura 2.8 Estructura del diagrama funcional

El diagrama consta de un bloque de puntos de operación, un bloque para el mecanismo y tres bloques para los actuadores. El primer bloque de actuadores es el 3PRS que contiene las tres funciones correspondientes a cada motor del eje Z. El segundo tiene las funciones de los actuadores X e Y. El último bloque contiene la función del actuador del eje C rotacional. La figura 2.9 muestra la representación abstracta de la función de uno de los actuadores.

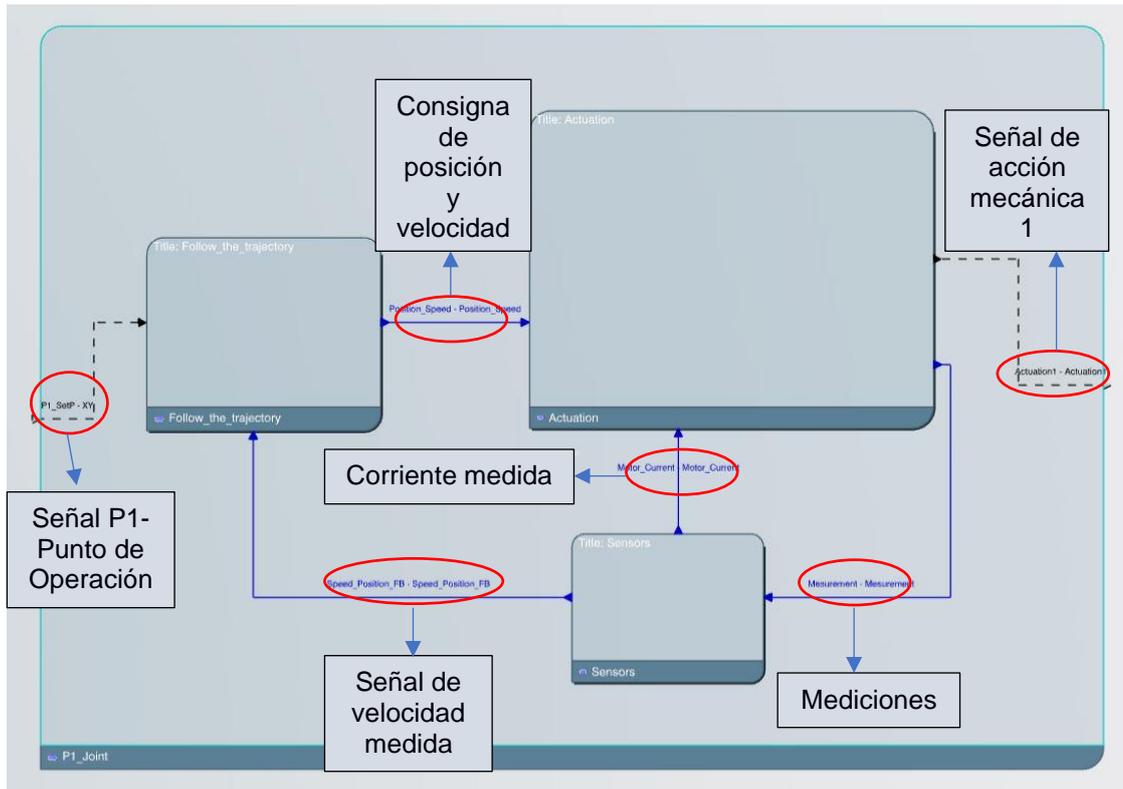


Figura 2.9 Estructura del diagrama funcional de la junta prismática uno

Estas son las estructuras de menor nivel del diagrama. Se puede observar las tres funciones necesarias para el funcionamiento de un actuador. Se necesita una función que cree la trayectoria, una que cree la actuación y una que procese la información de retroalimentación. El diagrama también representa las señales tanto internas como externas.

2.2.3 Diseño lógico (L)

El diseño de la estructura lógica trata de llevar el concepto abstracto del diagrama funcional a una implementación de los componentes reales [31]. Lo que se realizó en esta etapa fue la implementación estructural de los elementos y sus conexiones lógicas. En la figura 2.10 se muestra el diagrama lógico completo y las señales necesarias.

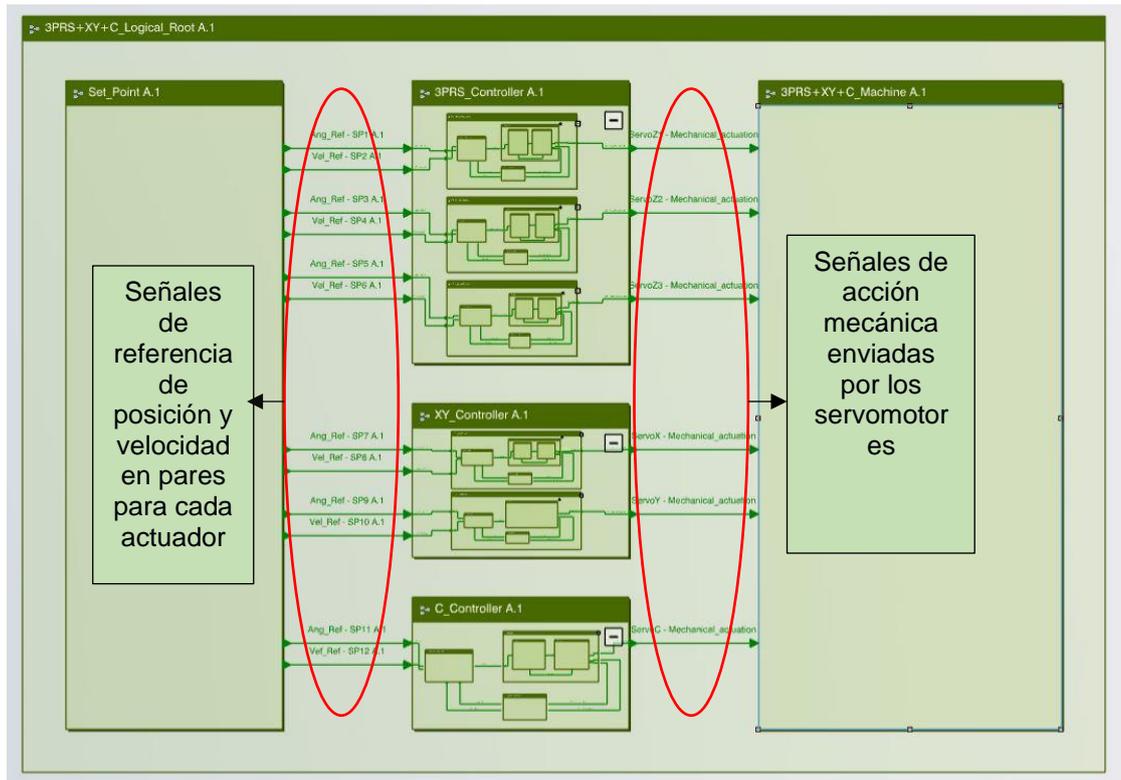


Figura 2.10 Estructura del diagrama lógico

Del bloque de puntos de operación salen doce señales, seis pares como referencia a cada motor. Cada par consta de una señal de velocidad y posición angulares deseada. De cada actuador se emite una señal de accionamiento mecánico a la junta correspondiente. La figura 2.11 muestra cómo se conformó la estructura de control de uno de los servomotores.

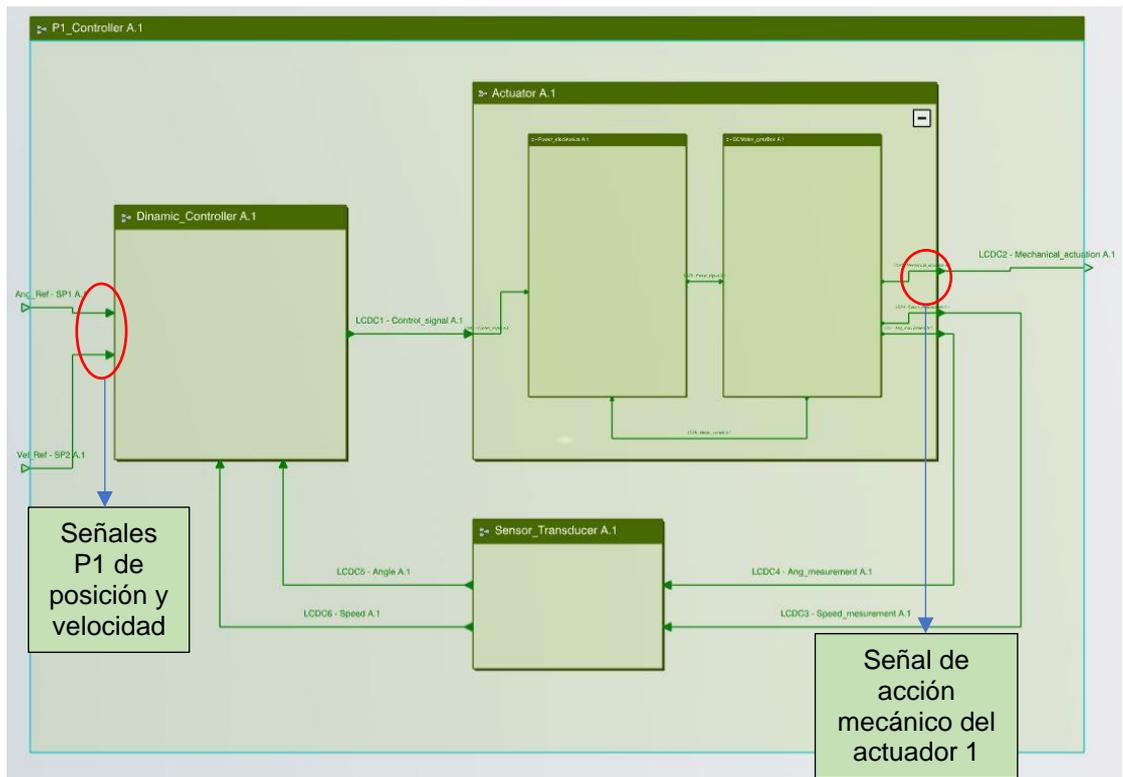


Figura 2.11 Estructura del diagrama lógico del servomotor de la junta prismática uno

El dispositivo está compuesto del controlador dinámico, el actuador y el sensor-transductor. En este bloque se realizó posteriormente la configuración del comportamiento físico de cada componente. La figura 2.12 muestra cómo se diseñó la estructura del actuador de este servomotor y sus señales respectivas.

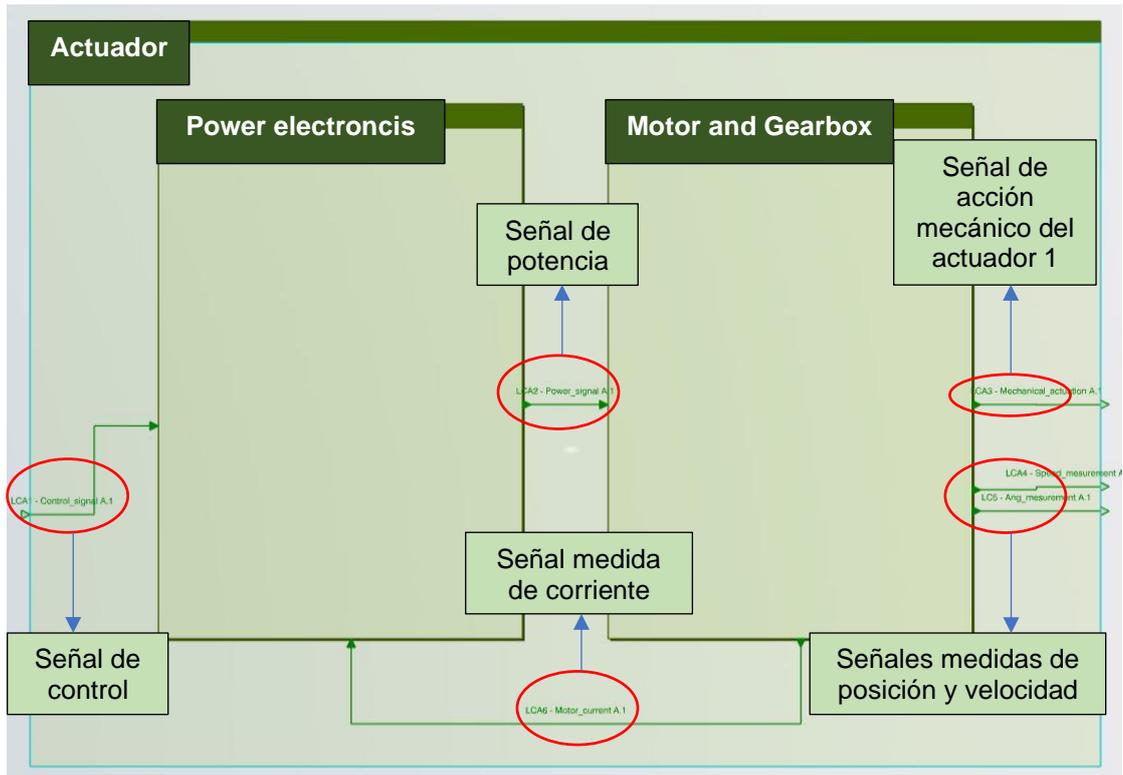


Figura 2.12 Estructura del diagrama lógico del actuador de la junta prismática uno

2.2.4 Diseño físico (P)

El diseño físico constó de seis etapas las cuales en orden fueron: creación de las partes, aplicación de materiales, ensamble del sistema, revisión de grados de libertad, definición de pesos y creación de la representación mecánica.

La figura 2.13 y 2.14 muestran el resultado del ensamblaje.

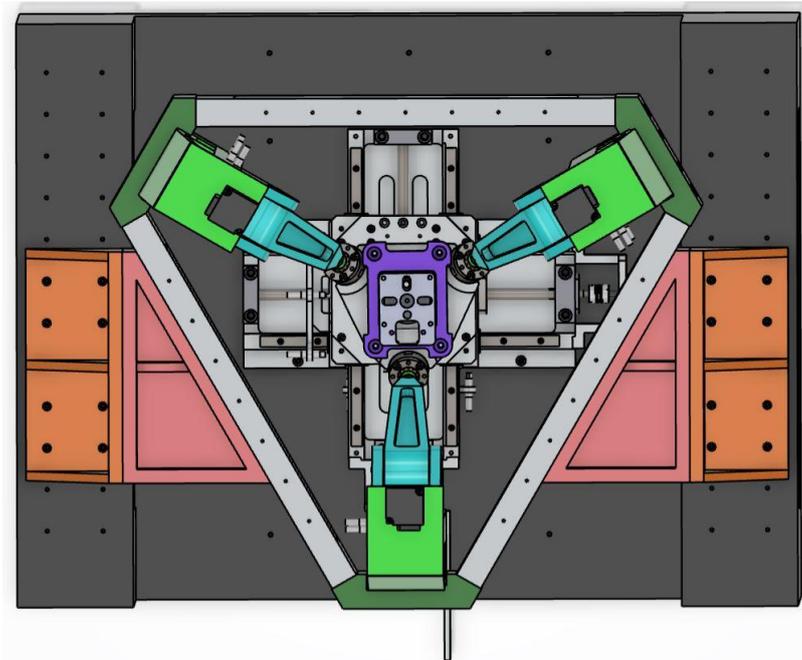


Figura 2.13 Vista superior del modelo 3D 3PRR+XY+C

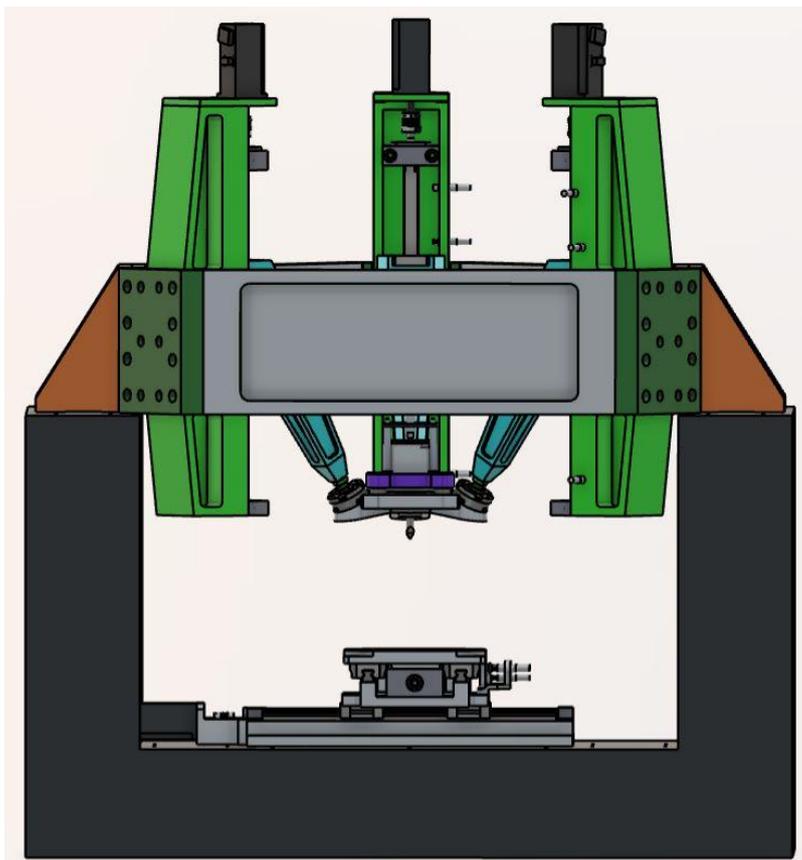


Figura 2.14 Vista frontal del modelo 3D 3PRR+XY+C

La asignación de materiales en este proyecto permitió el análisis de pesos mediante software a través del cómputo de este. Para realizar este procedimiento fue necesaria la creación de los materiales con sus respectivas propiedades.

Adicionalmente, para los elementos de material no uniforme (sensores, motores, etc.) el peso fue obtenido de las hojas de datos correspondientes. La tabla 2.1 detalla estos elementos.

Tabla 2.1 Elementos de masa no uniforme

N.º	Nombre	Tipo	Referencia
1	Guías Lineales IKO LRX-12-R300-C2-T1-H	Mecánica	[32]
2	Husillos NSK PSS1205N1D0421	Mecánica	[33]
3	Rotulas HEPHAIST SRJ012C	Mecánica	[34]
4	Soportes Husillo NSK 606ZZ WBK08S-01B	Mecánica	[35]
5	Soportes Husillo NSK 606ZZ WBK08-01B	Mecánica	[35]
6	Soportes Husillo NSK S7201	Mecánica	[35]
7	Soportes Husillo NSK S7000	Mecánica	[35]
8	Soportes Husillo NSK S706	Mecánica	[36]
9	Soportes Husillo NSK S728	Mecánica	[36]
10	RULAND PCMR-25-8-6-A	Mecánica	[37]
11	HarmonicDrive FHA-11C-100-D200-EM1	Actuador	[38]
12	Mitsubishi HF-MP13	Actuador	[39]
13	Omron E2B-S08KN04-WP-B1	Sensor	[40]
14	KISTLER 9017B	Sensor	[41]

2.3 Análisis Denavit-Hartenberg 3PRS

El mecanismo paralelo 3PRS posee 3 grados de libertad; cada pierna posee una junta prismática (verde), revoluta (celeste) y esférica (gris). La figura 2.15 representa la cinemática del mecanismo. Se define el sistema de referencia O que representa

el sistema global, y el p que representa el sistema local ubicado en la cabeza de la herramienta.

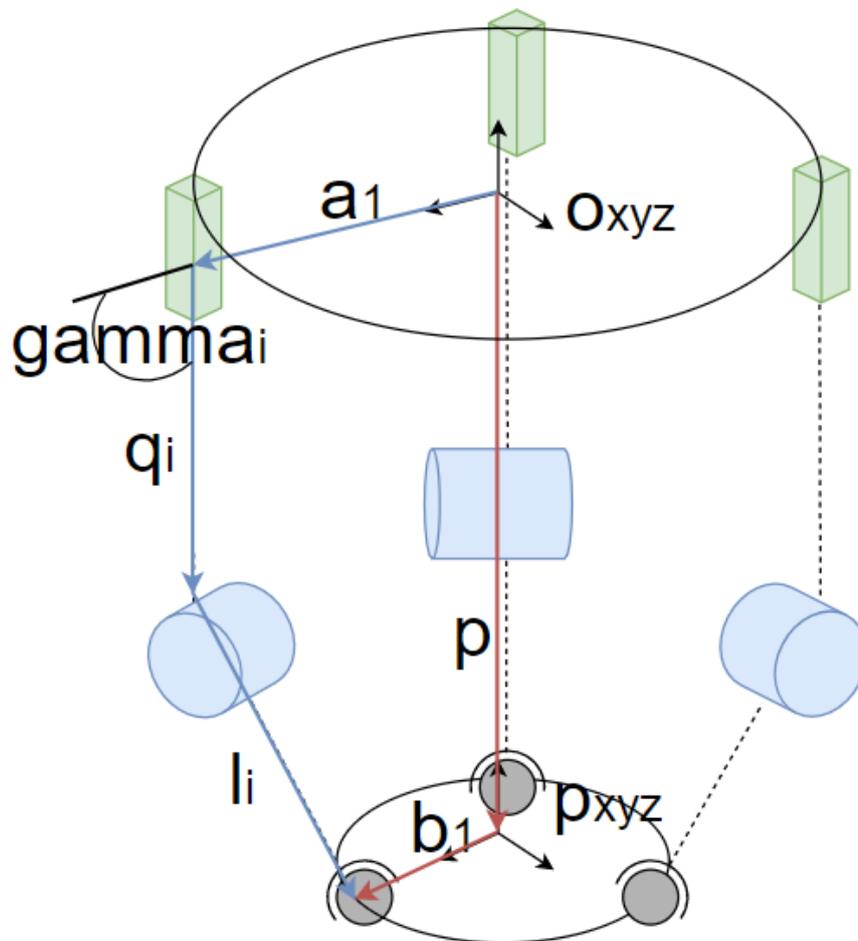


Figura 2.15 Diagrama esquemático 3PRS

Para modelar las piernas se hizo uso de la notación de Denavit-Hartenberg. La tabla 2.2 presenta los parámetros para cada pierna y par cinemático. Para modelar el sistema de coordenadas se utilizó el sistema de ángulos de Euler XYZ (*pitch*, *roll* *yaw*).

Tabla 2.2 Parámetros D-H 3PRS

i	$\theta_{i,1}$	$d_{i,1}$	$a_{i,1}$	$\alpha_{i,1}$	$\theta_{i,2}$	$d_{i,2}$	$a_{i,2}$	$\alpha_{i,2}$
1	$\pi/2$	d_1	0	γ_1	θ_1	0	0	$-\pi/2$
2	$\pi/2$	d_2	0	γ_2	θ_2	0	0	$-\pi/2$
3	$\pi/2$	d_3	0	γ_3	θ_3	0	0	$-\pi/2$

A partir de estos parámetros se obtuvo la matriz de rotación del sistema p con respecto al sistema global de coordenadas O. La ecuación 2.1 muestra la matriz de rotación resultante.

$${}^oR_p = \begin{bmatrix} C_\alpha C_\beta & C_\alpha S_\beta S_\phi - S_\alpha C_\phi & C_\alpha S_\beta S_\phi - S_\alpha S_\phi \\ S_\alpha C_\beta & S_\alpha S_\beta S_\phi + C_\alpha S_\beta & S_\alpha S_\beta C_\phi + C_\alpha S_\phi \\ -S_\beta & C_\beta S_\phi & C_\beta C_\phi \end{bmatrix} \quad (2.1)$$

Siendo $C_x = \cos(x)$ y $S_x = \sin(x)$

Las coordenadas para el elemento terminal son representadas en la ecuación 2.2 y las restricciones en la 2.3.

$$x = [x_p \quad y_p \quad z_p \quad \phi \quad \beta \quad \alpha]^T \quad (2.2)$$

$$q = [q_1^T \quad q_2^T \quad q_3^T]^T \quad (2.3)$$

Los elementos de la matriz de juntas tienen la forma expresada en la ecuación 2.4.

$$q_i = \begin{bmatrix} d_i \\ \theta_i \end{bmatrix} \quad (2.4)$$

Para determinar las componentes del punto b con respecto al sistema local p se usaron las expresiones de la ecuación 2.5.

$${}^p b_1 = \begin{bmatrix} h \\ 0 \\ 0 \end{bmatrix} \quad {}^p b_2 = \begin{bmatrix} -\frac{1}{2}h \\ -\frac{\sqrt{3}}{2}h \\ 0 \end{bmatrix} \quad {}^p b_3 = \begin{bmatrix} -\frac{1}{2}h \\ -\frac{\sqrt{3}}{2}h \\ 0 \end{bmatrix} \quad (2.5)$$

Así mismo, para determinar las componentes del punto a con respecto al sistema local O se usaron las expresiones de la ecuación 2.6.

2.4 Análisis cinemático 3PRS

El lazo vectorial utilizado para el análisis se encuentra representado con sus dos caminos (azul y rojo) en la figura 2.15 y su ecuación correspondiente es la 2.6

$$p + b_i = a_i + q_i u_{i,1} + l_i u_{i,2} \quad (2.6)$$

En la ecuación 2.6 las variables u son vectores unitarios que definen la línea de acción de la junta prismática y el que une la revoluta con la junta prismática. Estos vectores son función de d y theta (θ) respectivamente. Se puede eliminar theta si se despeja la variable l y se eleva al cuadrado la ecuación [42]. Obteniendo así la ecuación 2.7 que nos permitirá conseguir la ecuación 2.8 mediante manipulaciones algebraicas. Esta última describe la relación entre la coordenada asociada y el par prismático.

$$[p + b_i - a_i - q_i u_{i,1}]^T [p + b_i - a_i - q_i u_{i,1}] = l_i^2 \quad (2.7)$$

$$d_i = c_i^T \cdot u_{i,1} \pm \sqrt{(c_i^T \cdot u_{i,1})^2 - c_i^T \cdot c_i + l_i^2} \quad (2.8)$$

Siendo $c_i = p + b_i - a_i$.

De este modo, se puede hallar theta despejándolo de la ecuación 2.8 luego de reemplazar d en la 2.6.

Este sistema presenta 3 grados de libertad, por lo que se necesitó encontrar la relación entre 3 de las 6 coordenadas de la herramienta con respecto al resto. Estas se consiguen a través de la premisa establecida en las ecuaciones 2.9, 2.10 y 2.11, la cual define que el movimiento de una pierna respecto al sistema de coordenadas $O_{i,2}$ tiene como componente del eje z el valor de cero [43].

$$z_1^T [{}^{1,2}R_o p + {}^{1,2}R_o {}^oR_p {}^p b_1] = 0 \quad (2.9)$$

$$z_2^T [{}^{2,2}R_o p + {}^{2,2}R_o {}^oR_p {}^p b_2] = 0 \quad (2.10)$$

$$z_3^T [{}^{2,3}R_o p + {}^{2,3}R_o {}^oR_p {}^p b_3] = 0 \quad (2.11)$$

Siendo $z_i = [0 \ 0 \ 1]^T$ y la matriz ${}^{1,2}R_o$ de rotación entre el sistema $O_{i,2}$ respecto a O. Las ecuaciones 2.12, 2.13 y 2.14 presentan la solución del sistema anterior para tres coordenadas.

$$x_p = -h s_\alpha c_\beta \quad (2.12)$$

$$y_p = -\frac{1}{2} h (c_\alpha c_\beta - s_\alpha s_\beta s_\phi - c_\alpha c_\phi) \quad (2.13)$$

$$\alpha = \text{atan} (s_\beta s_\phi, c_\phi - c_\beta) \quad (2.14)$$

A partir de estas coordenadas independientes se puede obtener las restantes del mecanismo, siendo esta la solución para la posición de la herramienta.

Para obtener la resolución de velocidad para el sistema se tuvo que determinar inicialmente la velocidad de B_i respecto a $O_{i,2}$ como establece la ecuación 2.15

$${}^{i,2}V_{B_i} = {}^iJ_q \vec{q}_i \quad (2.15)$$

Siendo el jacobiano del sistema el expresado en la ecuación 2.16.

$${}^i J_q = \begin{bmatrix} -\sin(\theta_i) & 0 \\ -\cos(\theta_i) & l_i \\ 0 & 0 \end{bmatrix} \quad (2.16)$$

Así también la ecuación 2.17 expresa la velocidad de la herramienta con respecto al punto B_i .

$${}^{i,2}V_{B_i} = [{}^{i,2}R_o - {}^{i,2}R_o {}^o R_p {}^p \tilde{b}_i] \quad (2.17)$$

A partir de las ecuaciones 2.16 y 2.17 se obtuvo una relación entre la herramienta y las juntas expresada en la ecuación 2.18.

$$J_q \dot{q} = J_x \dot{x} \quad (2.18)$$

La ecuación 2.18 resuelve el problema de velocidad, sin embargo, fue necesario expresarla en función a los tres grados de libertad; esto se expresa en la ecuación 2.19.

$$q_a = J_p^{-1} \dot{x}_p \quad (2.19)$$

La cual considera la actuación mediante el par prismático define a J como se expresa en la ecuación 2.20.

$$J_p^{-1} = (J_q^{-1} J_x)([1 \ 3 \ 5], :) \quad (2.20)$$

El problema de aceleración tuvo una resolución similar a la de velocidad, relacionando la aceleración el punto B_i respecto a la herramienta y a cada pierna, con estas consideraciones se definió la ecuación 2.21 de aceleración.

$$\ddot{q}_i = {}^i J_q (a_{b_i} - {}^i \dot{J}_q \dot{q}) \quad (2.21)$$

Describiendo el último término de la ecuación anterior en la ecuación 2.22 como:

$${}^i J_q \dot{q} = \begin{bmatrix} -c\theta_i \ddot{\theta}_i \dot{d}_i \\ -s\theta_i \ddot{\theta}_i \dot{d}_i \end{bmatrix} \quad (2.22)$$

Para la aceleración del punto B se hizo una relación mediante la aceleración de x descrita a continuación en la ecuación 2.23.

$$a_{B_i} = [{}^{i,2}R_0 \quad -{}^{i,2}R_0 {}^0R_p {}^p \tilde{b}_i] \ddot{x}_p + {}^{i,2}R_0 \vec{\omega}_p \times (\vec{\omega}_p \times {}^0R_p {}^p b_i) \quad (2.23)$$

Así, finalmente, se obtuvo que la ecuación 2.24 expresa la aceleración de la herramienta respecto a los tres grados independientes.

$$\dot{\omega} = J_r a_p + J_r \dot{v}_p \quad (2.24)$$

2.5 Análisis Dinámico 3PRS

Para una simplificación del modelo se usó la estrategia de separar en dos sistemas el mecanismo 3PRS. El primero se refiere a la plataforma móvil (la herramienta) y el segundo se refiere a las piernas. Esto permite representar a las piernas como una cadena cinemática de lazo abierto [44]. La herramienta fue analizada en función a las coordenadas de las coordenadas teniendo como resultado la ecuación 2.25.

$$\vec{f}_p = \begin{bmatrix} m_p \vec{g} - m_p \vec{a}_p \\ -I_p \omega_p - \omega_p \times (I_p \omega_p) \end{bmatrix} \quad (2.25)$$

Habiendo determinado las dinámicas de las piernas y de la plataforma, el modelo en conjunto se obtiene mediante proyecciones sobre el espacio de juntas activas [45]. Esto resulta en la ecuación general 2.26.

$$\tau = J_k^T \vec{f}_p + G_1^T \vec{h} + J_k^T \vec{f}_e \quad (2.26)$$

La cual luego de manipulación y proyección al espacio de la plataforma y posteriormente al de las juntas activas permite el accionamiento en las juntas prismáticas de acuerdo con las ecuaciones 2.27 y 2.28

$$\tau = h_i^a + J_k^T [f_p + G_2^T h_i^p] \quad (2.27)$$

$$G_2 = (J_q^{-1} J_x) ([2 \quad 4 \quad 6], :) \quad (2.28)$$

2.6 Estructura de control

Para este sistema el control se realizó sobre cada actuador, ya que estos comandan los grados de libertad que mueven nuestro mecanismo. Cada actuador recibirá una consigna de posición y velocidad por lo que el control se realizará en cascada. La estructura del controlador se muestra en la figura 2.16.

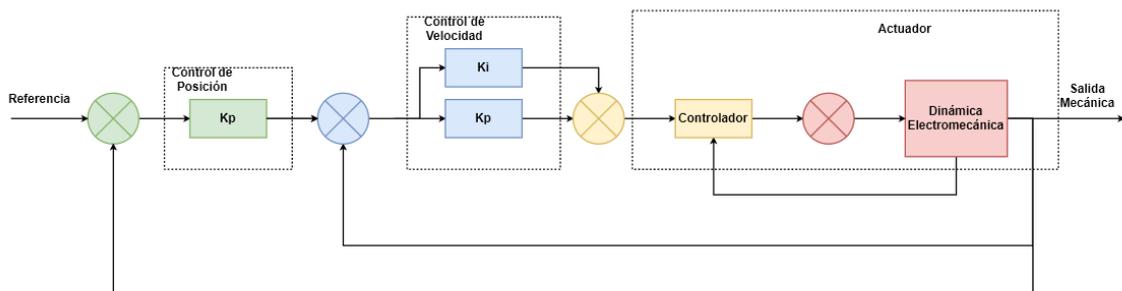


Figura 2.16 Estructura de control para un actuador

CAPÍTULO 3

3. RESULTADOS Y ANÁLISIS

3.1 Mecanismo 3PRS+XY+C 3D

Como resultado del ensamble se obtuvo un conjunto de veinte piezas compuestas las cuales han sido unidas mediante treinta y seis conexiones ingenieriles, todas descritas en la sección de apéndice. Este ensamble permitió crear una representación mecánica con seis grados de libertad que puede ser accionada mediante once comandos distintos como muestra la figura 3.1

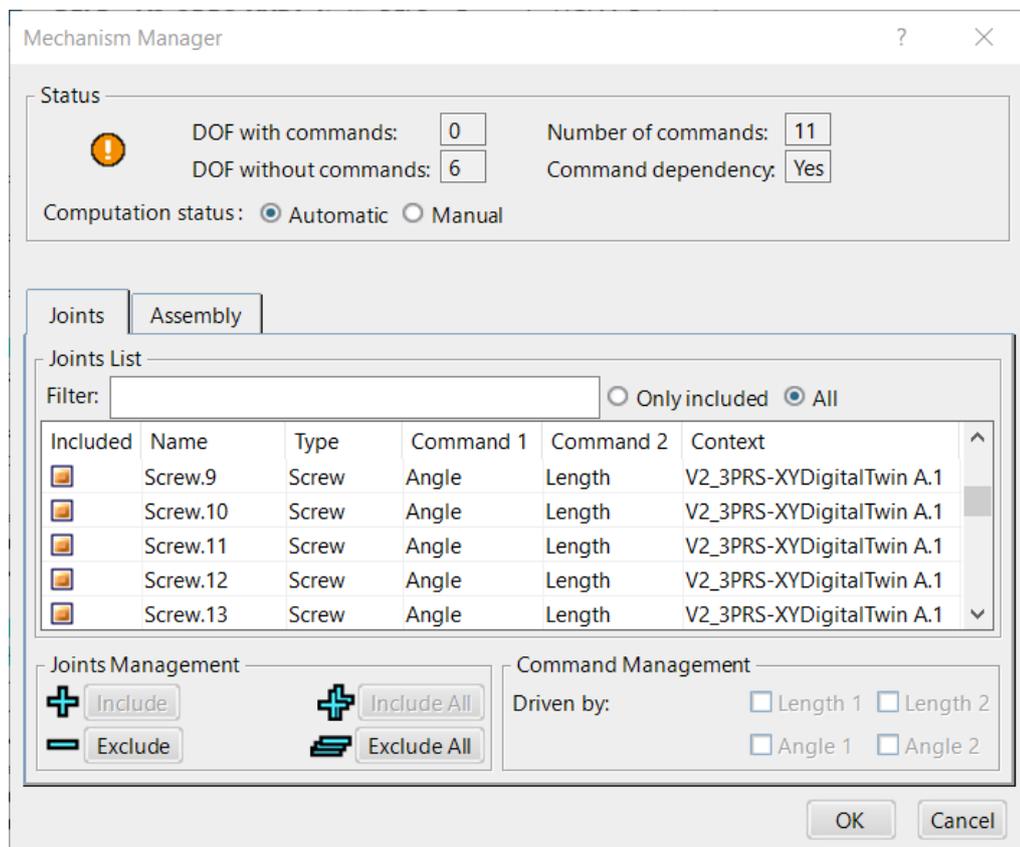


Figura 3.1 Configuración inicial del administrador del mecanismo

Al estado mostrado en la figura 3.1 se le ha configurado los comandos para conseguir que coincidan con el número de grados de libertad. Para conseguir esto se configuró las juntas de tornillo (*screw*) de la nueve a la trece por comando de tipo angular (*angle*), como se muestra en la figura 3.2.

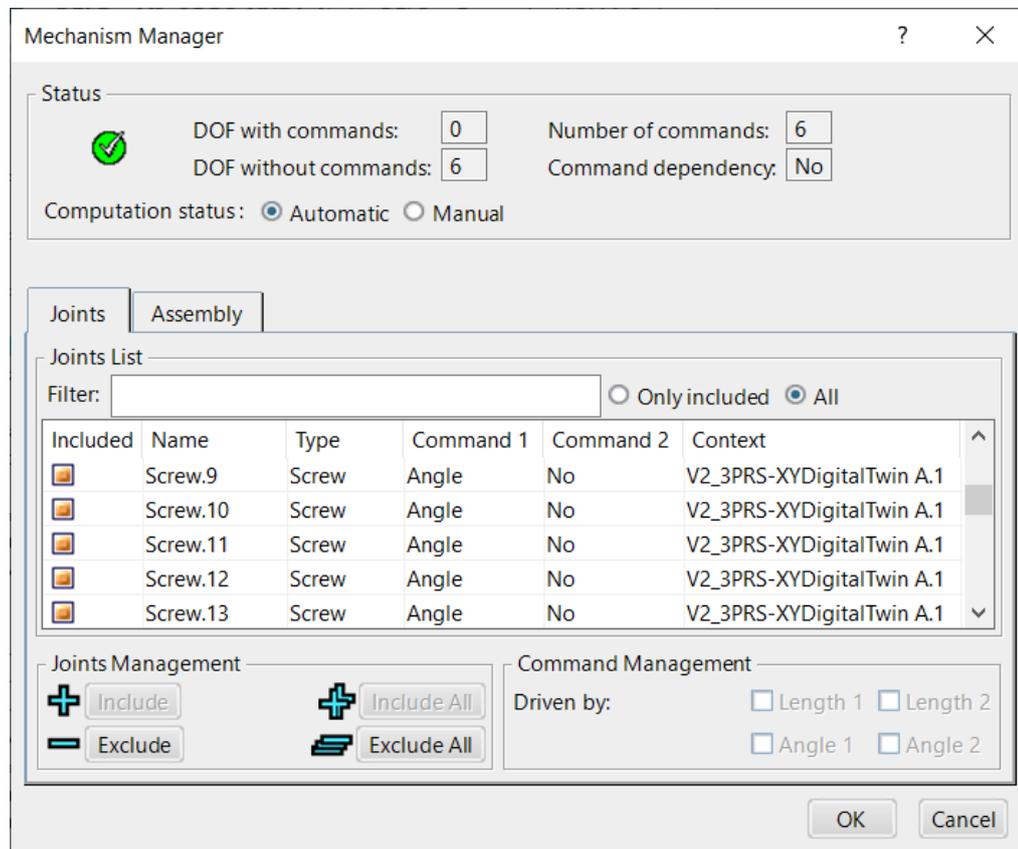


Figura 3.2 Configuración correcta del administrador del mecanismo

Con esto realizado, el simulador de mecanismo estuvo listo para comprobar el comportamiento y ser usado en conjunto con los comportamientos del modelo realizado en Modelica. La figura 3.3 muestra las entradas del sistema configuradas de modo que puedan ser accionadas a partir de un actuador.

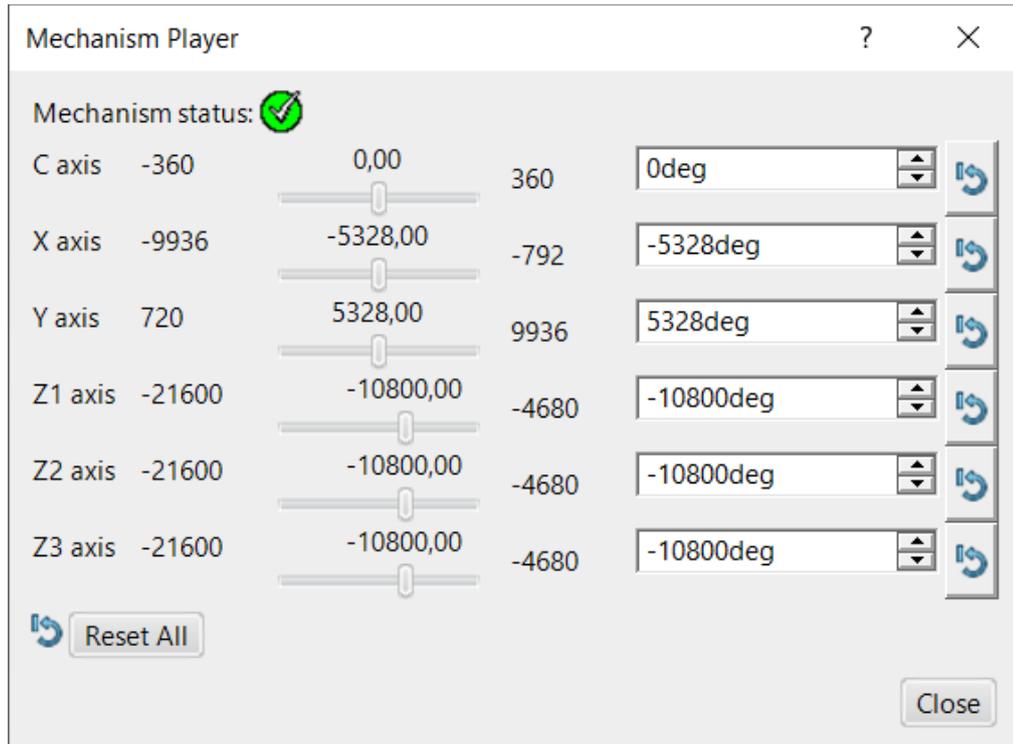


Figura 3.3 Interfaz de simulación del mecanismo

Tras computar las masas como se indica en el capítulo 2, se realizó una verificación del conjunto 3PRS+XY+C, para constatar que no existan elementos sin masa asignada y además se obtuvo la información de inercias y volumen. La figura 3.4 muestra el estado de madurez que posee el registro de masas y los porcentajes correspondientes a masas computadas y declaradas. Este porcentaje se calcula en base al número de elementos y no al volumen de estos.

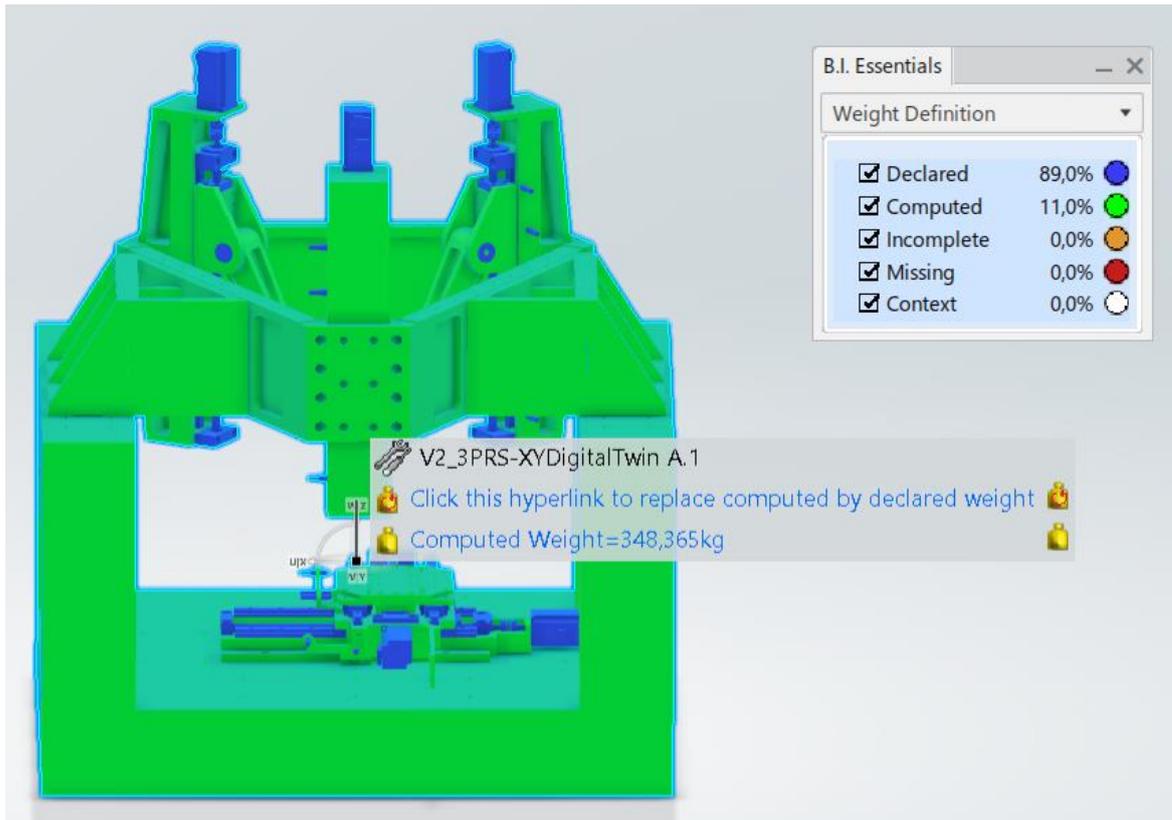


Figura 3.4 Estado de madurez del registro de pesos

La tabla 3.1 muestra los atributos principales obtenidos a partir de la definición de masas. Estos a partir de un sistema de referencia establecido en la posición que se muestra en la figura 3.4.

Tabla 3.1 Atributos del mecanismo 3PRS+XY+C

N.º	Atributo	Valor
1	Masa [tolerancia]	348,365kg [334,562kg - 362,203kg]
2	CdG sobre X	1,809mm
3	CdG sobre Y	13,239mm
4	CdG sobre Z	103,368mm
5	Volumen	0,122mm ³
6	Momento I _{xx} /G	28,669kg × m ²
7	Momento I _{yy} /G	43kg × m ²
8	Momento I _{zz} /G	38,13kg × m ²
9	Producto de Inercia I _{xy} /G	-0,011kg × m ²
10	Producto de Inercia I _{xz} /G	0,052kg × m ²
11	Producto de Inercia I _{yz} /G	0,187kg × m ²

3.2 Modelo del sistema

El modelo del sistema se realizó por bloques, cada uno contiene la estructura correspondiente para que el sistema funcione en base a los requerimientos y al análisis realizado en el capítulo 2. Este modelo se encuentra contenido en el bloque “3PRS+XY+C_Logical_Root”. La herramienta de modelamiento utilizada es el software Modelica.

3.2.1 Modelo del bloque Set_Point

Este bloque se modeló a partir de la consideración del capítulo dos que establece el accionamiento mediante las articulaciones pasivas del sistema 3PRS. Aquí se establece la posición y velocidad deseada para cada grado de libertad. Los valores de posición y velocidad son entregados mediante el bloque “kinematicPTP” mediante un arreglo matricial. La figura 3.5 muestra el diagrama de bloques que emite las doce señales de operación.

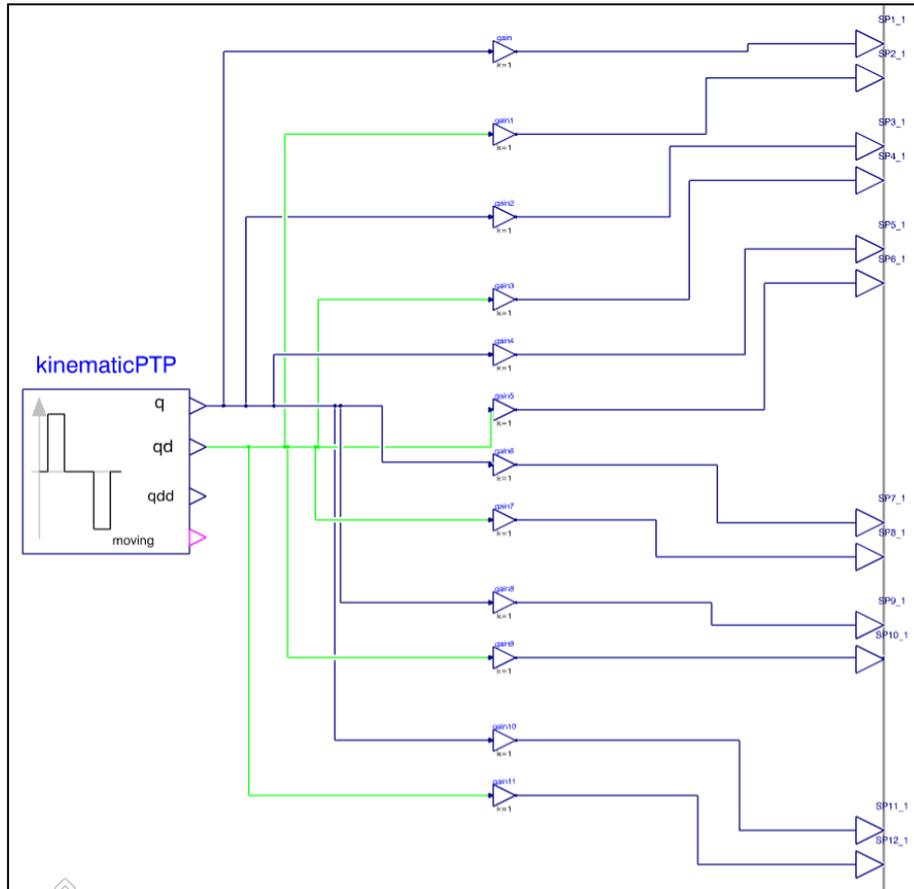


Figura 3.5 Diagrama interno del bloque Set Point

3.2.2 Modelo del bloque Dinamic_Controller

Aquí se definió la estructura del controlador de posición y velocidad para un actuador. Se realizó un control en cascada P-PI para posición y velocidad respectivamente, los parámetros de control se manejan mediante variables configuradas a partir de los requerimientos. La figura 3.6 muestra el controlador parametrizado y como entradas el punto de operación y la realimentación de las señales.

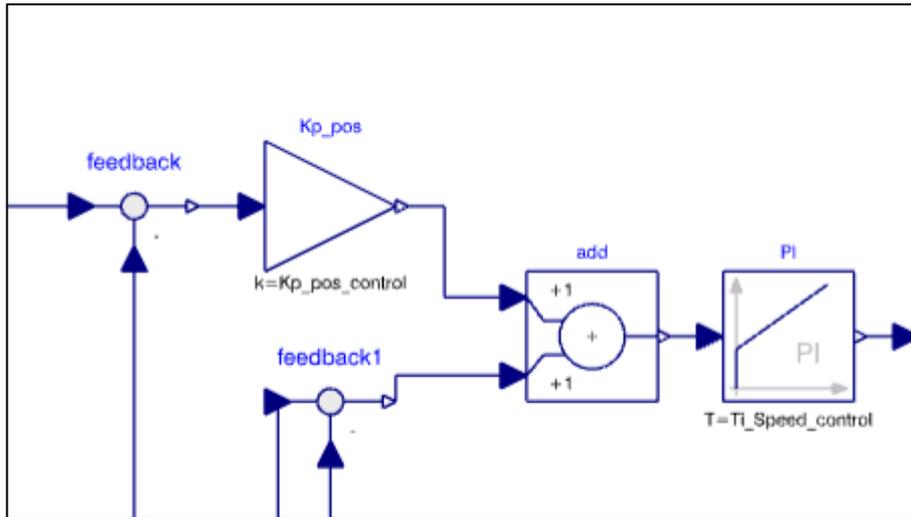


Figura 3.6 Diagrama de control dinámico

3.2.3 Diagrama del actuador

Estos bloques contienen el controlador de corriente, el modelo del circuito electromotriz y las características mecánicas del motor y su caja de engranes reductora. Se manejan las señales de potencia enviadas al motor y la corriente como realimentación al circuito de control.

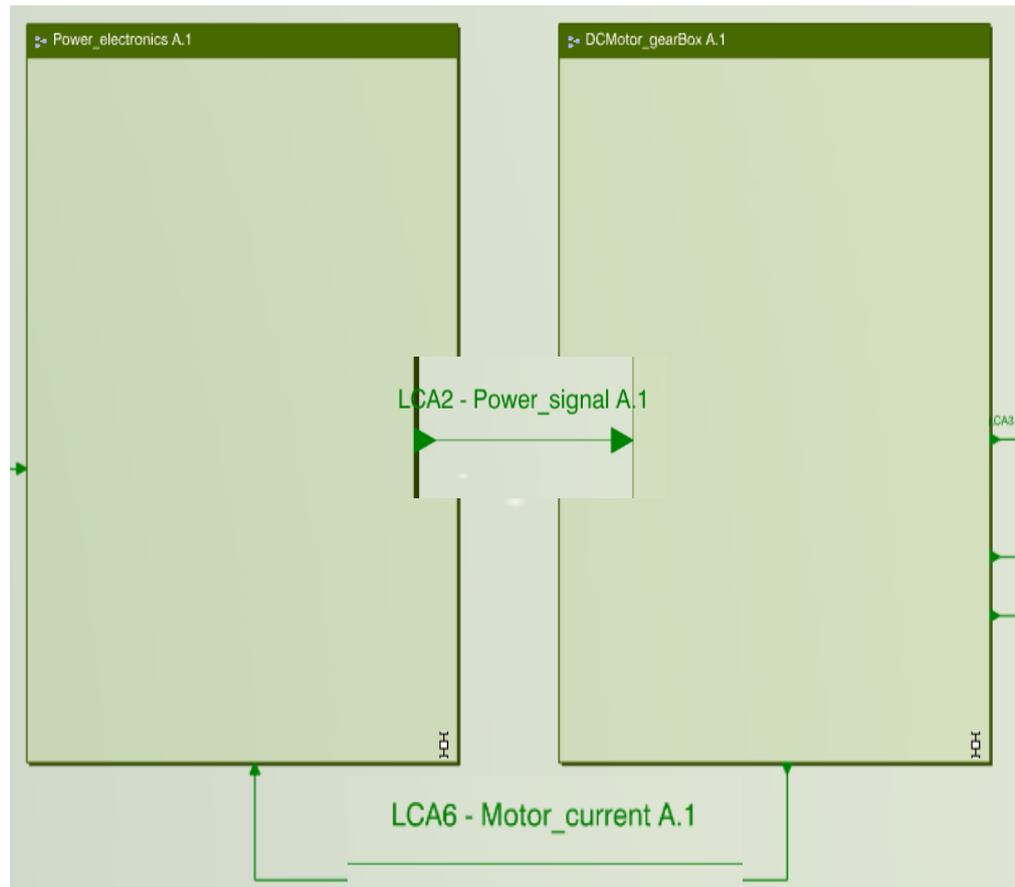


Figura 3.7 Bloques del modelo del actuador

Las señales mostradas en la figura 3.7 (con captura ampliada) son “Power_signal” que contiene la consigna de potencia enviada al actuador y “Motor_current” que realimenta el valor de corriente.

3.2.4 Modelo del bloque 3PRS+XY+C_Machine

En este bloque se describe el modelo completo del sistema mecánico (estructura, juntas y herramienta), se divide en bloques internos que contienen la información mecánica de cada parte diseñada y las conexiones ingenieriles realizadas en *CATIA-Mecanical Systems Design*. La figura 3.8 muestra las señales y el bloque que engloba el modelo mecánico del sistema.

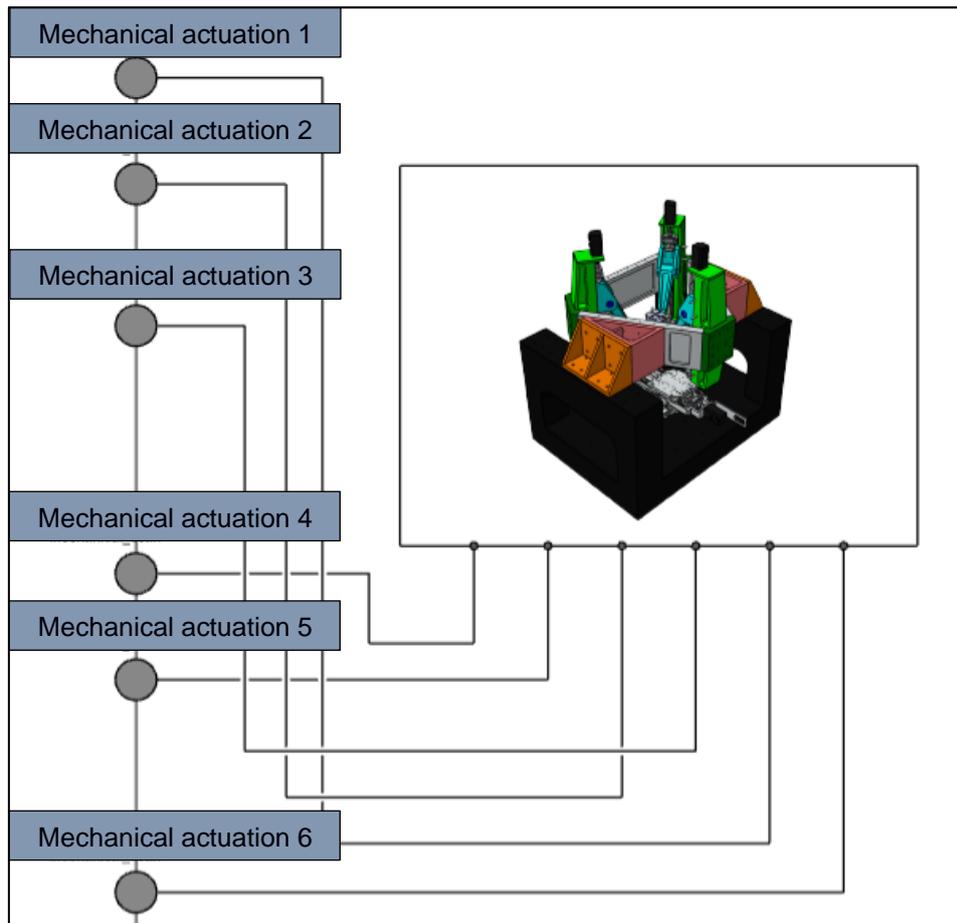


Figura 3.8 Bloque de conexión del modelo mecánico

Al modelo ingresan las seis señales procedentes de cada actuador. La figura 3.9 muestra la estructura interna del modelo mecánico. Incluye el sistema de referencia, la gravedad, los elementos modelados en 3D con sus propiedades y las juntas ingenieriles.

$$y_1 = -10710 \text{ Deg} \quad (3.4)$$

$$\Delta y = -180 \text{ Deg} \quad (3.5)$$

$$|v| = 50 \frac{\text{Deg}}{\text{s}} \quad (3.6)$$

$$|a| = 25 \frac{\text{Deg}}{\text{s}^2} \quad (3.7)$$

La figura 3.10 muestra la señal de posición entregada como referencia, la cual ha sido simulada durante seis segundos.

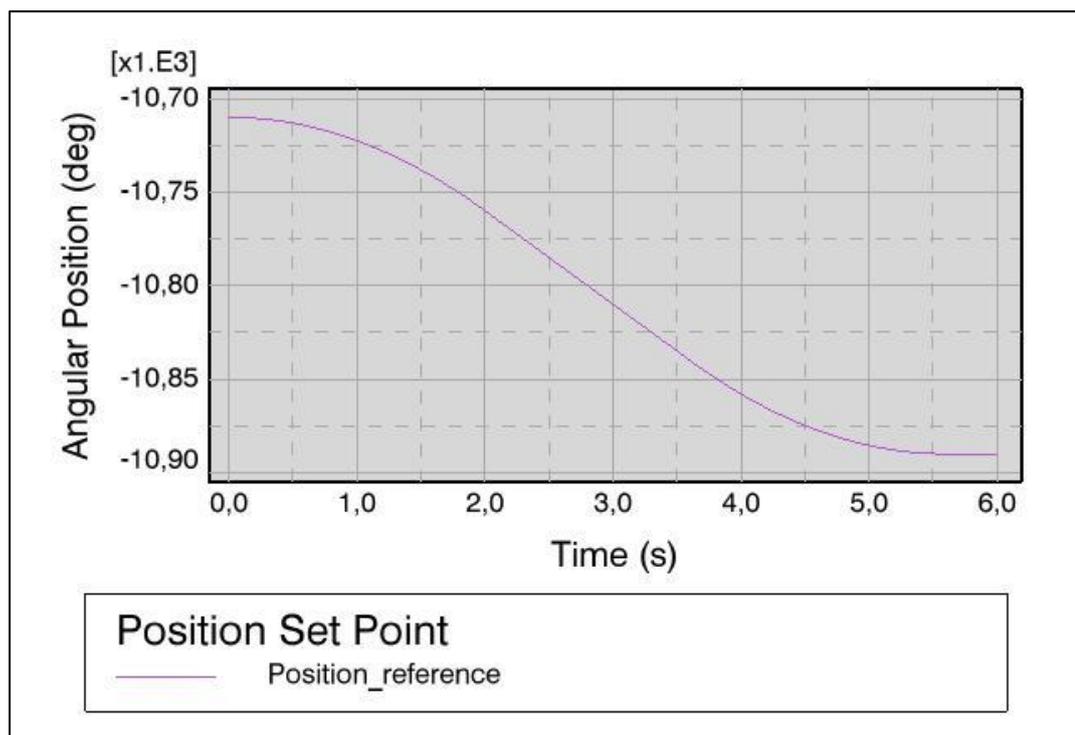


Figura 3.10 Señal de referencia de posición

La figura 3.11 muestra la señal de velocidad entregada como referencia, en la que se puede observar la incidencia de la aceleración.

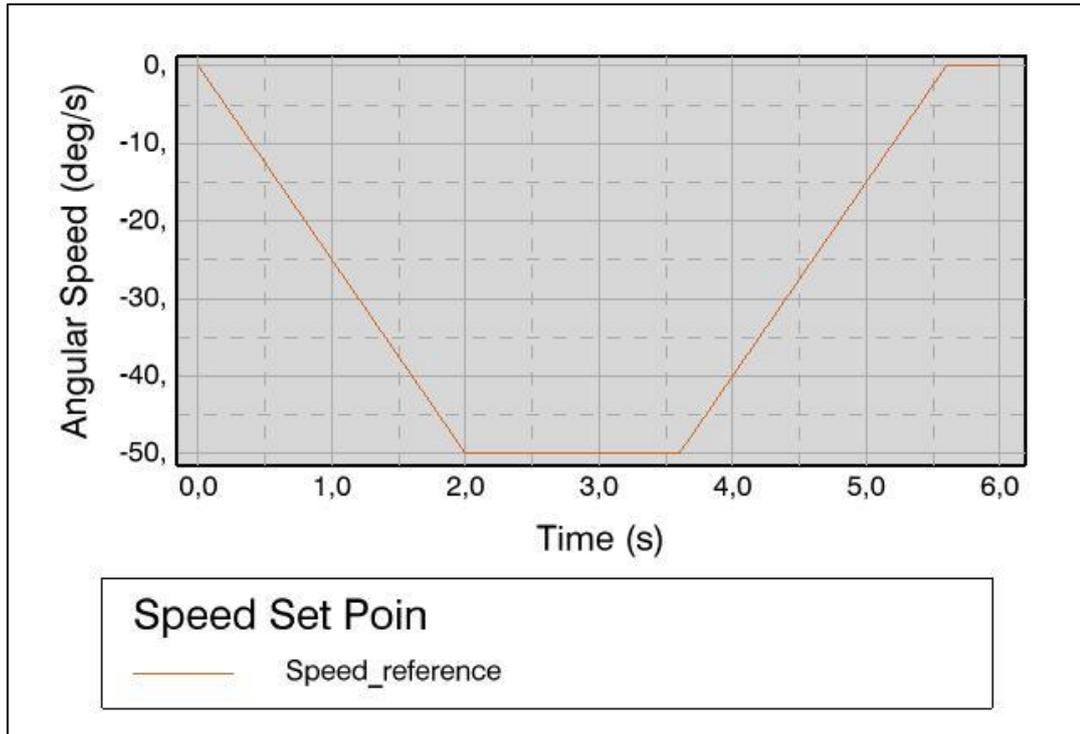


Figura 3.11 Señal de referencia de velocidad

El sistema presenta una respuesta de posición que se ajusta a la señal de referencia, demostrando el correcto funcionamiento del controlado. La figura 3.12 muestra la señal de posición medida para los primeros 300 milisegundos. Le toma 100 milisegundos al sistema conseguir la consigna del escalón como indica el marcador que se encuentra dos por ciento por debajo de la consigna. El comportamiento es sobre amortiguado. La figura 3.13 muestra la respuesta completa de seis segundos.

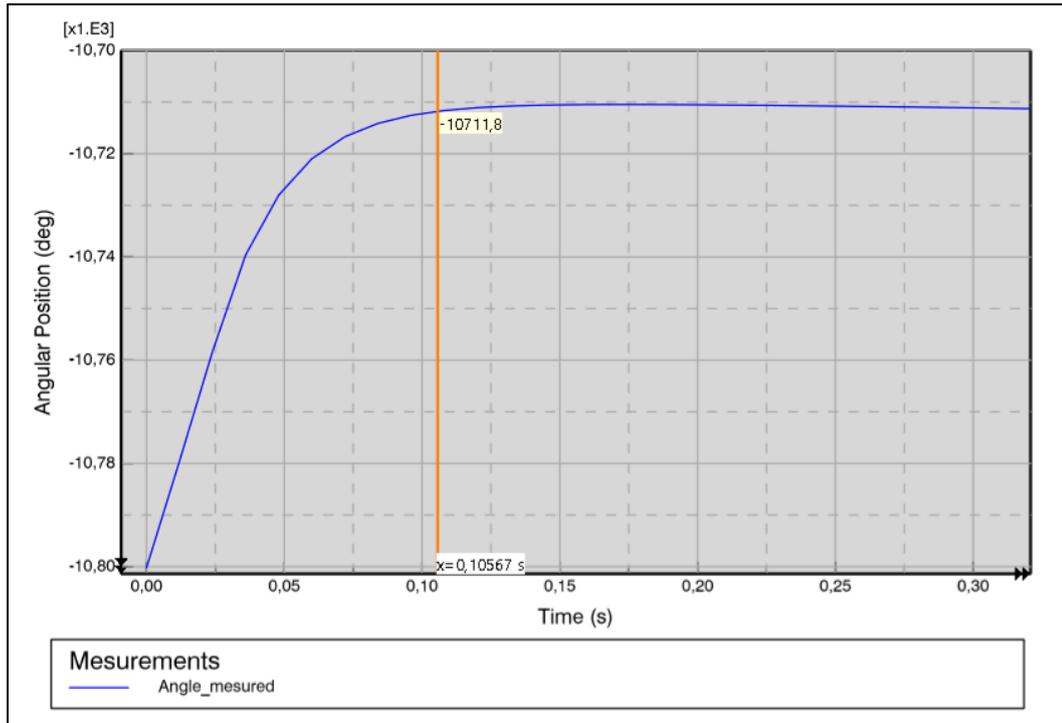


Figura 3.12 Respuesta de posición medida para el arranque

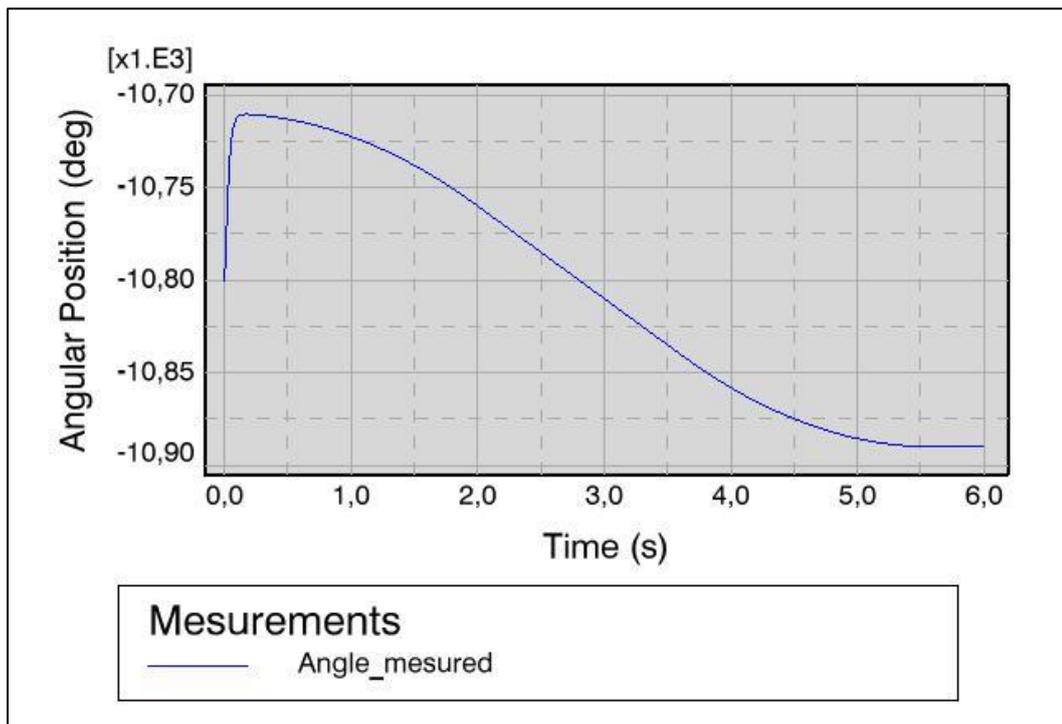


Figura 3.13 Respuesta de posición medida

La respuesta medida de velocidad se muestra en la figura 3.14, presenta un pico inicial que corresponde al comportamiento para el cumplimiento del impulso aplicado. Posterior a esto el sistema se estabiliza en la consigna de velocidad dada como referencia. La figura 3.15 muestra un acercamiento en el que se puede apreciar la semejanza con el comportamiento de la figura 3.11.

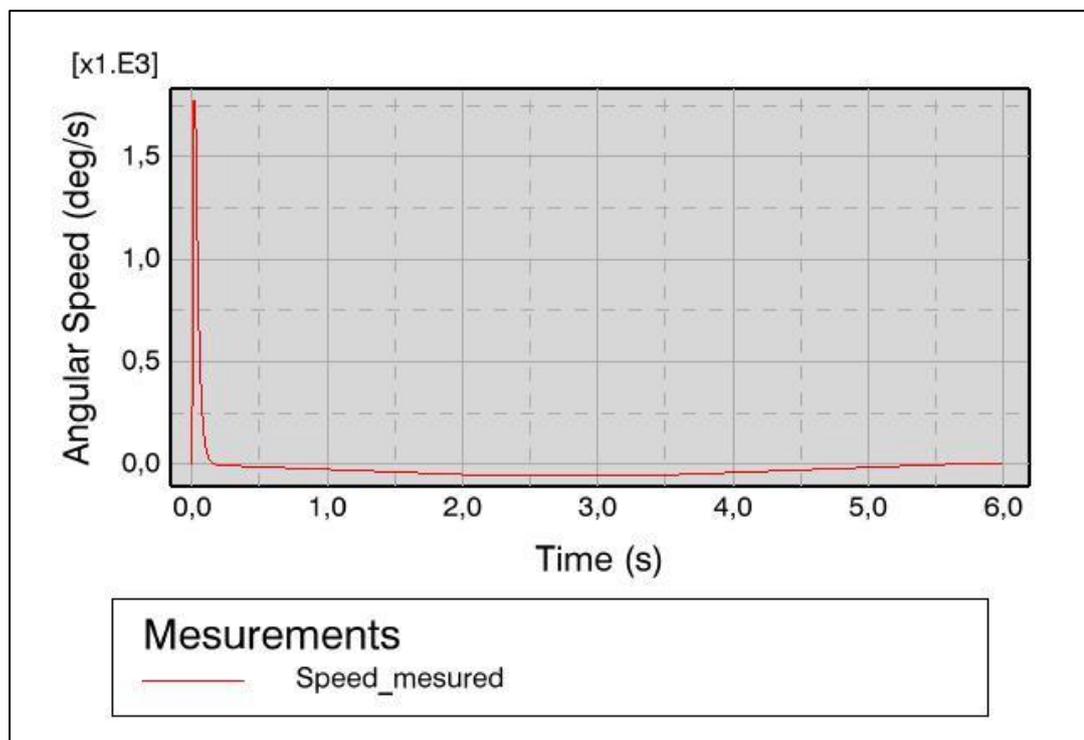


Figura 3.14 Respuesta de velocidad medida

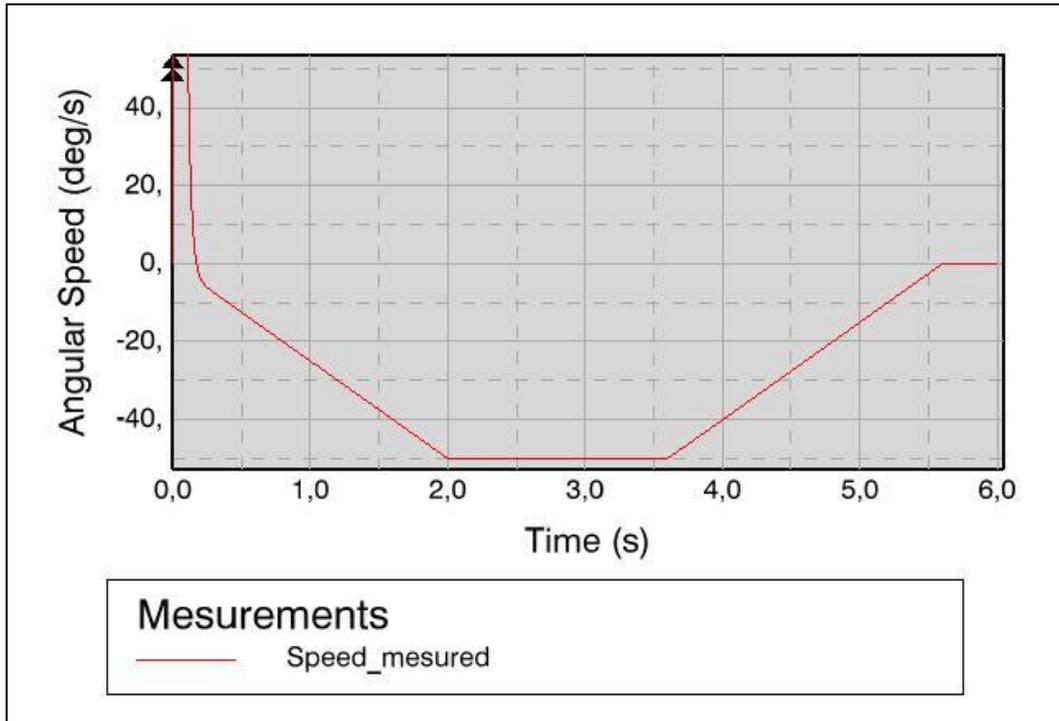


Figura 3.15 Respuesta de velocidad medida para $t > 100$ ms

El controlador que se diseñó posee tres parámetros configurables que son una ganancia proporcional de posición, un ganancia proporcional de velocidad y una constante de tiempo de integración. Estos fueron parametrizados como se muestra en la imagen 3.6, de modo que sus valores puedan ser modificados desde la sección de requerimientos. Las imágenes 3.16 y 3.17 muestran las ventanas de configuración de parámetros para los bloques de control “Kp_pos” y “PI” respectivamente.

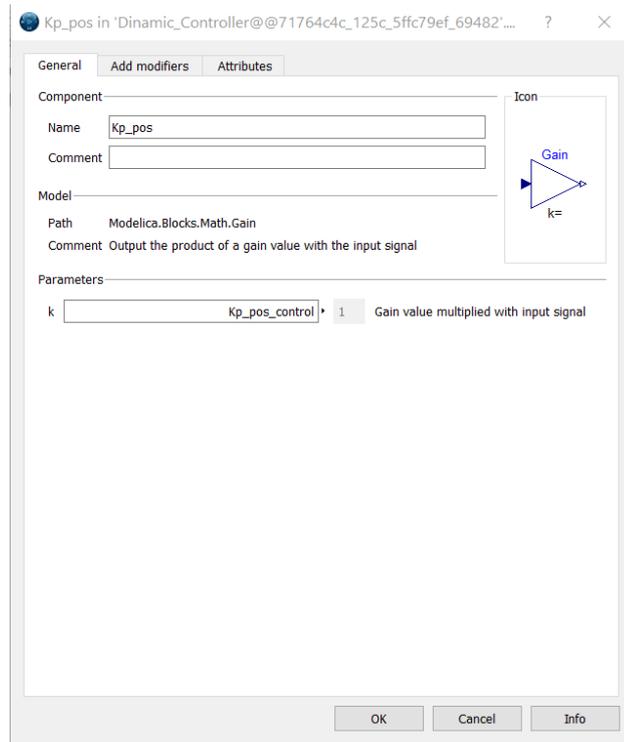


Figura 3.16 Ventana de configuración del controlador de posición

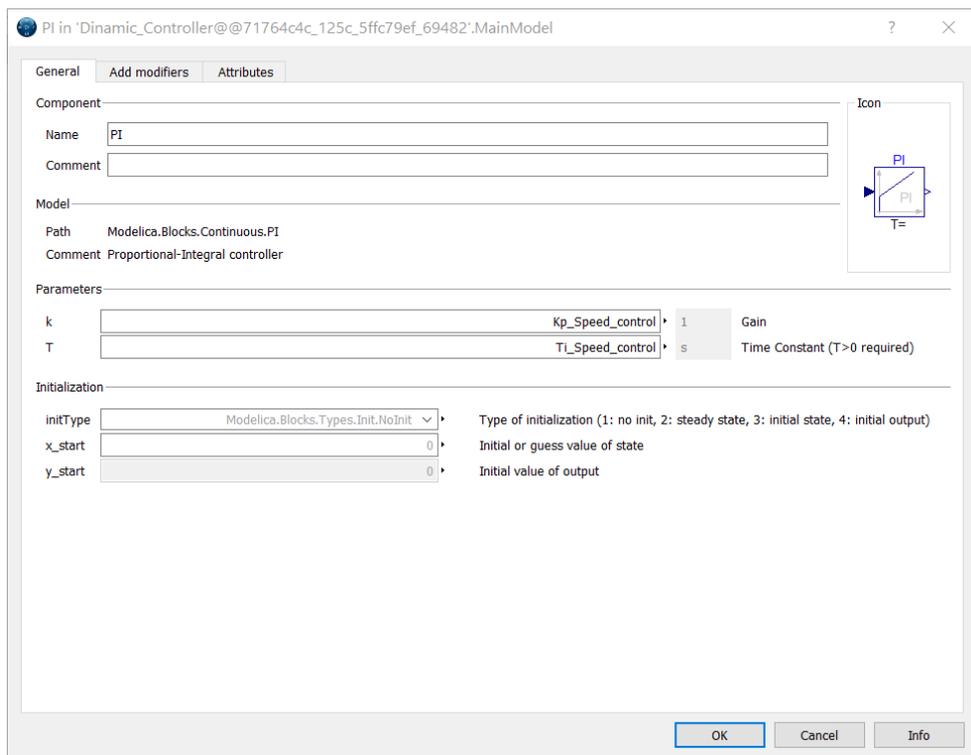


Figura 3.17 Ventana de configuración del controlador de velocidad

La figura 3.18 muestra en el recuadro de color rojo las variables de parámetros definidas para poder ser usadas en los bloques mostrados en las dos figuras anteriores. Estos sirven de conexión entre los requerimientos y los diagramas de bloques. En el recuadro de color azul se muestra el valor obtenido para cada parámetro mediante la relación. El recuadro amarillo muestra las relaciones establecidas para la vinculación entre los parámetros y los requerimientos.

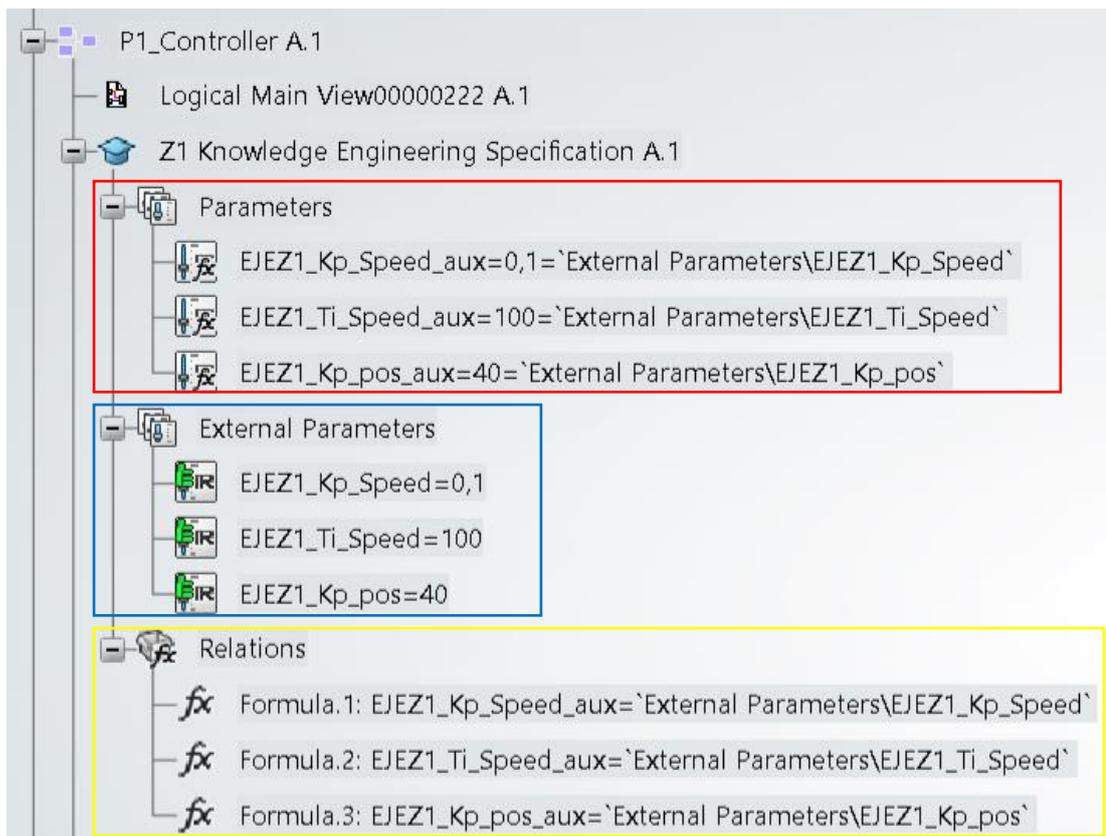


Figura 3.18 Parámetros de control y relaciones externas

La figura 3.19 muestra los parámetros de control y sus rangos de valores registrados en el gestor de proyectos de Enovia. El recuadro en color verde los señala.

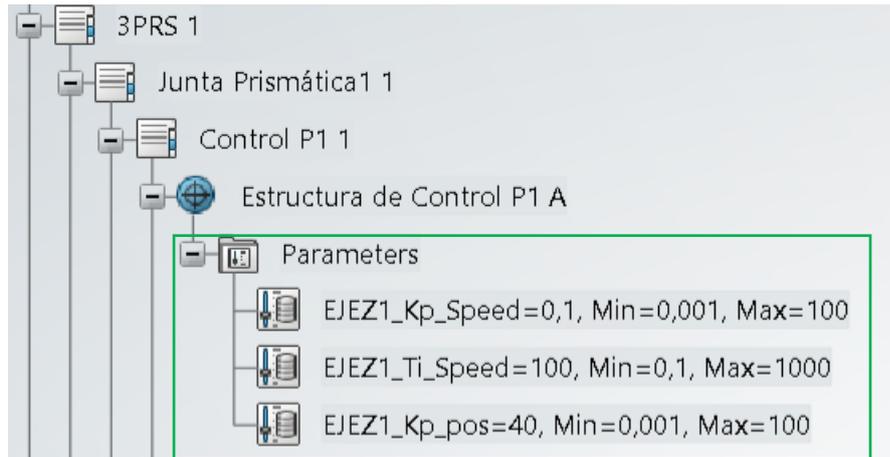


Figura 3.19 Parámetros de control definidos en los requerimientos

3.4 Análisis de costos

El modelo de negocio establecido para este proyecto es el de tipo servicio, por lo que fue requerido un análisis de ingresos y gastos para el diseño de un gemelo digital. Producto de este análisis se puede establecer un costo por hora del servicio.

Los egresos requeridos para el diseño de este proyecto se generan por el uso del conjunto de software de *Dassault Systèmes* y su plataforma 3DEXperience. El uso de estos elementos consta de una parte a manera de inversión inicial y una tasa periódica de renovación.

Dassault Systèmes agrupa sus aplicaciones a manera de roles. Los roles necesarios para el desarrollo de este proyecto se muestran en la Figura 3.20.

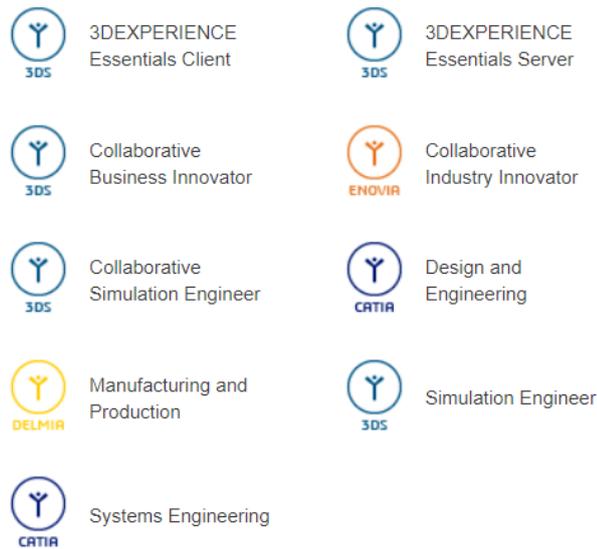


Figura 3.20 Roles utilizados en el proyecto

Estos roles incluyen herramientas para el diseño de partes y conexiones ingenieriles, simulación para ingeniería, diseño de modelos para sistemas y herramientas de colaboración y base de datos.

La inversión inicial se divide en dos rubros. La inversión inicial para el uso de 3DEXPERIENCE por un monto de \$10.000,00 y la inversión inicial por los roles mostrados en la figura 3.1 por un monto de \$5.000,00. Tanto para el uso de 3DEXPERIENCE como el de los roles se requiere una tasa anual de renovación. La tasa de 3DEXPERIENCE es de \$4.500,00 y la tasa por los roles es de \$900,00.

Estos rubros no pueden ser recuperados en un único proyecto pues esto inflaría el precio sin sentido alguno, es así como se los ha dividido de acuerdo con modelo de negocio.

La recuperación de esta inversión inicial fue amortizada a 3 años basándose en la tendencia a seguir por este tipo de negocio[46]. A partir de esto el gasto de software por inversión es de \$2.08 la hora de trabajo. La tasa anual de uso genera un gasto por hora de trabajo de \$2.25. El gasto de honorarios por hora para un ingeniero mecatrónico ha sido deducido a partir de cuánto cobra un ingeniero mecánico en promedio por nomina en Ecuador [47] considerando así un valor de \$4,00 la hora. Adicional a esto se agrega un valor de \$4,00 que estipula la ganancia del negocio de

acuerdo con la tendencia de mercado. La tabla 3.1 recapitula un detalle de los rubros de costo para el proyecto y en su última columna define el precio del servicio por hora de trabajo.

Tabla 3.2 Detalle de costos por hora para el cálculo del precio del servicio

N.º	Descripción	Valor monetario (\$) por hora
1	Inversión Inicial amortizada a 3 años	\$ 2,08
2	Tasas anuales por uso de software	\$ 2,25
3	Honorarios de diseñador	\$ 4,00
4	Ganancia del negocio	\$ 4,00
	Precio total del servicio	\$ 12,33

Como resultado de este análisis se tiene un precio al público de \$12,33 la hora para el diseño de gemelos digitales aplicables a industria 4.0. Para este proyecto se ha invertido en tiempo de diseño 150 horas por lo que el precio de este proyecto suma el total de \$1849,50.

CAPÍTULO 4

4. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- El gemelo digital diseñado cumple con los requerimientos de semejanza a la máquina real 3PRS+XY+C; éste ha sido construido respetando el modo de accionamiento del sistema real, sus características, propiedades y comportamientos.
- La alternativa escogida como solución (3DExperience) cumple con las herramientas necesarias para la creación de un modelo digital: manejo de base de datos para requerimientos, diseño de partes, creación de ensamblajes, diseño de modelos de comportamiento, simulación y obtención de gráficas características.
- El ensamble realizado cumple con las características de movimiento y restringe las singularidades del mecanismo 3PRS, posee 6 grados de libertad de acuerdo con los requerimientos y es accionado en sus juntas pasivas respetando el análisis dinámico.
- La representación mecánica configurada y el conjunto de modelos diseñados permiten la simulación dinámica en conjunto del sistema a partir de una entrada de posición y velocidad establecida como punto de operación.
- La relación de parámetros entre requerimientos y modelo permiten realizar cambios sobre el gemelo digital desde el gestor Enovia.

4.2 Recomendaciones

- Para la realización de gemelos digitales hay que partir siempre de un diseño funcional y de elementos necesarios para la conformación de la estructura lógica.
- Considerar las juntas sobre las que se acciona el mecanismo para la realización del modelo mecánico.
- Configurar la representación mecánica de modo que el accionamiento coincida con el modelo de comportamiento procedente del actuador.
- El diseño de este gemelo digital puede ser usado para la creación de modelos distintos gracias a su diseño modular.
- Implementar un sistema de base de datos en la nube para el manejo de la data procedente del sistema físico.
- Implementar un sistema de detección de fallos por inteligencia artificial que resulte en un sistema de mantenimiento predictivo para los equipos.
- Implementar un sistema de producción en base a demanda para evitar el uso excesivo de almacenamiento, mediante estrategias de predicción de comportamiento del mercado con inteligencia artificial.

BIBLIOGRAFÍA

- [1] “Romanos 11:36 Porque todas las cosas proceden de él, y existen por él y para él. ¡A él sea la gloria por siempre! Amén. | Nueva Versión Internacional - Español (NVI) | Descargar la Biblia App ahora.” <https://www.bible.com/es/bible/128/rom.11.36> (accessed Feb. 05, 2021).
- [2] J. Vishal, “Define Metal Cutting - Mechanical Site,” 2019. <https://www.mechanicalsite.com/metal-cutting> (accessed Nov. 05, 2020).
- [3] “Demain des usines pilotées par internet - Challenges.” https://www.challenges.fr/high-tech/demain-des-usines-pilotees-par-internet_7180 (accessed Nov. 05, 2020).
- [4] M. Fajardo-Pruna, L. López-Estrada, H. Pérez, E. Diez, and A. Vizán, “Analysis of a single-edge micro cutting process in a hybrid parallel-serial machine tool,” *Int. J. Mech. Sci.*, vol. 151, no. August 2018, pp. 222–235, 2019, doi: 10.1016/j.ijmecsci.2018.11.023.
- [5] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, “The industrial internet of things (IIoT): An analysis framework,” *Comput. Ind.*, vol. 101, pp. 1–12, Oct. 2018, doi: 10.1016/j.compind.2018.04.015.
- [6] S. Vaidya, P. Ambad, and S. Bhosle, “Industry 4.0 - A Glimpse,” in *Procedia Manufacturing*, Jan. 2018, vol. 20, pp. 233–238, doi: 10.1016/j.promfg.2018.02.034.
- [7] “What is the 3DEXPERIENCE platform? | Driving Digital Transformation in Mining.” <https://blogs.3ds.com/geovia/what-is-the-3dexperience-platform/> (accessed Feb. 22, 2021).
- [8] J. Lee, H. Davari, J. Singh, and V. Pandhare, “Industrial Artificial Intelligence for industry 4.0-based manufacturing systems,” *Manuf. Lett.*, vol. 18, pp. 20–23, Oct. 2018, doi: 10.1016/j.mfglet.2018.09.002.
- [9] S. H. Choi and A. M. M. Chan, “A virtual prototyping system for rapid product development,” *CAD Comput. Aided Des.*, vol. 36, no. 5, pp. 401–412, Apr. 2004, doi: 10.1016/S0010-4485(03)00110-6.
- [10] F. Dai, W. Felger, T. Fruhauf, M. Gobel, D. Reiners, and G. Zachmann, “Virtual Prototyping Examples for Automotive Industries,” 2001. https://www.researchgate.net/publication/2362802_Virtual_Prototyping_Examples

_for_Automotive_Industries (accessed Nov. 05, 2020).

- [11] Y. Altintas, P. Kersting, D. Biermann, E. Budak, B. Denkena, and I. Lazoglu, "Virtual process systems for part machining operations," *CIRP Ann. - Manuf. Technol.*, vol. 63, no. 2, pp. 585–605, Jan. 2014, doi: 10.1016/j.cirp.2014.05.007.
- [12] L. López-Estrada, M. Fajardo-Pruna, L. Sánchez-González, H. Pérez, L. Fernández-Robles, and A. Vizán, "Design and implementation of a stereo vision system on an innovative 6DOF single-edge machining device for tool tip localization and path correction," *Sensors (Switzerland)*, vol. 18, no. 9, 2018, doi: 10.3390/s18093132.
- [13] J. Kapás, "Industrial revolutions and the evolution of the firm's organization: an historical perspective," *J. Innov. Econ. Manag.*, vol. n° 2, no. 2, p. 15, 2008, doi: 10.3917/jie.002.0015.
- [14] A. Stăncioiu, "THE FOURTH INDUSTRIAL REVOLUTION 'INDUSTRY 4.0.'"
- [15] M. A. Pisching, A. A. Tasca, M. A. O. Pessoa, F. Junqueira, and P. E. Miyagi, "ARQUITETURA PARA DESENVOLVIMENTO DE SISTEMAS CIBER-FÍSICOS APLICADOS NA INDÚSTRIA INDÚSTRIA 4.0."
- [16] L. Baque Mite, M. Mariela Cadena Vásquez, D. Moserratte Salvatierra, and K. Alexander Quiñonez Ponce, "Análisis de los niveles de automatización de los procesos de producción de cacao Analysis of the levels of automation of cocoa production processes," *J. Bus. Entrep. Julio-diciembre*, vol. 4, no. 2, Jul. 2020, doi: 10.37956/jbes.v4i2.70.
- [17] S. Julio and N. Solano, *AUTOMATIZACIÓN DE LOS PROCESOS DE SECADO Y SELECCIÓN DEL CACAO ECUATORIANO CONSERVANDO LAS CARACTERÍSTICAS SENSORIALES Y PREVINIENDO RIESGOS LABORALES*. 2017.
- [18] M. Angel and A. Medina, "DISEÑO DEL SISTEMA SCADA DEL PROCESO DE TRANSPORTE DE CEMENTO PARA HOLCIM-ECUADOR," 2006.
- [19] C. A. Lara Pacheco, "Plan estratégico para una empresa de automatización industrial caso: Intelware S.A.," 2018, Accessed: Nov. 05, 2020. [Online]. Available: <http://repositorio.puce.edu.ec:80/xmlui/handle/22000/15441>.
- [20] S. J. Oks, M. Jalowski, A. Fritzsche, and K. M. Möslin, "Cyber-physical modeling and simulation: A reference architecture for designing demonstrators for industrial cyber-physical systems," in *Procedia CIRP*, Jan. 2019, vol. 84, pp. 257–264, doi:

- 10.1016/j.procir.2019.04.239.
- [21] J. Lee, B. Bagheri, and H. A. Kao, "A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems," *Manuf. Lett.*, vol. 3, pp. 18–23, Jan. 2015, doi: 10.1016/j.mfglet.2014.12.001.
- [22] "Industrial Internet of Things." <https://www.qrapp.org.uk/iot-industrial> (accessed Feb. 01, 2021).
- [23] "Los cambios que trae la industria 4.0." <https://blog.wearedrew.co/los-cambios-que-trae-la-industria-4.0> (accessed Feb. 01, 2021).
- [24] L.-V. Israel, L.-J. Ismael, C. Castuera Jorge, O. Rivera Arquímedes, and P. Cabrera Mario, "Hacia la Integración de un Sistema Inteligente de Manufactura: Consideraciones y Experimentos."
- [25] "Innovación colaborativa | ENOVIA – Dassault Systèmes." <https://www.3ds.com/es/productos-y-servicios/enovia/> (accessed Feb. 01, 2021).
- [26] "CATIA Disciplinas - Dassault Systèmes®." <https://www.3ds.com/es/productos-y-servicios/catia/disciplinas/engineering/> (accessed Feb. 01, 2021).
- [27] "The Modelica Association — Modelica Association." <https://www.modelica.org/> (accessed Feb. 01, 2021).
- [28] "What is 3DEXPERIENCE® platform by Dassault Systemes? -." <https://www.dte.co.uk/products/what-is-3dexperience-platform/> (accessed Nov. 05, 2020).
- [29] O. L. de Weck, "Fundamentals of Systems Engineering Session 2 Requirements Definition," in *MIT OpenCourseWare*, 2015, p. 19, Accessed: Dec. 17, 2020. [Online]. Available: https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-842-fundamentals-of-systems-engineering-fall-2015/lecture-notes/MIT16_842F15_Ses2_Req.pdf.
- [30] J. M. Llata García, "PSE-FCM: Functional Design for Servo-Systems," *Peer Learning EXPERIENCE-Dassault Systèmes*, 2017. <https://peerlearningexperience.3ds.com/mod/scorm/player.php> (accessed Dec. 17, 2020).
- [31] J. M. Llata García, "PSE-FCM: Logical Design for Servo-Systems," *Peer Learning EXPERIENCE-Dassault Systèmes*, 2017. <https://peerlearningexperience.3ds.com/mod/scorm/player.php> (accessed Dec. 17, 2020).

- [32] IKO-THOMPSON LTD, *Linear Motion Rolling Guide Series General Catalog IKO-THOMPSON LTD.*, vol. 06. 2012.
- [33] NSK, “NSK Standard Ball Screws: Compact FA Series,” no. 800, 2012, [Online]. Available: https://stevenengineering.com/Tech_Support/PDFs/NSK_BALL-SCREW-FA.pdf.
- [34] Hephaist Seiko Co, “Spherical Rolling Joint,” pp. 2–3, 2013, [Online]. Available: <http://www.myostat.ca/SRJoint#>.
- [35] M. Sapti, *Super Precision Bearings*, vol. 53, no. 9. 2019.
- [36] NSK, *Rodamientos de bolas de contacto angular*. .
- [37] Ruland, “PCMR25-8-6-A,” pp. 6–7.
- [38] T. M. Control, “Harmonic Drive actuator.”
- [39] E. C. Mitsubishi, “Servo Motor HF-MP Series General-Purpose AC Servo MELSERVO-J3,” pp. 1–4, 2012, [Online]. Available: <https://www.koninghartman.nl/UserFiles/Product/Datasheet/MR-HF-MP.pdf>.
- [40] Omron, “E2B PROXIMITY SENSORS,” 2013.
- [41] Kistler, *3-Component Force Sensor*. 2010.
- [42] Y. Li and Q. Xu, “Kinematics and inverse dynamics analysis for a general 3-PRS spatial parallel mechanism,” *Robotica*, vol. 23, no. 2, pp. 219–229, 2005, doi: 10.1017/S0263574704000797.
- [43] M. Díaz-Rodríguez *et al.*, “Kinematics and Inverse Dynamic Model of a 3-PRS Paralel Robot CinemáticaCinem´Cinemática y DinámicaDin´Dinámica Inversa de un Robot Paralelo 3-PRS.” Accessed: Jan. 31, 2021. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-01538627>.
- [44] S. Provenzano, “Resolución del problema dinámico directo de cadenas abiertas ...,” *Universidad de los Andes Revista Científica*, p. 2002.
- [45] W. Khalil and O. Ibrahim, “General solution for the dynamic modeling of parallel robots,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2004, vol. 2004, no. 4, pp. 3665–3670, doi: 10.1109/robot.2004.1308828.
- [46] Franquicias y Negocios, “¿Cuándo se recupera la inversión?” <https://www.infofranquicias.com/cd-7953/Cuando-se-recupera-la-inversion.aspx> (accessed Feb. 01, 2021).
- [47] Cuanto Cobro, “Cuanto Cobra un Ingeniero Mecánico en Ecuador.”

http://ecuador.cuanticobro.com/?q=ingeniero_mecanico&start=2720 (accessed Feb. 01, 2021).

APÉNDICES

APÉNDICE A

Bloques del Modelo de Comportamiento

En este apéndice se describen los principales bloques que conforman el modelo de comportamientos realizado para la máquina 3PRS+XY+C. Se incluyen imágenes con información adicional y código Modelica que permita comprender el comportamiento de los bloques.

A.1 Bloque KinematicPTP2

Este bloque es el generador de función para la prueba realizada en el capítulo tres. Genera una señal de acuerdo con las restricciones cinemáticas de posición, velocidad y aceleración dadas, además del tiempo de inicio. Las figuras Figura A.1.1 y Figura A.1.2 muestran información más detallada, parámetros y conectores. La Figura A.1.3 muestra el código Acausal de Modelica que describe el funcionamiento.

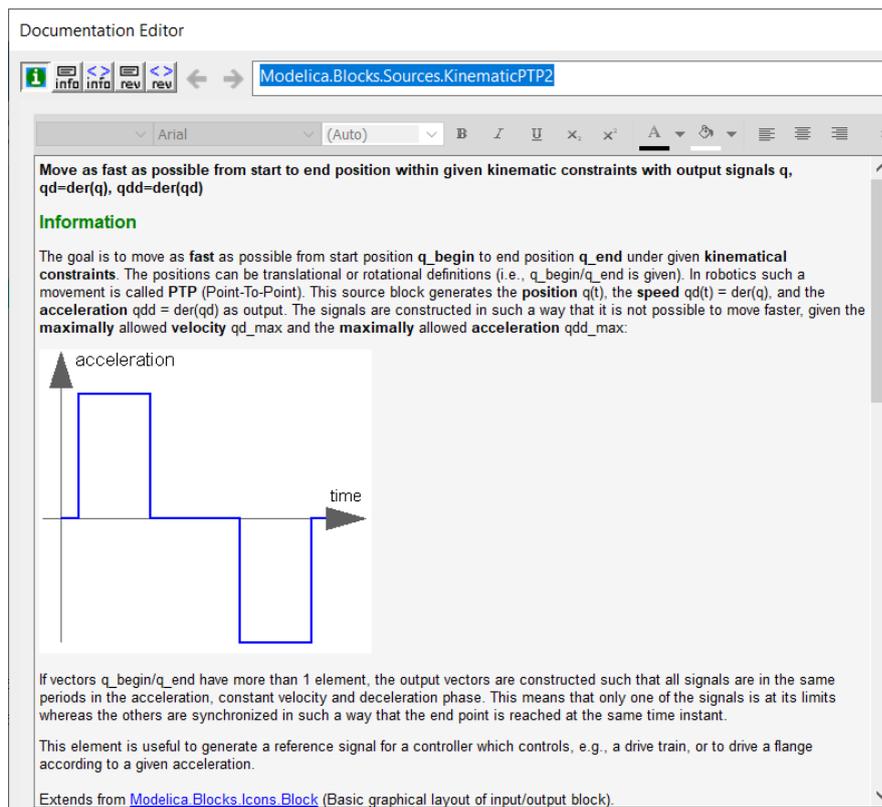


Figura A.1 Descripción 1 del bloque KinematicPTP2

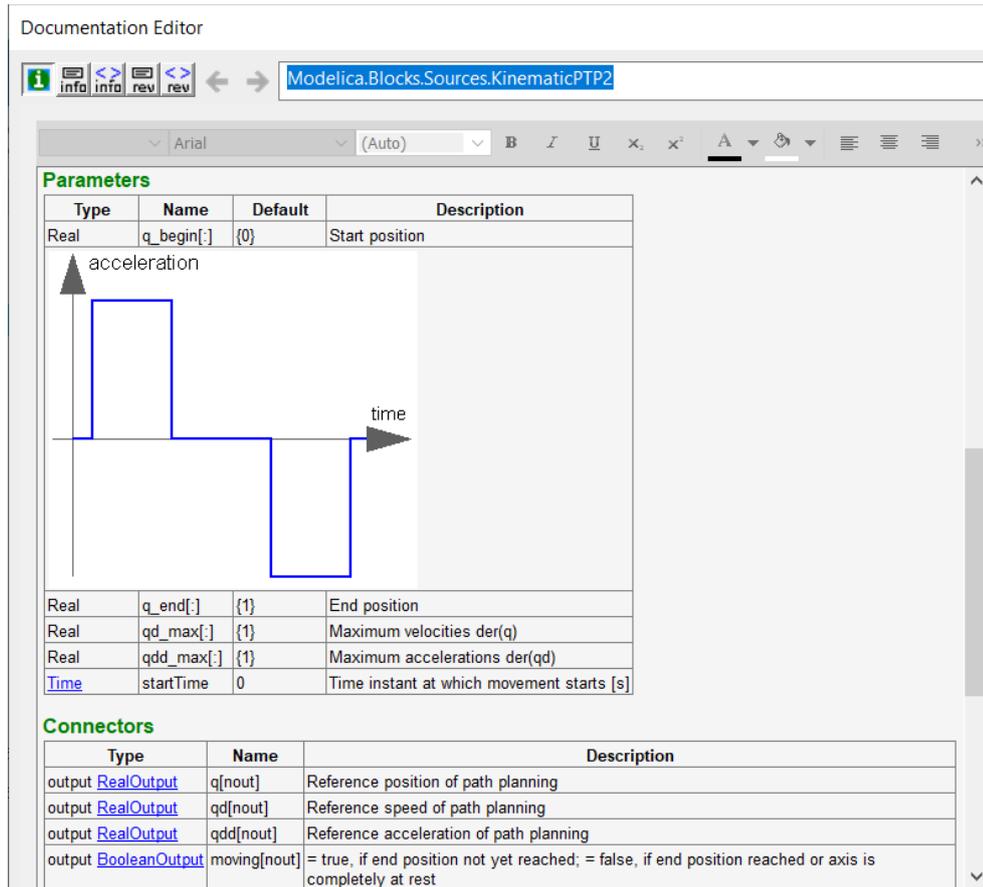


Figura A.2 Descripción 2 del bloque KinematicPTP2

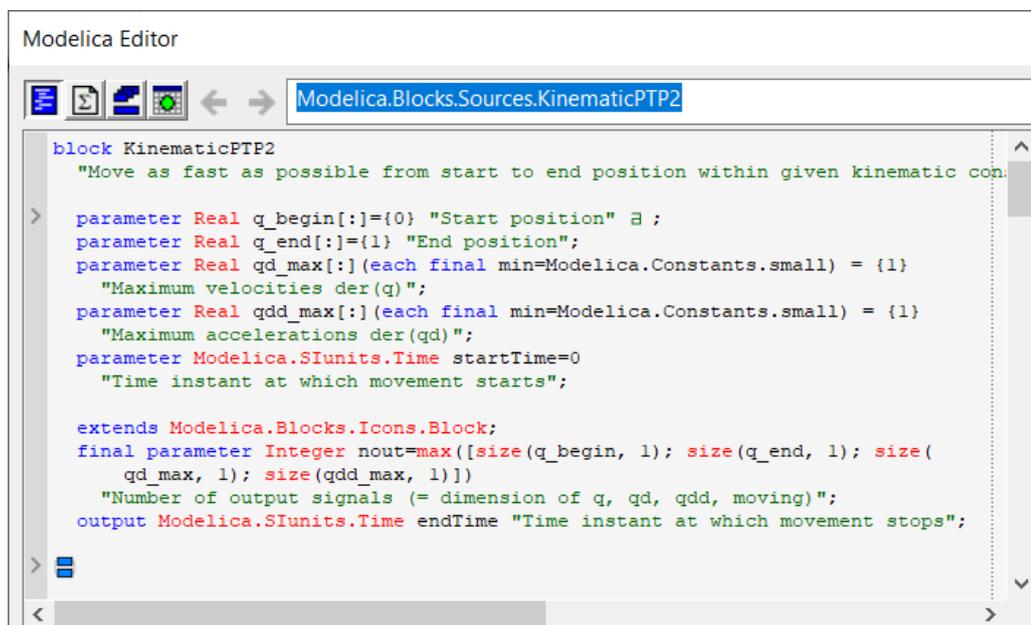
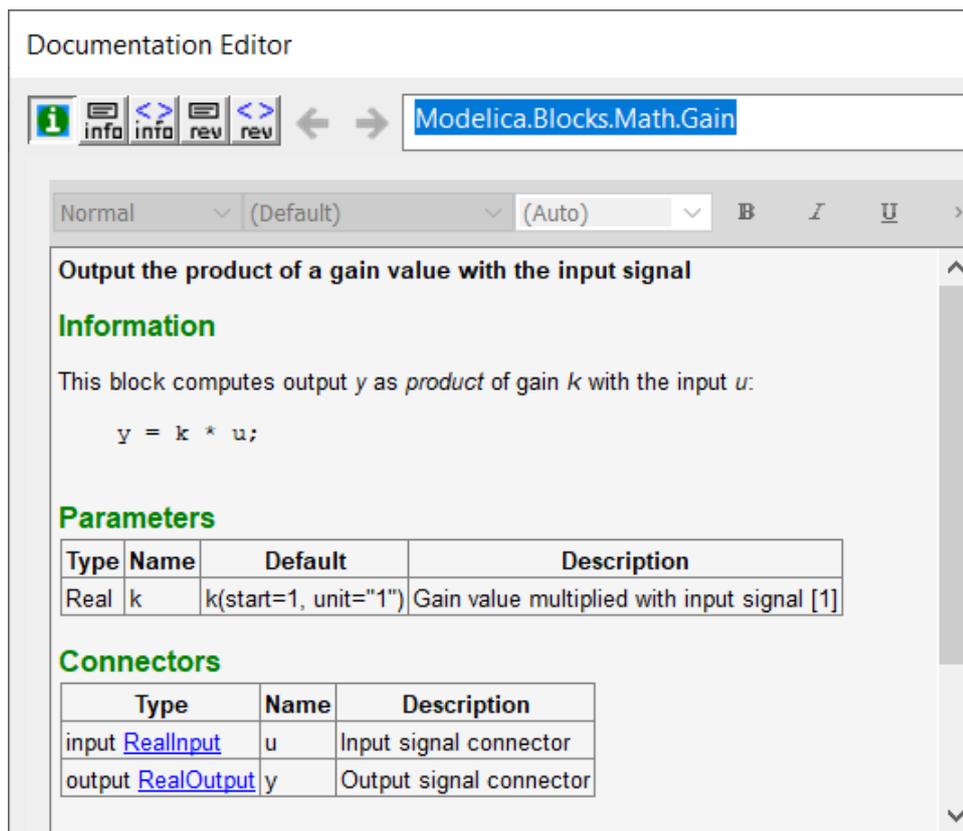


Figura A.3 Programación en Modelica del bloque KinematicPTP2

A.2 Bloque Gain

Este es un bloque que permite la multiplicación de una señal por un valor real y devuelve una salida producto de esta operación. Fue utilizado para configurar la ganancia proporcional del controlador de posición. La figura A.4 muestra la información, parámetros y conectores necesarios para cumplir con el comportamiento mostrado en la figura A.5.



Documentation Editor

Modelica.Blocks.Math.Gain

Normal (Default) (Auto) B I U >>

Output the product of a gain value with the input signal

Information

This block computes output y as *product* of gain k with the input u :

$$y = k * u;$$

Parameters

Type	Name	Default	Description
Real	k	k(start=1, unit="1")	Gain value multiplied with input signal [1]

Connectors

Type	Name	Description
input RealInput	u	Input signal connector
output RealOutput	y	Output signal connector

Figura A.4 Descripción del bloque Gain

```

Modelica Editor
Modelica.Blocks.Math.Gain

block Gain "Output the product of a gain value with the input signal"
  parameter Real k(start=1, unit="1") "Gain value multiplied with input signal";
  equation
    y = k*u;
  end Gain;

```

Figura A.5 Programación en Modelica del bloque Gain

A.3 Bloque Feedback

Este bloque solo recibe parámetros mediante los conectores, recibe dos señales para devolver su diferencia en una tercera. La figura A.6 muestra la información y los conectores usados para el comportamiento definido mostrado en la figura A.7.

Documentation Editor
Modelica.Blocks.Math.Feedback

Normal (Default) (Auto) B I U x₂ x² A >>

Output difference between commanded and feedback input

Information

This blocks computes output y as *difference* of the commanded input $u1$ and the feedback input $u2$:

$$y = u1 - u2;$$

Example:

```
parameter:  n = 2
```

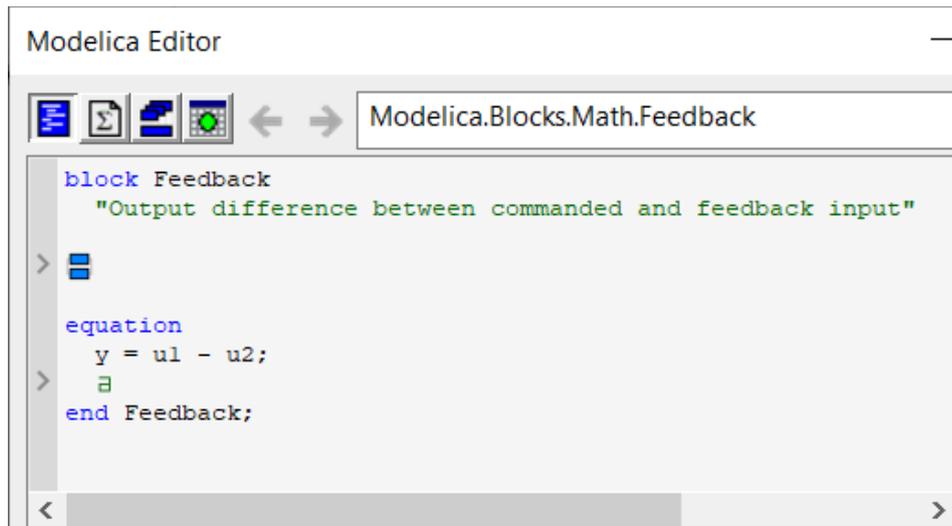
results in the following equations:

$$y = u1 - u2$$

Connectors

Type	Name	Description
input RealInput	u1	
input RealInput	u2	
output RealOutput	y	

Figura A.6 Descripción del bloque Feedback



The image shows a screenshot of the Modelica Editor interface. The title bar reads "Modelica Editor". Below the title bar is a toolbar with several icons, including a blue square, a sigma symbol, a blue square with a white circle, and a green circle with a white square. To the right of the toolbar is a text field containing the path "Modelica.Blocks.Math.Feedback". The main area of the editor displays the following code:

```
block Feedback
  "Output difference between commanded and feedback input"
>
equation
  y = u1 - u2;
end Feedback;
```

Figura A.7 Programación en Modelica del bloque Feedback

A.4 Bloque Add

Este es un bloque de suma utilizado que recibe parámetros mediante los conectores y el bloque de configuración. Recibe dos señales de entrada y devuelve una tercera con la suma de estos valores. En la figura A.8 se muestra la información completa, los parámetros y conectores utilizados. La Figura A.9 muestra la sintaxis, en la que además que se muestra la posibilidad de agregar una constante como parámetro, a cada señal, a manera de ganancia.

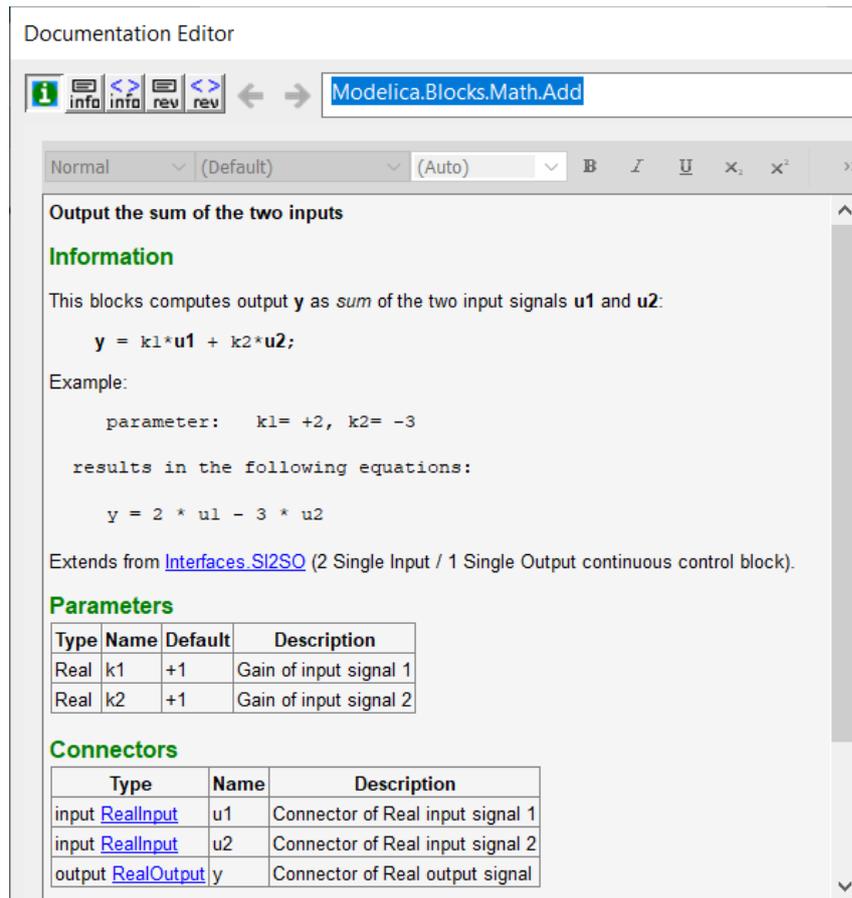


Figura A.8 Descripción del bloque Feedback

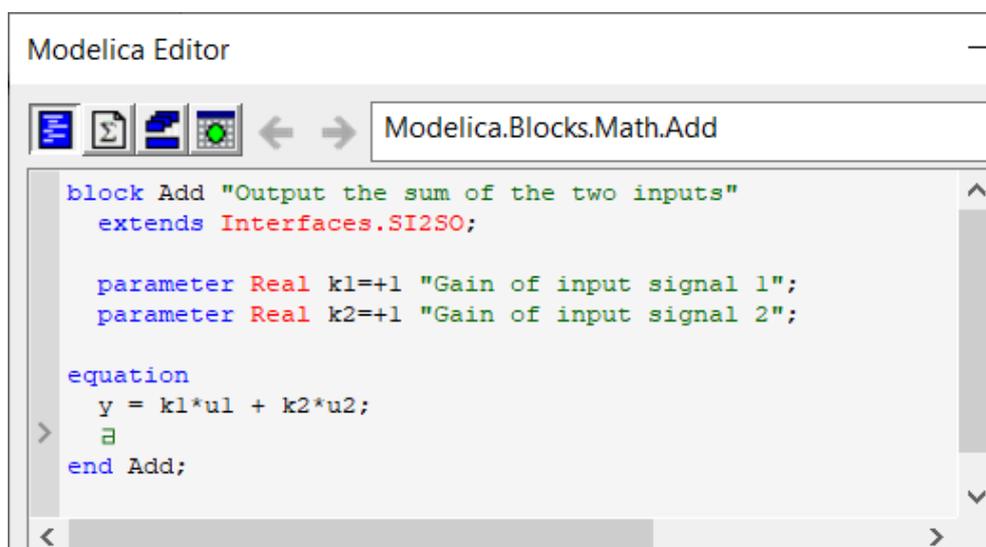


Figura A.9 Programación en Modelica del bloque Add

A.5 Bloque PI

Este es un bloque de control proporcional integral, que recibe como parámetros una constante real como ganancia proporcional y un tiempo de integración. La figura A.10 incluye la información de la forma en la que se define la función del controlador y la Figura A.11 muestra los parámetros y conectores necesarios. La figura A.12 muestra la sintaxis de inicialización y operación necesarias para que se cumpla el comportamiento.

Documentation Editor

Modelica.Blocks.Continuous.PI

Normal (Default) (Auto) B I U x₁ x² A ↻ ⋮ >>

Proportional-Integral controller

Information

This blocks defines the transfer function between the input u and the output y as PI system:

$$y = k * \left(1 + \frac{1}{T*s} \right) * u$$
$$= k * \frac{T*s + 1}{T*s} * u$$

If you would like to be able to change easily between different transfer functions (FirstOrder, SecondOrder, ...) by changing parameters, use the general model class **TransferFunction** instead and model a PI SISO system with parameters $b = \{k*T, k\}$, $a = \{T, 0\}$.

Example:

```
parameter: k = 0.3, T = 0.4
```

results in:

$$y = 0.3 \frac{0.4 s + 1}{0.4 s} * u$$

It might be difficult to initialize the PI component in steady state due to the integrator part. This is discussed in the description of package [Continuous](#).

Extends from [Interfaces.SISO](#) (Single Input Single Output continuous control block).

Figura A.10 Descripción 1 del bloque PI

Documentation Editor

Modelica.Blocks.Continuous.PI

Parameters

Type	Name	Default	Description
Real	k	1	Gain [1]
Time	T	T(start=1, min=Modelica.Cons...	Time Constant (T>0 required) [s]
Initialization			
Init	initType	Modelica.Blocks.Types.Init.N...	Type of initialization (1: no init, 2: steady state, 3: initial state, 4: initial output)
Real	x_start	0	Initial or guess value of state
Real	y_start	0	Initial value of output

Connectors

Type	Name	Description
input	RealInput u	Connector of Real input signal
output	RealOutput y	Connector of Real output signal

Figura A.11 Descripción 2 del bloque PI

Modelica Editor

Modelica.Blocks.Continuous.PI

```

block PI "Proportional-Integral controller"
  import Modelica.Blocks.Types.Init;
  parameter Real k(unit="1") = 1 "Gain";
  parameter SIunits.Time T(start=1,min=Modelica.Constants.small)
    "Time Constant (T>0 required)";
  parameter Modelica.Blocks.Types.Init initType=Modelica.Blocks.Types.Init.NoInit
    "Type of initialization (1: no init, 2: steady state, 3: initial state, 4: init
  & ;
  parameter Real x_start=0 "Initial or guess value of state"
  & ;
  parameter Real y_start=0 "Initial value of output" & ;

  extends Interfaces.SISO;
  output Real x(start=x_start) "State of block";

  initial equation
    if initType == Init.SteadyState then
      der(x) = 0;
    elseif initType == Init.InitialState then
      x = x_start;
    elseif initType == Init.InitialOutput then
      y = y_start;
    end if;
  equation
    der(x) = u/T;
    y = k*(x + u);
  & ;
end PI;

```

Figura A.12 Programación en Modelica del bloque PI

A.6 Bloque World

Este bloque define nuestro sistema de referencia global tridimensional y la fuerza de gravedad sobre la que se encuentra nuestro modelo. Es indispensable en un sistema de múltiples elementos para una simulación en conjunto. La Figura A.13 muestra la principal información acerca del bloque.

Documentation Editor

Modelica.Mechanics.MultiBody.World

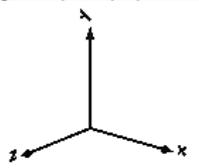
Normal (Default) (Auto) B I U x₁ x₂ A ↻

World coordinate system + gravity field + default animation definition

Information

Model **World** represents a global coordinate system fixed in ground. This model serves several purposes:

- It is used as **inertial system** in which the equations of all elements of the MultiBody library are defined.
- It is the world frame of an **animation window** in which all elements of the MultiBody library are visualized.
- It is used to define the **gravity field** in which a multi-body model is present. Default is a uniform gravity field where the gravity acceleration vector g is the same at every position. Additionally, a point gravity field or no gravity can be selected. Also, function `gravityAcceleration` can be redeclared to a user-defined function that computes the gravity acceleration, see example [Examples.Elementary.UserDefinedGravityField](#).
- It is used to define **default settings** of animation properties (e.g., the diameter of a sphere representing by default the center of mass of a body, or the diameters of the cylinders representing a revolute joint).
- It is used to define a **visual representation** of the world model (= 3 coordinate axes with labels), of the defined gravity field and of a ground plane perpendicular to the gravity direction.



Since the gravity field function is required from all bodies with mass and the default settings of animation properties are required from nearly every component, exactly one instance of model **World** needs to be present in every model on the top level. The basic declaration needs to be:

```
inner Modelica.Mechanics.MultiBody.World world
```

Note, it must be an **inner** declaration with instance name **world** in order that this world object can be accessed from all objects in the model. When dragging the "World" object from the package browser into the diagram layer, this declaration is automatically generated (this is defined via annotations in model **World**).

All vectors and tensors of a mechanical system are resolved in a frame that is local to the corresponding component. Usually, if all relative joint coordinates vanish, the local frames of all components are parallel to each other, as well as to the world frame (this holds as long as a `Parts.FixedRotation` component is **not** used). In this "reference configuration" it is therefore alternatively possible to resolve all vectors in the world frame, since all frames are parallel to each other. This is often very convenient. In order to give some visual support in such a situation, in the icon of a **World** instance two axes of the world frame are shown and the labels of these axes can be set via parameters.

Figura A.13 Descripción del bloque World

A.7 Bloque BodyShapeWithFrames

Este bloque permite la instanciación de cada pieza creada en el diseñador de partes 3D. Posee información de los puntos de conexión entre piezas y de las propiedades que estas poseen. La figura A.14 posee la descripción del bloque y los parámetros configurables del mismo. La figura A.15 muestra la inicialización de las variables que contienen la información de las propiedades de la pieza.

Documentation Editor

CATIAMultiBody.Parts.BodyShapeWithFrames

Rigid body with mass, inertia tensor, different shapes for animation, and two frame connectors (12 potential states)

Information

Rigid body with mass and inertia tensor and **two frame connectors**. All parameter vectors have to be resolved in frame_a. The **inertia tensor** has to be defined with respect to a coordinate system that is parallel to frame_a with the origin at the center of mass of the body. The coordinate system **frame_b** is always parallel to **frame_a**.

By default, this component is visualized by any **shape** that can be defined with `Modelica.Mechanics.MultiBody.Visualizers.FixedShape`. This shape is placed between frame_a and frame_b (default: `length(shape) = Frames.length(r)`). Additionally a **sphere** may be visualized that has its center at the center of mass. Note, that the animation may be switched off via parameter `animation = false`.

The following shapes can be defined via parameter **shapeType**, e.g., `shapeType="cone"`:

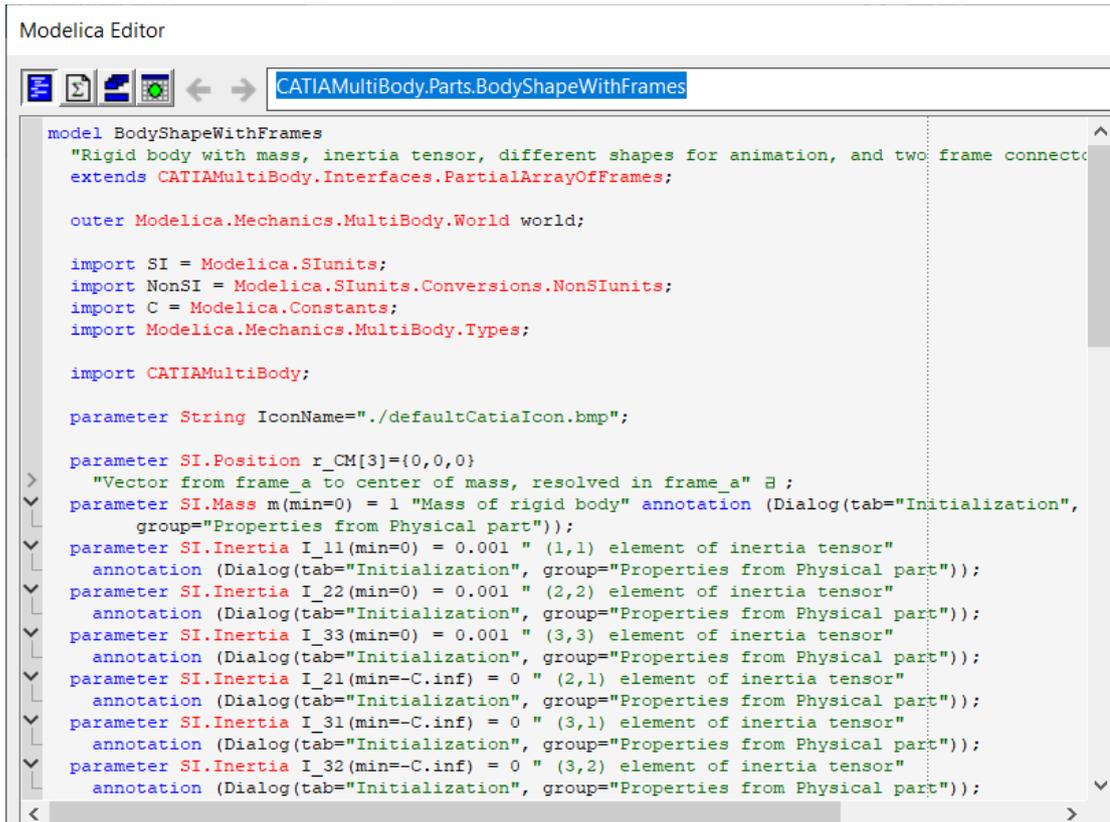
A `BodyShape` component has potential states. For details of these states and of the "Advanced" menu parameters, see model [MultiBody.Parts.Body](#).

Extends from [CATIAMultiBody.Interfaces.PartialArrayOfFrames](#).

Parameters

Type	Name	Default	Description
String	IconName	"/defaultCatalcon.bmp"	
Boolean	useColorConnectors	false	=true, if color should be enabled during animation
Boolean	animateBodyFrames	false	=true, if shapes for debugging body frames should be visible
Boolean	animateBodies	false	=true, if shapes for debugging bodies should be visible
Boolean	animateVRML	if PartName == "" then true ...	
Boolean	useCOG	false	=true, if connector for center of gravity should be visible
Boolean	isVolumeWithoutMass	if m <= 0 and isVolume == tr...	
Physical animation			
String	PartName	""	
Boolean	hidePhysicalPart	false	Hide associated Physical Part
VRML animation			
ShapeType	shapeType	"cylinder"	Type of shape
Color	colorVRML	Modelica.Mechanics.MultiBody...	Color of VRML shape

Figura A.14 Descripción del bloque World



```
Modelica Editor
CATIAMultiBody.Parts.BodyShapeWithFrames

model BodyShapeWithFrames
  "Rigid body with mass, inertia tensor, different shapes for animation, and two frame connecto
  extends CATIAMultiBody.Interfaces.PartialArrayOfFrames;

  outer Modelica.Mechanics.MultiBody.World world;

  import SI = Modelica.SIunits;
  import NonSI = Modelica.SIunits.Conversions.NonSIunits;
  import C = Modelica.Constants;
  import Modelica.Mechanics.MultiBody.Types;

  import CATIAMultiBody;

  parameter String IconName="./defaultCatiaIcon.bmp";

  parameter SI.Position r_CM[3]={0,0,0}
    "Vector from frame_a to center of mass, resolved in frame_a" # ;
  parameter SI.Mass m(min=0) = 1 "Mass of rigid body" annotation (Dialog(tab="Initialization",
    group="Properties from Physical part"));
  parameter SI.Inertia I_11(min=0) = 0.001 " (1,1) element of inertia tensor"
    annotation (Dialog(tab="Initialization", group="Properties from Physical part"));
  parameter SI.Inertia I_22(min=0) = 0.001 " (2,2) element of inertia tensor"
    annotation (Dialog(tab="Initialization", group="Properties from Physical part"));
  parameter SI.Inertia I_33(min=0) = 0.001 " (3,3) element of inertia tensor"
    annotation (Dialog(tab="Initialization", group="Properties from Physical part"));
  parameter SI.Inertia I_21(min=-C.inf) = 0 " (2,1) element of inertia tensor"
    annotation (Dialog(tab="Initialization", group="Properties from Physical part"));
  parameter SI.Inertia I_31(min=-C.inf) = 0 " (3,1) element of inertia tensor"
    annotation (Dialog(tab="Initialization", group="Properties from Physical part"));
  parameter SI.Inertia I_32(min=-C.inf) = 0 " (3,2) element of inertia tensor"
    annotation (Dialog(tab="Initialization", group="Properties from Physical part"));
```

Figura A.15 Programación en Modelica del bloque BodyShapeWithFrames

A.8 Bloque Rigid

Este bloque define el comportamiento de las juntas rígidas, las cuales requieren de dos conectores que aceptaran las coordenadas de asociación para cada elemento. La figura A.16 muestra la información de parámetros y conectores que permite la creación de una junta rígida. La figura A.17 muestra la programación Modelica de este bloque.

Documentation Editor

CATIAMultiBodyJoints.Rigid

Normal (Default) (Auto) B I U x₂ x² A >>

Information

Extends from [CATIAMultiBody.Interfaces.PartialJoint](#).

Parameters

Type	Name	Default	Description
Animation			
Boolean	animation	false	= true, if animation shall be enabled (show axis as cylinder)

Connectors

Type	Name	Description
Frame_a	frame_a	Coordinate system a fixed to the component with one cut-force and cut-torque
Frame_b	frame_b	Coordinate system b fixed to the component with one cut-force and cut-torque

Figura A.16 Descripción del bloque Rigid

Modelica Editor

CATIAMultiBodyJoints.Rigid

```

model Rigid
  extends CATIAMultiBody.Interfaces.PartialJoint;
equation
  connect(frame_a, frame_b) a ;
  a
end Rigid;

```

Figura A.17 Programación en Modelica del bloque Rigid

A.9 Bloque Revolute

El bloque de junta revoluta posee una mayor cantidad de parámetros, pues incluye límites para definir los rangos de movimiento. La figura A.18 tiene la información de Parámetros y conectores necesarios para la creación de una junta revoluta. La figura A.19 muestra la inicialización de los parámetros para cumplir el comportamiento de la junta.

Documentation Editor

info info rev rev CATIAMultiBody.Joints.Revolute

Normal (Default) (Auto) B I U x, x² A ↻

Parameters

Type	Name	Default	Description
RotationalLimiter	rotationalLimiter	rotationalLimiter(phi_min=ph...	
Animation			
Boolean	animation	false	= true, if animation shall be enabled (show axis as cylinder)
Control			
Boolean	useAxisFlange	rotationControl <> CATIAMult...	= true, if axis flange is enabled
ControlType	rotationControl	CATIAMultiBody.Interfaces.Co...	type of control for the rotational flanges
Initialization			
Real	n[3]	{0,0,1}	
Angle	phi_start	0	[rad]
AngularVelocity	w_start	0	[rad/s]
AngularAcceleration	a_start	0	[rad/s²]
Advanced			
Boolean	fixedStartValues_phi	if rotationControl == Contro...	
StateSelect	stateSelect	if rotationControl == Contro...	Priority to use joint angle phi and w=der(phi) as states
Limiters			
LimiterType	limiterType	LimiterType.OutOfBoundWarning	Modelization of joints limits
Angle	phi_min	0	Joint min limit [rad]
Angle	phi_max	0	Joint max limit [rad]
Boolean	hasMinLimit_phi	false	activate min limit
Boolean	hasMaxLimit_phi	false	activate max limit
Connectors			
Type	Name	Description	
Frame_a	frame_a	Coordinate system a fixed to the component with one cut-force and cut-torque	
Frame_b	frame_b	Coordinate system b fixed to the component with one cut-force and cut-torque	
Flange_a	axis	1-dim. rotational flange that drives the joint	
Flange_b	bearing	1-dim. rotational flange of the drive bearing	

Figura A.18 Descripción del bloque Revolute

```
Modelica Editor
CATIAMultiBody.Joints.Revolute

> parameter Real n[3]={0,0,1} @ ;
>
> constant SI.Angle phi_offset=0 @ ;
> parameter SI.Angle phi_start=0 @ ;
> parameter SI.AngularVelocity w_start=0 @ ;
> parameter SI.AngularAcceleration a_start=0 @ ;
>
> parameter Boolean useAxisFlange=rotationControl <> CATIAMultiBody.Interfaces.ControlType.None
>   "= true, if axis flange is enabled" @ ;
>
> parameter Boolean fixedStartValues_phi=if rotationControl == ControlType.ByForce
>   then true else false @ ;
>
> parameter StateSelect stateSelect=if rotationControl == ControlType.ByForce
>   then StateSelect.always else StateSelect.prefer
>   "Priority to use joint angle phi and w=der(phi) as states" @ ;
>
> parameter ControlType rotationControl=CATIAMultiBody.Interfaces.ControlType.None
>   "type of control for the rotational flanges" @ ;
>
>
> parameter LimiterType limiterType=LimiterType.OutOfBoundWarning
>   "Modelization of joints limits" @ ;
> parameter SI.Angle phi_min=0 "Joint min limit" @ ;
> parameter SI.Angle phi_max=0 "Joint max limit"
>   @ ;
> parameter Boolean hasMinLimit_phi=false "activate min limit" @ ;
> parameter Boolean hasMaxLimit_phi=false "activate max limit" @ ;
>
>
protected
> parameter Boolean annotRuleVisible_rotation=if rotationControl ==
>   CATIAMultiBody.Interfaces.ControlType.ByPosition then true else false;
equation
> assert(
>   (limiterType <> LimiterType.OutOfBoundWarning and limiterType <>
>     LimiterType.OutOfBoundError) or ((hasMinLimit_phi == false or phi_min <=
>       jointNonPlanar.phi) and (hasMaxLimit_phi == false or phi_max >=
>         jointNonPlanar.phi)),
>   "Out of bounds for CATIAMultiBody.Joints.Revolute: phi is outside defined boundaries.",
>   if limiterType == LimiterType.OutOfBoundWarning then AssertionLevel.warning
>     else AssertionLevel.error);
```

Figura A.19 Programación en Modelica del bloque Revolute

A.10 Bloque Screw

Este bloque permite la creación de una junta tipo tornillo, la cual posee una gran cantidad de parámetros que sirven para limitar el comportamiento. La figura A.20 muestra los parámetros configurables y las señales que maneja mediante conectores. La figura A.21 muestra la variedad de configuraciones permitidas en el código Modelica.

Documentation Editor

CATIAMultiBody.Joints.Screw

Normal (Default) (Auto)

Parameters

Type	Name	Default	Description
Real	ratio	1	Screw speed ratio
Animation			
Boolean	animation	false	= true, if animation shall be enabled (show axis as cylinder)
Control			
Boolean	useAxisFlange	(rotationControl <> CATIAMul...	= true, if axis flange is enabled
ControlType	rotationControl	CATIAMultiBody.Interfaces.Co...	type of control for the rotational flanges
ControlType	translationControl	CATIAMultiBody.Interfaces.Co...	type of control for the translational flanges
Initialization			
Real	n[3]	{0,0,1}	
Real	phi_start	0	
Real	s_start	0	
Real	w_start	0	
Real	v_start	0	
Real	a_start	0	
Real	wd_start	0	
Advanced			
Boolean	fixedStartValues_phi	(rotationControl == ControIT...	
StateSelect	stateSelect_phi	if rotationControl == Contro...	Priority to use joint angle phi and w=der(phi) as states
Boolean	fixedStartValues_s	(translationControl == Contr...	
StateSelect	stateSelect_s	if translationControl == Con...	Priority to use joint coordinates (s, v) as states

Connectors

Type	Name	Description
Frame_a	frame_a	Coordinate system a fixed to the component with one cut-force and cut-torque
Frame_b	frame_b	Coordinate system b fixed to the component with one cut-force and cut-torque
Flange_a	axis_revolute	1-dim. rotational flange that drives the joint
Flange_b	bearing_revolute	1-dim. rotational flange of the drive bearing
Flange_a	axis_prismatic	1-dim. translational flange that drives the joint
Flange_b	bearing_prismatic	1-dim. translational flange of the drive bearing

Figura A.20 Descripción del bloque Screw

```
Modelica Editor
CATIAMultiBody.Joints.Screw

extends CATIAMultiBody.Interfaces.PartialJoint;

import CATIAMultiBody.Interfaces.ControlType;

parameter Boolean useAxisFlange=(rotationControl <> CATIAMultiBody.Interfaces.ControlType.None
or translationControl <> CATIAMultiBody.Interfaces.ControlType.None)
"= true, if axis flange is enabled" ;

parameter Real n[3]={0,0,1} ;

parameter Real phi_start=0 ;
parameter Real s_start=0 ;
parameter Real w_start=0 ;
parameter Real v_start=0 ;
parameter Real a_start=0 ;
parameter Real wd_start=0 ;

parameter Boolean fixedStartValues_phi=(rotationControl == ControlType.ByForce)
;
parameter StateSelect stateSelect_phi=if rotationControl == ControlType.ByForce
then StateSelect.always else StateSelect.prefer
"Priority to use joint angle phi and w=der(phi) as states" ;
parameter Boolean fixedStartValues_s=(translationControl == ControlType.ByForce)
;
parameter StateSelect stateSelect_s=if translationControl == ControlType.ByForce
then StateSelect.always else StateSelect.prefer
"Priority to use joint coordinates (s, v) as states" ;

parameter ControlType rotationControl=CATIAMultiBody.Interfaces.ControlType.None
"type of control for the rotational flanges" ;
parameter ControlType translationControl=CATIAMultiBody.Interfaces.ControlType.None
"type of control for the translational flanges" ;

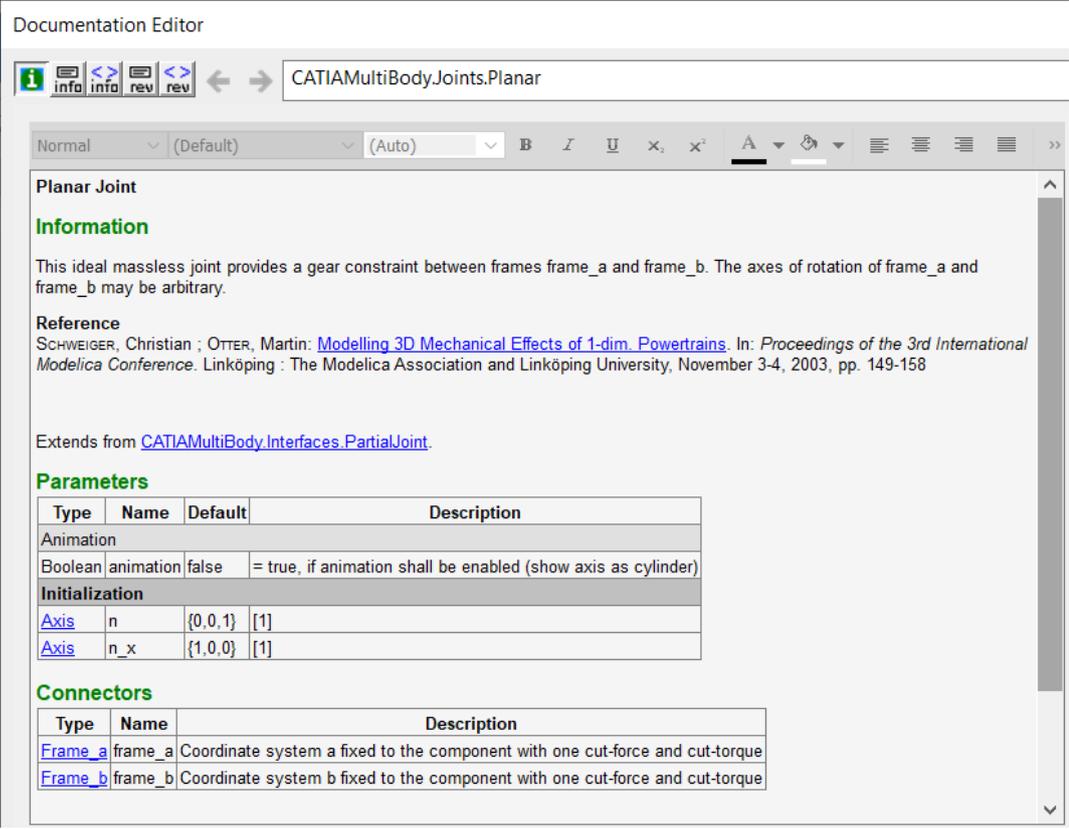
parameter Real ratio=1 "Screw speed ratio";

protected
parameter Boolean annotRuleVisible_rotation=(rotationControl ==
CATIAMultiBody.Interfaces.ControlType.ByPosition);
parameter Boolean annotRuleVisible_translation=(translationControl ==
CATIAMultiBody.Interfaces.ControlType.ByPosition);
equation
end Screw;
```

Figura A.21 Programación en Modelica del bloque Screw

A.11 Bloque Planar

Este bloque sirve para la creación de un modelo de junta de plano o coincidencia de superficie plana. La figura A.22 muestra los parámetros que recibe el bloque y los conectores que sirven de interfaz para las señales. La figura A.23 muestra las funciones necesarias para la creación de un elemento de este tipo.



Documentation Editor

CATIAMultiBodyJoints.Planar

Normal (Default) (Auto) B I U x x' A

Planar Joint

Information

This ideal massless joint provides a gear constraint between frames `frame_a` and `frame_b`. The axes of rotation of `frame_a` and `frame_b` may be arbitrary.

Reference

SCHWEIGER, Christian ; OTTER, Martin: [Modelling 3D Mechanical Effects of 1-dim. Powertrains](#). In: *Proceedings of the 3rd International Modelica Conference*. Linköping : The Modelica Association and Linköping University, November 3-4, 2003, pp. 149-158

Extends from [CATIAMultiBody.Interfaces.PartialJoint](#).

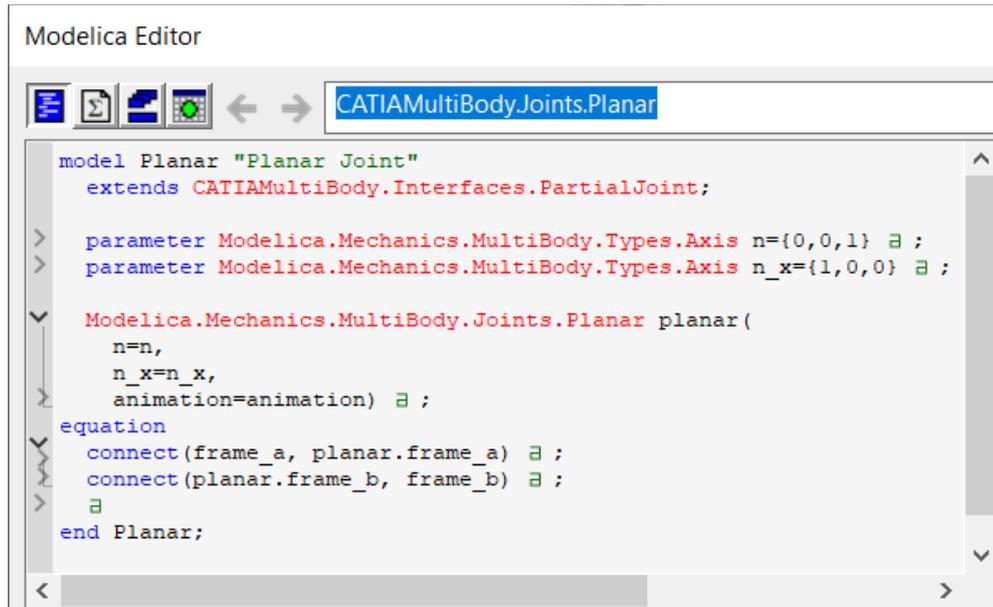
Parameters

Type	Name	Default	Description
Animation			
Boolean	animation	false	= true, if animation shall be enabled (show axis as cylinder)
Initialization			
Axis	n	{0,0,1}	[1]
Axis	n_x	{1,0,0}	[1]

Connectors

Type	Name	Description
Frame_a	frame_a	Coordinate system a fixed to the component with one cut-force and cut-torque
Frame_b	frame_b	Coordinate system b fixed to the component with one cut-force and cut-torque

Figura A.22 Descripción del bloque World



The image shows a screenshot of the Modelica Editor interface. The title bar reads "Modelica Editor". Below the title bar is a toolbar with several icons, including a search icon, a save icon, a refresh icon, and a run icon. The main window displays the code for the "Planar Joint" block. The code is as follows:

```
model Planar "Planar Joint"
  extends CATIAMultiBody.Interfaces.PartialJoint;

  > parameter Modelica.Mechanics.MultiBody.Types.Axis n={0,0,1} @ ;
  > parameter Modelica.Mechanics.MultiBody.Types.Axis n_x={1,0,0} @ ;

  Modelica.Mechanics.MultiBody.Joints.Planar planar (
    n=n,
    n_x=n_x,
    animation=animation) @ ;
  equation
  connect(frame_a, planar.frame_a) @ ;
  connect(planar.frame_b, frame_b) @ ;
  @
end Planar;
```

Figura A.23 Programación en Modelica del bloque Planar

A.12 Bloque Spherical

Este bloque permite la creación del modelo para una junta esférica, comparte características con la junta planar, pero esta sirve para coincidencia de planos curvos. La figura A.24 muestra los parámetros y conectores que se necesitan para este tipo de elementos. La figura A.25 muestra el código Modelica para la creación de este modelo.

Documentation Editor

CATIAMultiBody.Joints.Spherical

Normal (Default) (Auto) B I U x₂ x² A >>

Information

Extends from [CATIAMultiBody.Interfaces.PartialJoint](#).

Parameters

Type	Name	Default	Description
Animation			
Boolean	animation	false	= true, if animation shall be enabled (show axis as cylinder)
Initialization			
Boolean	fixed_angles_start	false	
Boolean	fixed_w_rel_a_start	false	

Connectors

Type	Name	Description
Frame_a	frame_a	Coordinate system a fixed to the component with one cut-force and cut-torque
Frame_b	frame_b	Coordinate system b fixed to the component with one cut-force and cut-torque

Figura A.24 Descripción del bloque Spherical

Modelica Editor

CATIAMultiBody.Joints.Spherical

```

model Spherical
  extends CATIAMultiBody.Interfaces.PartialJoint;

  > parameter Boolean fixed_angles_start=false @ ;
  > parameter Boolean fixed_w_rel_a_start=false @ ;

  > // MultiBodyCutJoints.SphericalCutJoint joint(automatic={
equation
  connect(frame_a, joint.frame_a) @ ;
  connect(joint.frame_b, frame_b) @ ;
  @
end Spherical;

```

Figura A.25 Programación en Modelica del bloque Spherical

APÉNDICE B

Descripción del ensamble 3D de la máquina 3PRS+XY+C

Este apéndice muestra como está conformado el ensamble que integra todo el mecanismo y las características adicionales que posee. Se describen los elementos con el nombre que han sido creados en la aplicación y se detalla una descripción de estos.

El ensamble toma el nombre de 3PRS-XYDigitalTwin, este posee una representación del mecanismo, una definición restrictiva de comportamientos, 20 elementos hijos que se muestran en la figura B.1 y 36 conexiones ingenieriles para estos las que se muestran en la figura B.2.



Figura B.1 Elementos del ensamble digital



Figura B.2 Juntas del ensamble digital

A continuación, se describe y se muestra cada elemento del ensamble. Cabe recalcar que algunos de estos elementos también poseen subensambles, cuyas partes serán mostradas en imágenes. Todos los subensambles poseen únicamente juntas fijas para ser unidos.

B.1 Herramienta

Esta pieza es la punta de la herramienta de corte de metal, la cual posee un solo filo. La figura B.3 muestra su forma. La herramienta se encuentra unida al Conjunto tripodo y su movimiento relativo es el movimiento del eje C del mecanismo.



Figura B.3 Herramienta

B.2 PSS1205N1D1D0421

Esta pieza es el eje vertical sobre el que se deslizan las correderas de las tres juntas prismáticas correspondientes a los tres ejes z paralelos. La figura B.4 muestra su forma



Figura B.4 PSS1205N1D1D0421

B.3 Mesa inferior

Esta pieza es el elemento de conexión entre el bastidor y la mesa de trabajo. Posee una unión de tipo fija con el bastidor y sobre esta se ubica la mesa superior. La figura B.5 muestra su forma y la figura B.6 muestra los elementos que la conforman.

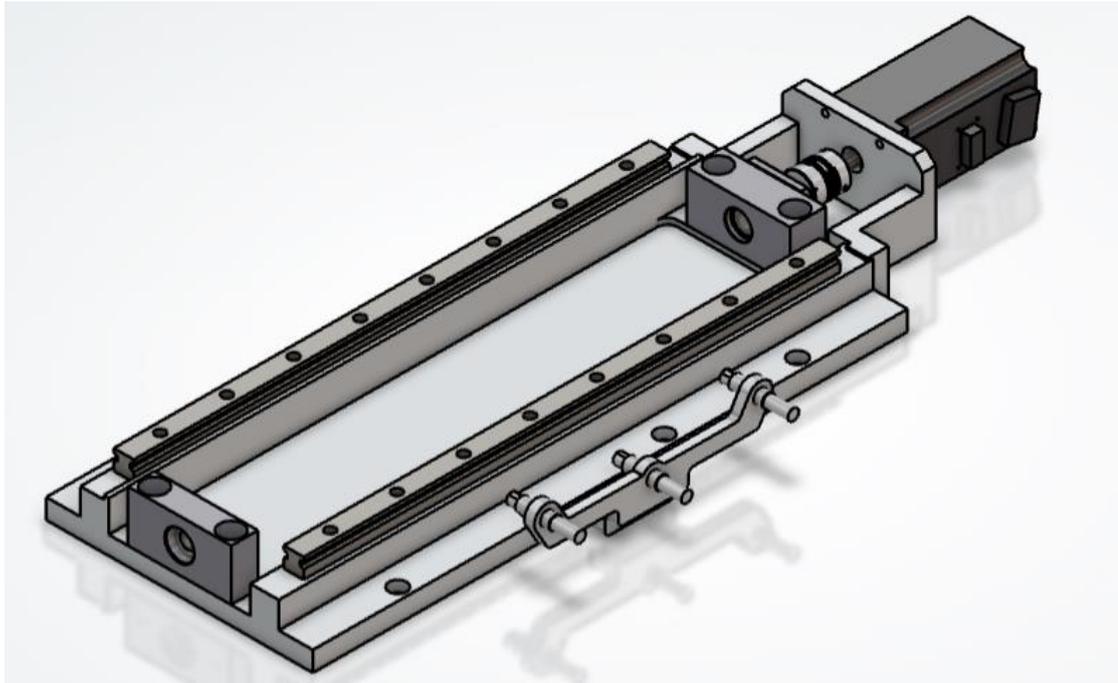


Figura B.5 Mesa inferior

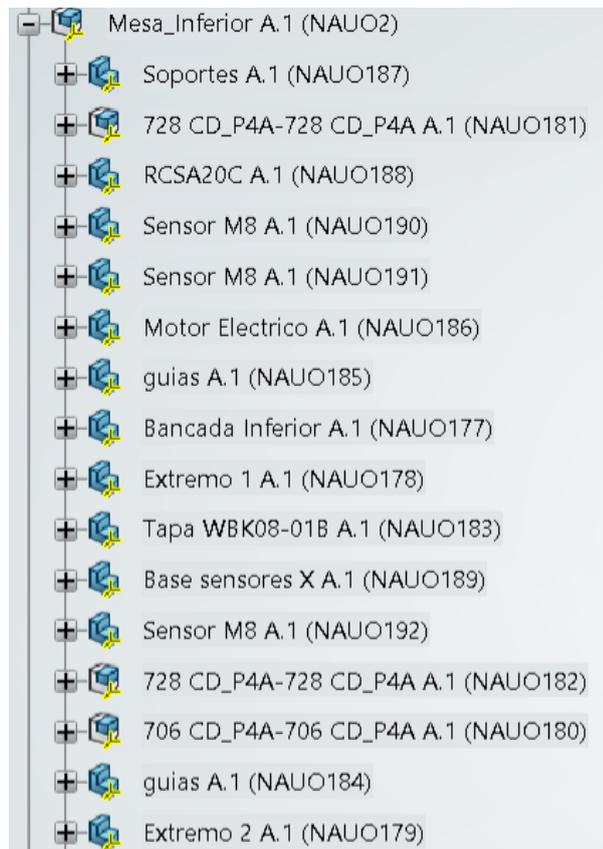


Figura B.6 elementos de la Mesa inferior

B.4 Mesa superior

Esta pieza se conecta con la mesa inferior. Su movimiento deslizante respecto a la mesa inferior produce el movimiento en X de la mesa de trabajo. La figura B.7 muestra la forma de esta mesa y la B.8 los elementos que la conforman.

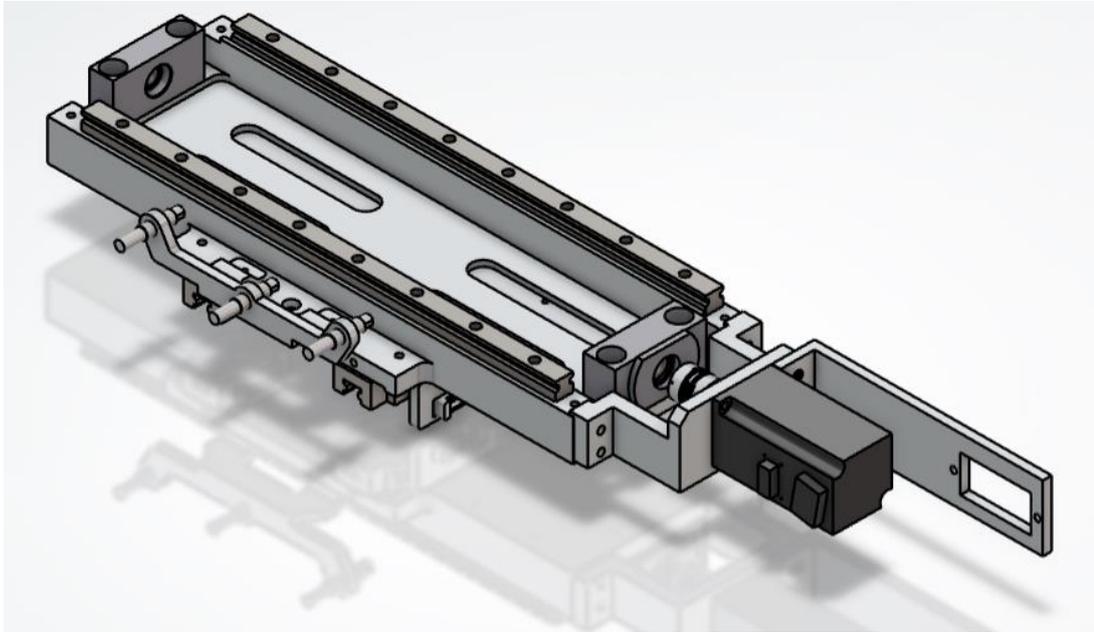


Figura B.7 Mesa superior



Figura B.8 elementos de la Mesa superior

B.5 Conjunto tripodo

Esta pieza se conecta con los tres brazos de los ejes z paralelos. Cada brazo posee una junta esférica que permite el posicionamiento del conto tripodo, mediante las distintas configuraciones posibles de altura para los tres brazos. La figura B.9 muestra su forma y la figura B.10 sus elementos.

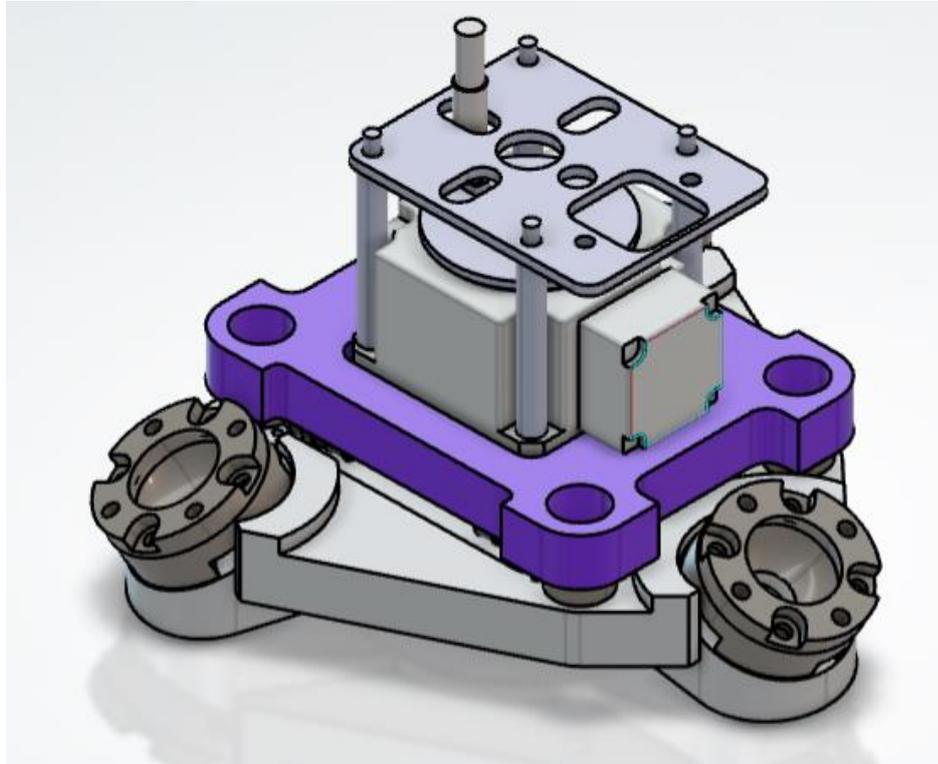


Figura B.9 Conjunto tripodo



Figura B.10 elementos del Conjunto tripodo

B.6 Conjunto Brazo

Esta pieza es el elemento de unión entre las juntas prismáticas de los ejes z paralelos y el Conjunto tripodo mediante la junta esférica. La figura B.11 muestra la forma del brazo y la figura B.12 muestran los elementos que lo conforman.

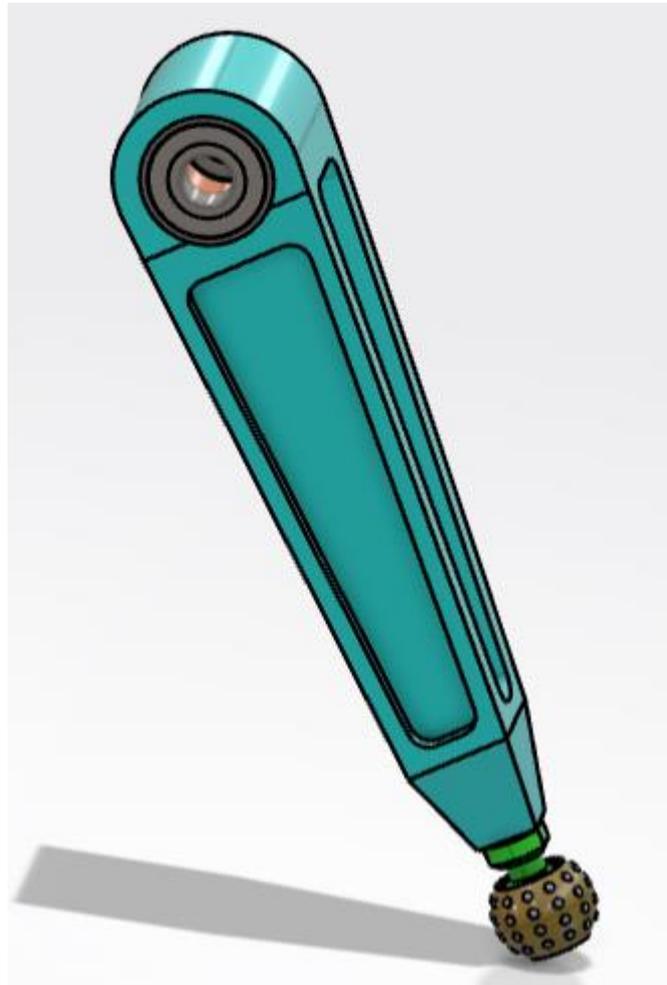


Figura B.11 Conjunto Brazo

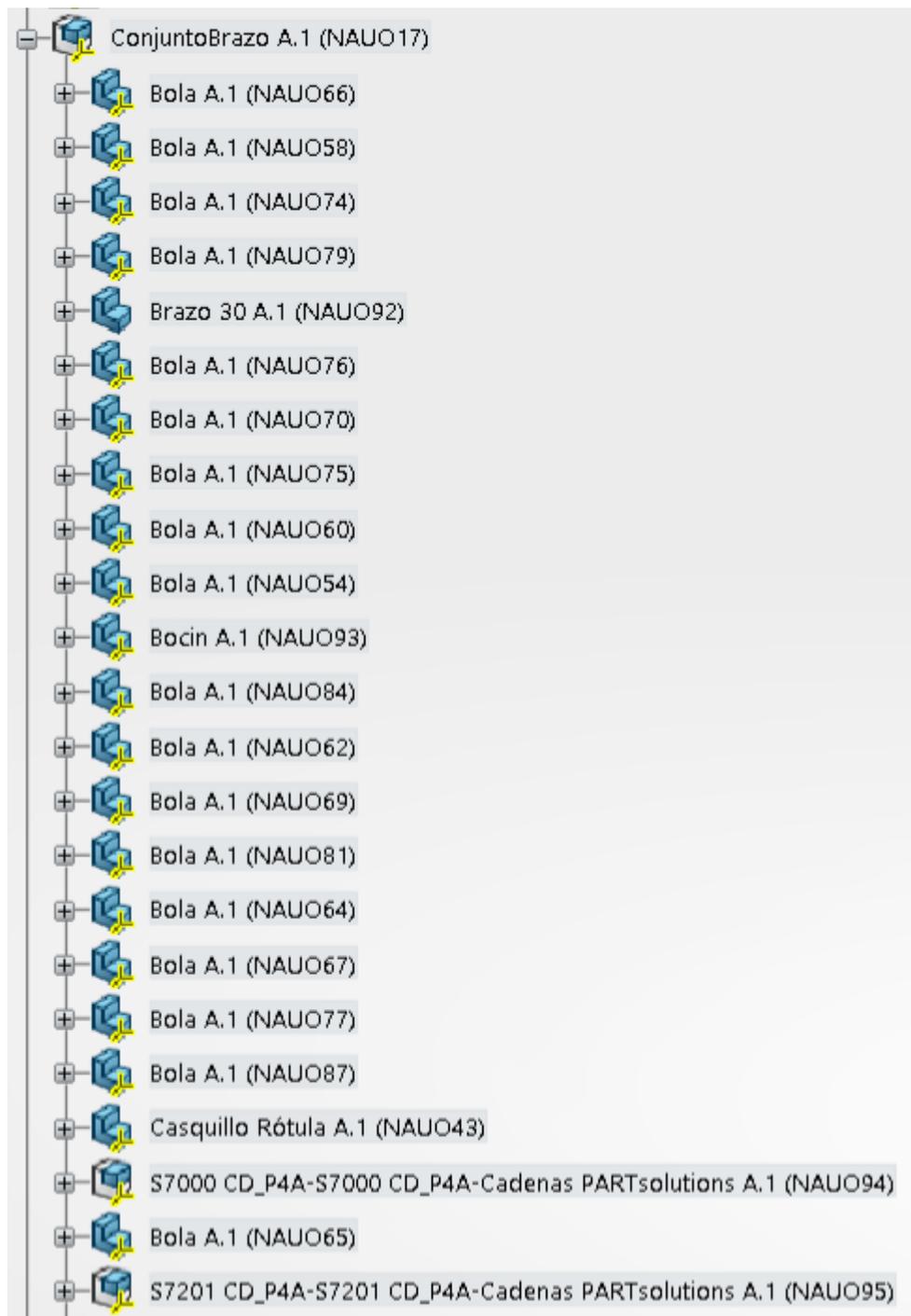


Figura B.12 elementos del Conjunto Brazo

B.7 Eje 2 O SW 328mm

Esta pieza es el eje de deslizamiento para las juntas X Y de la base. Posee únicamente una diferencia longitudinal con respecto a los ejes Z. La figura B.12 muestra la forma del elemento.



Figura B.13 Eje 2 O SW 328mm

B.8 Mesa vertical

Este elemento es la conexión entre el bastidor y los elementos del eje z. posee las guías y límites de recorrido para el desplazamiento de las correderas que se unen a estos. La figura B.14 muestra su forma y la B.15 los elementos que lo conforman.

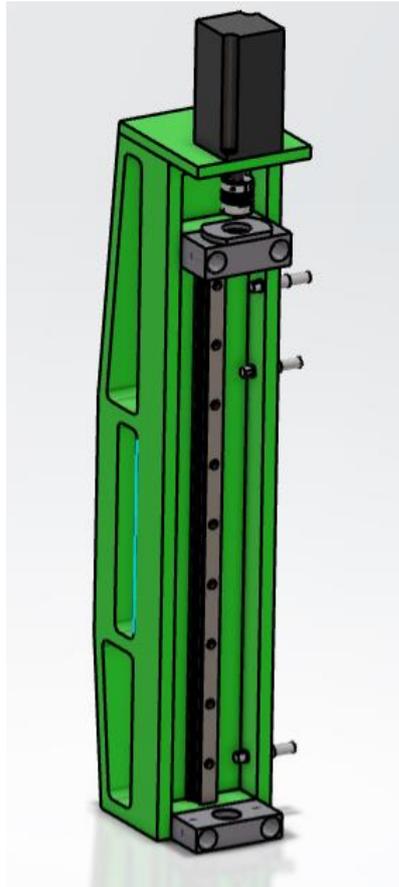


Figura B.14 Mesa vertical



Figura B.15 elementos de la Mesa vertical

B.9 Bastidor

Esta pieza es la principal estructura del mecanismo, sobre esta se sostienen todos los demás elementos. La figura B.16 muestra su forma y la B.17 sus elementos.

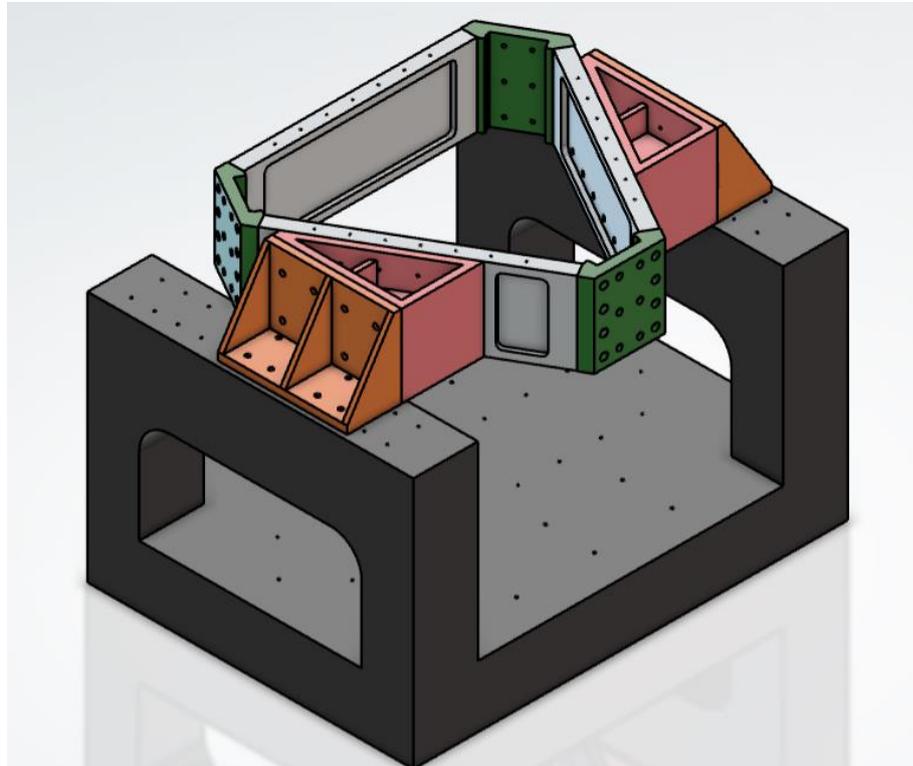


Figura B.16 Bastidor

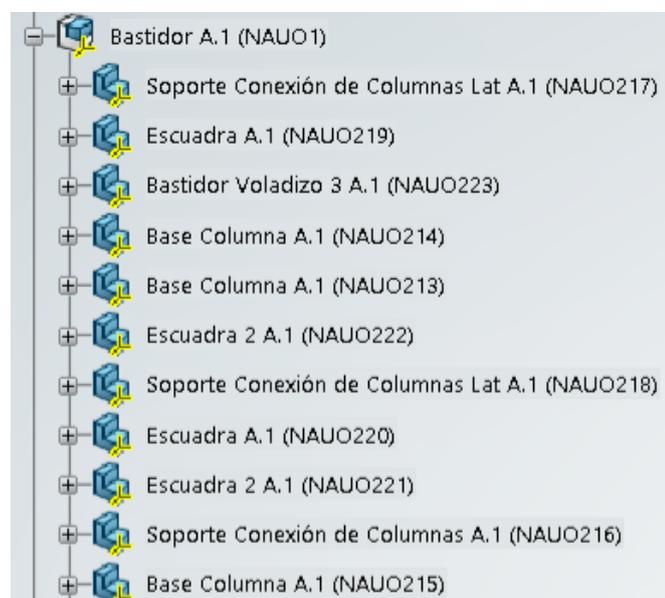


Figura B.17 elementos del Bastidor

B.10 Conjunto corredera

Esta pieza es la corredera que se encuentra en cada eje z. esta pieza permite el cambio de posición para cada brazo. El movimiento sobre esta es controlado mediante un tornillo y un rodamiento de bolas. La forma del elemento se muestra en la figura B.18 y la figura B.19 muestra los elementos que lo conforman.

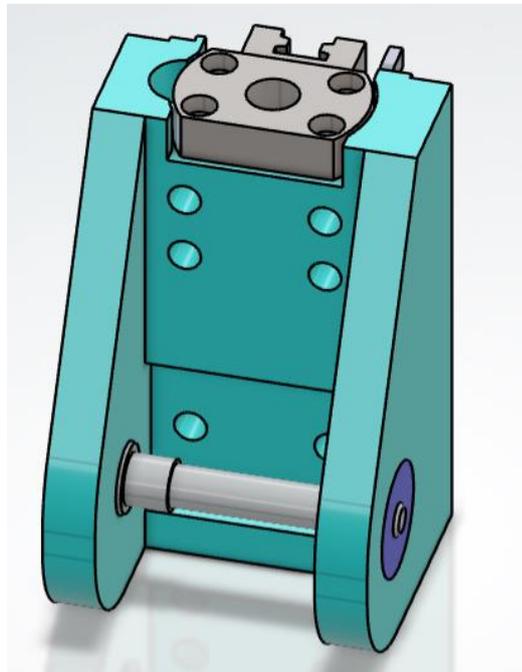


Figura B.18 Conjunto corredera



Figura B.19 elementos del Conjunto corredera

B.11 Mesa trabajo

Esta pieza es la base sobre la que se colocan los elementos que se desean mecanizar. Está conectada a la mesa superior mediante una junta prismática y su movimiento relativo produce el movimiento en Y de la cama de trabajo XY. La figura B.19 muestra la forma del elemento y la B.20 sus partes.

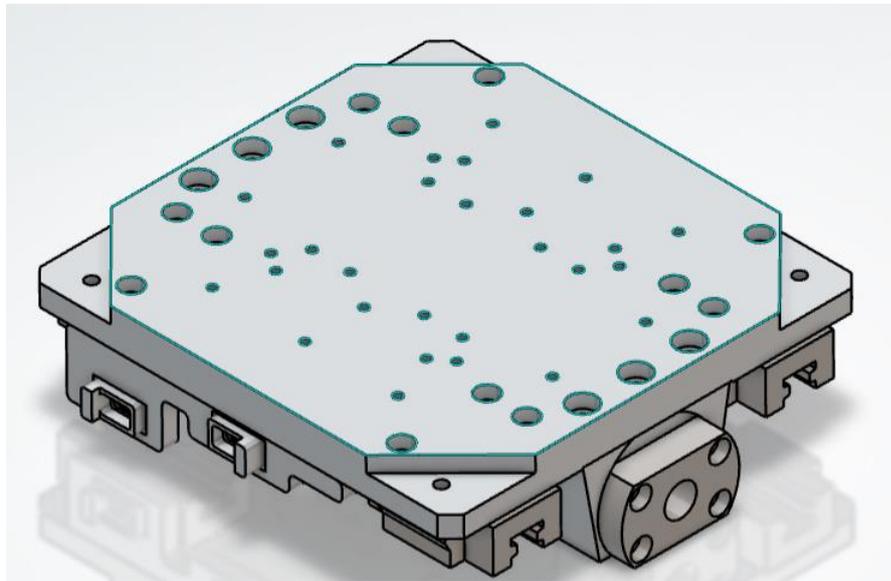


Figura B.19 Mesa trabajo



Figura B.20 elementos de la Mesa trabajo