

**SolidWorks®/**

**MathWorks®** (MATLAB® Simulink® Simscape™)

**Proyecto Final**

Enrique Rocha Espinoza

Joan Carlos Monfil Huitle

José Antonio Tapia Tapia

## ÍNDICE

1	CONEXIÓN SOLIDWORKS CON MATHWORKS.....	3
1.1	Descarga de datos.....	3
1.2	Habilitar plugin en SolidWorks.....	4
2	BLOQUE SIMULINK – PS CONVERTER.....	8
3	BLOQUE REPEATING SEQUENCE INTERPOLATED .....	11
4	BLOQUE DE TRANSFORMACION .....	13
5	BLOQUE DE PIEZA (ESLABON) .....	16
6	BLOQUE DE UNION ROTACIONAL.....	18
7	BLOQUE DE UNION PRISMATICA.....	21
8	COLOCAR SOLIDOS EXTERENOS AL ROBOT .....	23
9	COLOCAR EN LA BRIDA UNA HERRAMIENTA .....	27
10	Referencias .....	30

## Descripción del proyecto

Este proyecto fue parte de evaluación de la materia de *Dinámica y Cinemática de Robots* en el cual se solicitó a los estudiantes utilizar las herramientas de software de MathWorks® para analizar el robot PUMA 560 obteniendo su cinemática directa e inversa, para posteriormente simular el funcionamiento que tendría el robot, para realizar una tarea de *Pick and Place* a través del programa de modelado físico/digital Simscape™ con su control dado en bloques en Simulink® exportando el modelo del robot desde un programa de diseño por computadora CAD, el cual realizaba el ecosistema de MathWorks®, para este proyecto el CAD utilizado fue SolidWorks®.

## 1 CONEXIÓN SOLIDWORKS CON MATHWORKS

### 1.1 Descarga de datos

Antes de instalar cualquier plugin es necesario contar con una licencia activa del CAD compatible con el plugin de Simscape™ Multibody™.

Para tener una conexión entre el software CAD y MathWorks® se tiene que instalar los paquetes de datos necesarios para el plugin de Simscape™ Multibody™ siguiendo los pasos mostrados a continuación:

#### PASO 1:

Verificar que los programas a utilizar tengan la misma estructura que tiene la computadora de su selección a 32 o 64 bits.

#### PASO 2:

Abrir el siguiente enlace para descargar el plugin otorgado por MathWorks [https://la.mathworks.com/campaigns/offerings/download\\_smlink.html](https://la.mathworks.com/campaigns/offerings/download_smlink.html) (es necesario contar con una cuenta activa de MathWorks®).

#### PASO 3:

Una vez seguido los pasos para autenticar las credenciales de quien requiere el plugin, seleccionar la versión de MATLAB instalada en su computadora (ej. MATLABR2022b).

#### PASO 4:

Una vez seleccionada la versión de MATLAB nos desplegará un menú, en el cual debemos seleccionar los archivos dependiendo del sistema operativo de nuestra

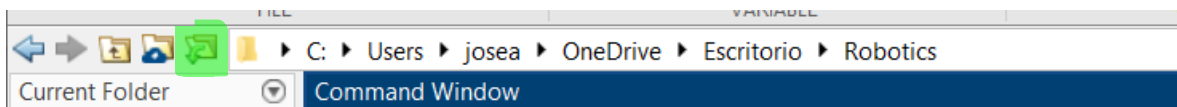
computadora, es importante descargar y guardar ambos archivos en una carpeta independiente y recordar la ubicación de la carpeta dentro de la computadora.

#### ✓ Simscape Multibody Link 7.6 – Release 2022b (Simscape Multibody 7.6)

Simscape Multibody 7.6	
Win64 (PC) Platform	<a href="#">smlink.r2022b.win64.zip</a> <a href="#">install_addon.m</a>
UNIX (64-bit Linux)	<a href="#">smlink.r2022b.glnxa64.zip</a> <a href="#">install_addon.m</a>
Mac OS X (64-bit Intel)	<a href="#">smlink.r2022b.maci64.zip</a> <a href="#">install_addon.m</a>

#### PASO 5:

Correr MATLAB como administrador y añadir el *path* de la carpeta donde se guardaron los archivos de instalación, dando clic sobre la carpeta con flecha verde, así seleccionamos el *path* de la carpeta destino (ubicación de los archivos de instalación).



#### PASO 6:

Posteriormente dentro del *command prompt* escribir el siguiente comando ***install\_addon('zipname')*** donde “zipname” es el nombre del archivo .zip descargado previamente y guardado en la carpeta seleccionada.

#### PASO 7:

Para que conectar el CAD con el plugin es necesaria la conexión entre MATLAB y el plugin, para realizar esta acción es necesario escribir en el *command prompt* el siguiente comando: ***regmatlabserver***.

### 1.2 Habilitar plugin en SolidWorks

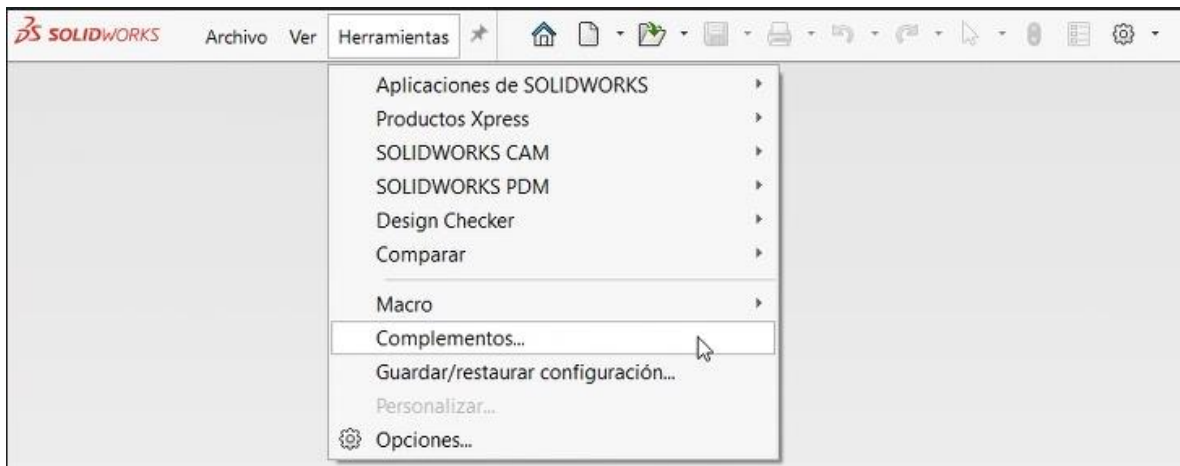
Ahora se habilita el plugin dentro de SolidWorks para ello es necesario contar con una licencia activa.

#### PASO 8:

Antes de abrir SolidWorks en el *command prompt* de MATLAB escribimos el siguiente comando: ***smlink\_linksw*** para habilitar la conexión entre softwares.

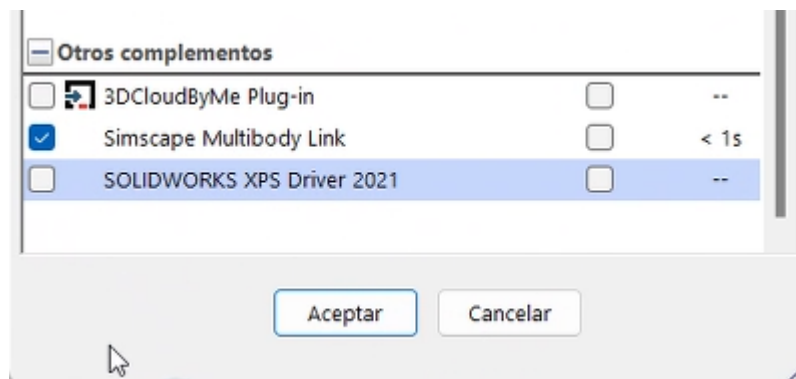
#### PASO 9:

Una vez activa la conexión entre softwares abrir SolidWorks en la parte superior izquierda encontramos el apartado “herramientas” lo seleccionamos y se nos desplegará un menú, en donde buscamos el apartado “complementos”.



PASO 10:

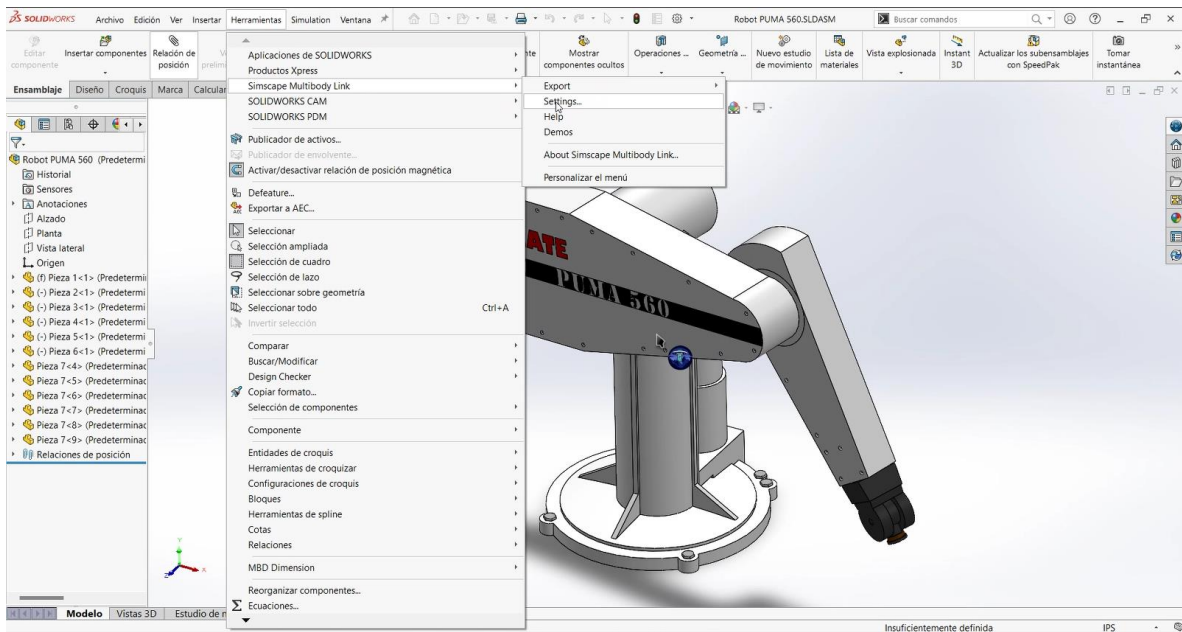
Buscamos el apartado o submenú “otros complementos” y a continuación activamos el plugin “Simscape Multibody link” es importante encenderlo cada que abrimos el software de SolidWorks®, ya que cuando el programa se cierra se desactiva.



PASO 11:

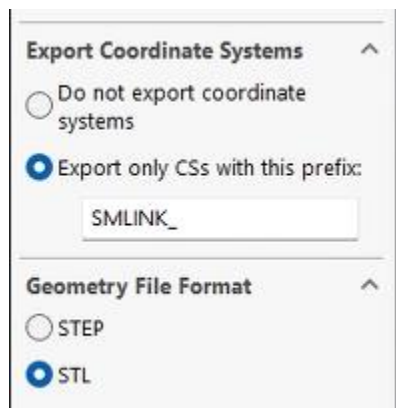
Para este paso es necesario ya contar con la pieza o el ensamble que queremos exportar a Simscape, en este proyecto ya contábamos con el diseño del robot PUMA560, se abre el CAD y a continuación revisamos que el plugin esté encendido.

En la parte superior izquierda buscamos “herramientas” se desplegará un menú y en los primeros elementos del menú se debe visualizar “Simscape Multibody Link”.



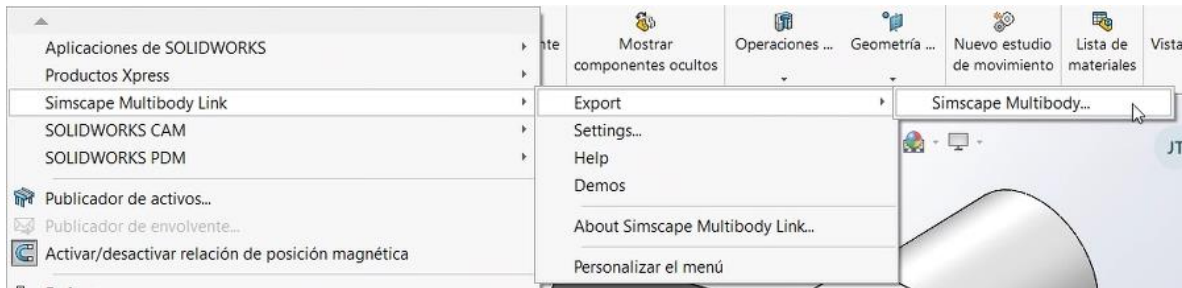
## PASO 12:

Seleccionamos “Settings...” para configurar como creará el modelo para Simscape, abrimos el menú y seleccionamos en el apartado de *Export Coordinate Systems* el “Export only CSs with this prefix SMLINK\_” y en el apartado de *Geometry File Format* seleccionamos “STL” esto nos dará los archivos para Simulink y los diagramas a bloques de cada articulación de nuestro ensamble o robot.



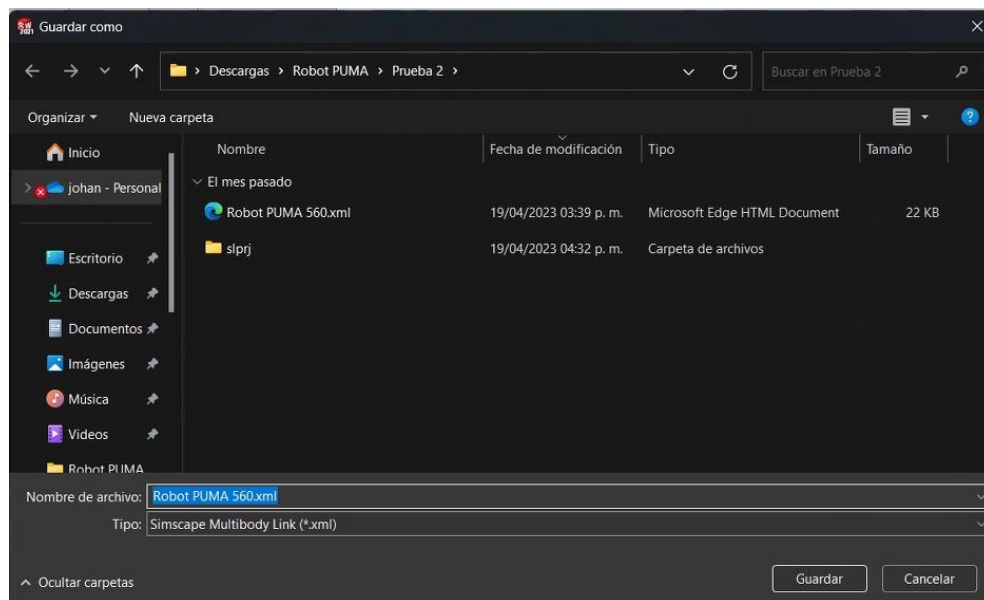
## PASO 13:

Volvemos a seleccionar el menú de “Simscape Multibody Link” dentro de “herramientas” seleccionando en esta ocasión “export” y posteriormente “Simscape Multibody...”



#### PASO 14:

Aparecerá una ventana del sistema operativo en donde te solicita guardes el archivo con extensión .xml, este archivo será leído por MathWorks para su uso interno en Simulink y Simscape, se recomienda guardar en una carpeta dentro de la carpeta donde tienen los archivos de descarga.



#### PASO 15:

Dentro de la carpeta con la extensión .xml se descargaron los archivos STL, estos son la estructura (eslabones) del modelo a simular en Simscape.

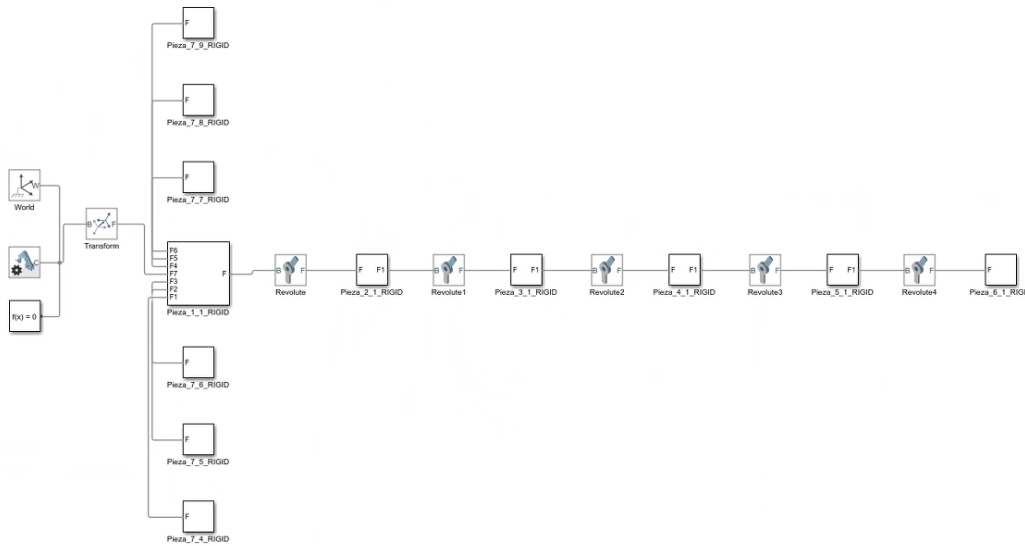
#### PASO 16:

Dentro de MATLAB® en el *path* de la carpeta seleccionar la carpeta donde guardamos el archivo .xml y los STL.

#### PASO 17:

Importar el archivo .xml para trabajar con él, en el *command prompt* de MATLAB® utilizar el comando `smimport('nombre_del_xml')` en donde el `nombre_del_xml` es el

nombre que tiene el archivo mencionado y una vez dando correr obtendremos el diagrama de bloques de cada una de las articulaciones o eslabones del sistema exportado.



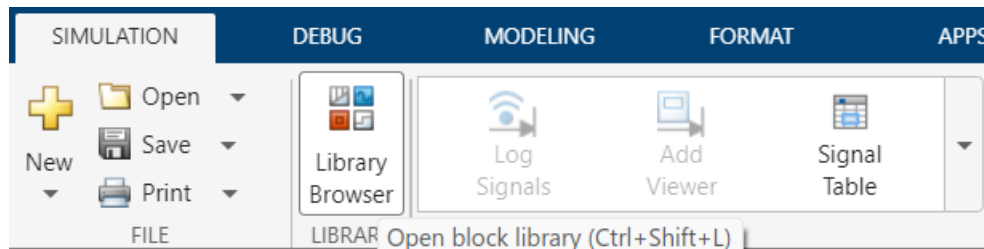
## 2 BLOQUE SIMULINK – PS CONVERTER

Es necesario agregar un bloque, el cual nos ayudara para poder interconectar las señales, esto se debe porque la paquetería de Simscape trabaja con señales físicas, por lo cual no se puede conectar directamente o interconectar con señales o bloques de simulink. Es por ello por lo que se utiliza un convertidor, el mismo se encuentra en la librería de Simscape (Library Browser), Este bloque nos ayuda a convertir las señales de simulink a Simscape o viceversa.

Para utilizar este bloque se hace lo siguiente:

PASO 1:

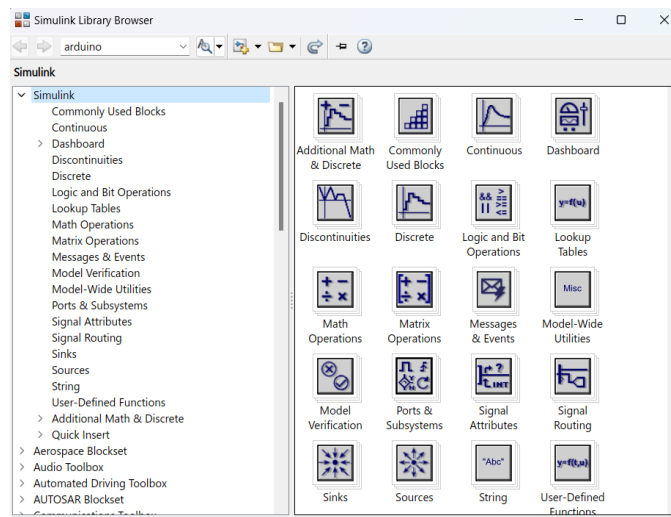
Identificar la librería



PASO 2:

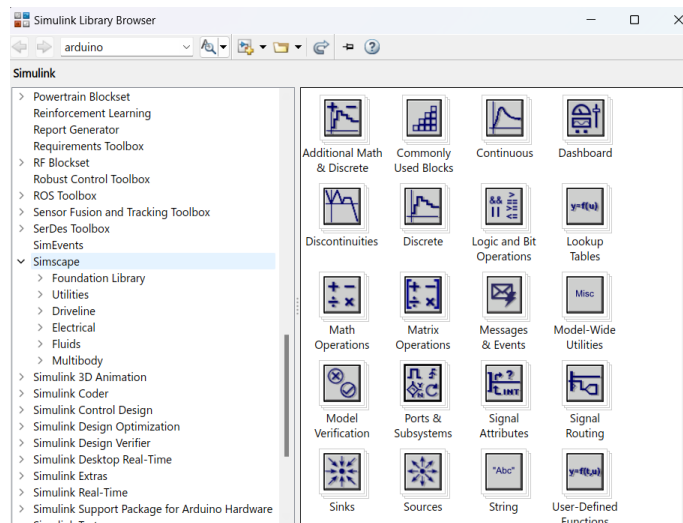


Una vez identificado se dará click y se abrirá una ventana como se muestra a continuación:



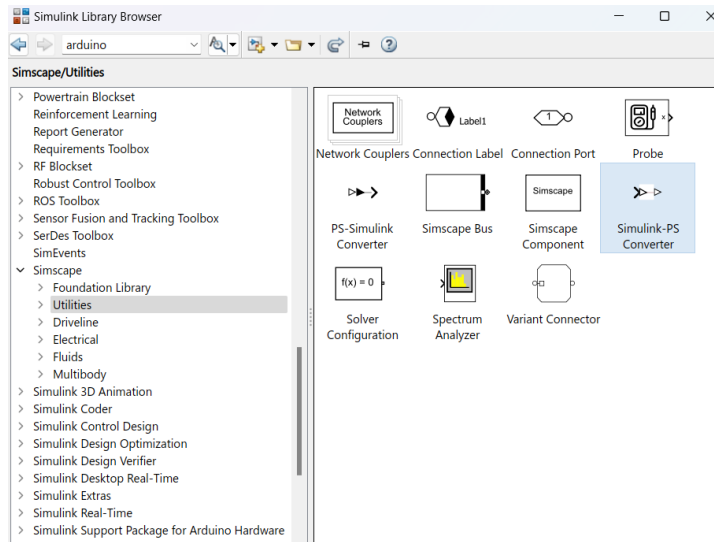
PASO 3:

En la ventana abierta se buscará el apartado de “Simscape”.



PASO 4:

En el apartado de Simscape se dará clic a “Utilities”; una vez seleccionada esa opción se usará el bloque “Simulink – PS Converter”.



Una vez colocado del bloque de “Simulink – PS Converter”, se deberá configurar para poder reconocer las señales y convertirla de forma correcta, para ello se realiza lo siguiente:

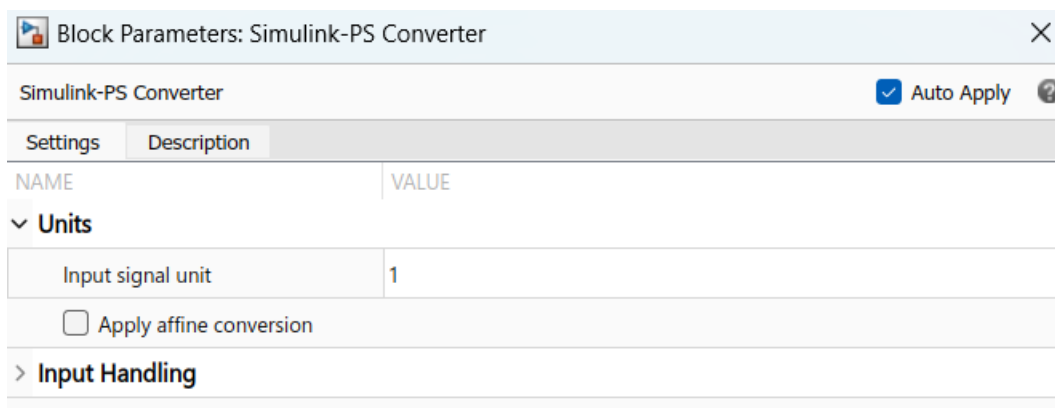
PASO 1:

Se selecciona el bloque “Simulink – PS Converter”



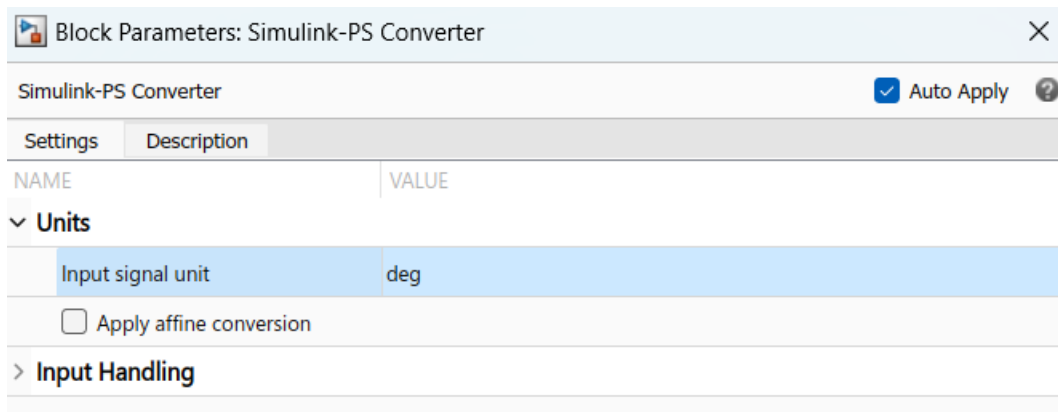
PASO 2:

Una vez selecciona se abrirá una ventana



PASO 3:

Se modificará en “Units” el “Input signal unit”, esto para especificar las unidades dimensionales en las que se estará trabajado, en este caso se trabajará en grados, como no está la opción se escribe “deg” de forma manual.



### 3 BLOQUE REPEATING SEQUENCE INTERPOLATED

Teniendo en cuenta lo mencionado se utilizarán señales o una entrada que nos proporcione señales, en este caso utilizaremos un bloque que la entrada sea de grados, grados los cuales nos sirven para mover las articulaciones, como se mencionó antes, las articulaciones utilizadas trabajan en grados.

Para poderle dar señales a la revolote se utiliza el bloque “Repeating Sequence Interpolated”; este bloque genera una secuencia periódica de tiempo discreto basada en los valores de los parámetros Vector de valores de tiempo y Vector de valores de salida. Tal vez suene muy complicado esto, pero en otras palabras es; utiliza vectores de posición (grados) y vectores de tiempo (tiempo en lo que se moverá y llegara).

Para poderle dar movimiento correcto a las articulaciones y con uso del bloque “Repeating Sequence Interpolated”, se hace lo siguiente:

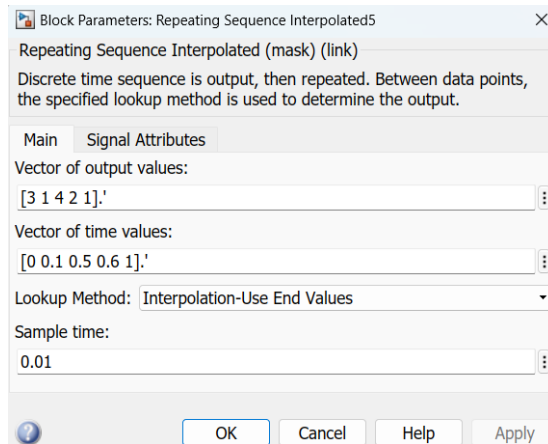
PASO 1:

Se selecciona el bloque “Repeating Sequence Interpolated”



PASO 2:

Al seleccionarlo (doble clic), se abrirá una ventana, en dicha ventana podemos visualizar principalmente el “Main” y “Signal Attributes”.



### PASO 3:

Solo se utilizará el “Main”, en este se puede ver “Vector of output values”, “Vector of time values”, “Lookup Method”, “Sample time”; respecto al “Signal Attributes”, no lo utilizaremos ya que no nos brinda ayuda para utilizar las articulaciones

Para la parte de los vectores de posición (Vector of output values)

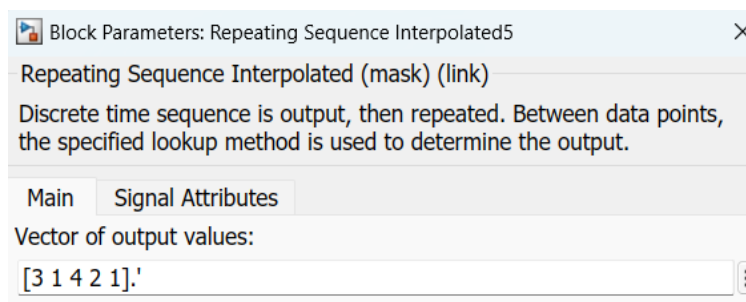
### PASO 1:

Se escribe en corchetes, punto y una apostrofe, dentro de los corchetes irán los valores en grados, los cuales por ser grados se manejarán de 0 a 360 grados, estos pueden ser positivos o negativos, todo dependerá de la configuración que le de uno.

Nota: no hay mínimo o un máximo a utilizar, con que sea la misma cantidad de valores en grados que los valores en tiempo está bien: ejemplo: vector de posición 5 y vector de tiempo 5, si funcionara si los valores no son iguales no correrá y marcará error.

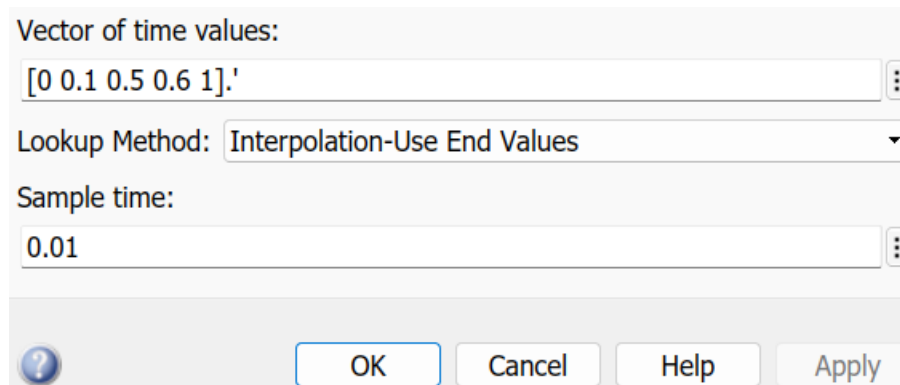
### PASO 2:

Así mismo es mucha ayuda el saber que los valores no marcaran error si no son subsecuentes o no son escalados



### PASO 3:

Respecto al “Vector of time values” se pondrá el valor del tiempo, estos deberán tener una secuencia y deberá ser escalonado; es decir, no se puede poner un número inferior al antes puesto. El valor o matriz de los numero de tiempo es dado por el uno ya que dependerá del tiempo de trabajo a simular; es necesario recordar que debe ser del mismo tamaño de la matriz de los grados (Vector of output values).



Vector of time values:

[0 0.1 0.5 0.6 1].'

Lookup Method: Interpolation-Use End Values

Sample time:

0.01

? OK Cancel Help Apply

### PASO 4:

Respecto a las demás opciones, se dejará tal cual aparezca, ya que nos ayuda demasiado en la configuración, el tiempo (Sample time) se quedare asi ay que nos brinda una simulación decente y en cuanto al “Lookup Method” se queda asi ya que se darán datos en forma de señales.

Este bloque al utilizar señales y con la configuración en las revolutes hará que el robot se mueva de forma correcta y sin errores, así mismo es necesario decir que el movimiento se quedara pausada si pones los mismos datos repetidas veces, pero es parte de la simulación, no es de preocupar

Nota: mientras más valores tengan los vectores

## 4 BLOQUE DE TRANSFORMACION



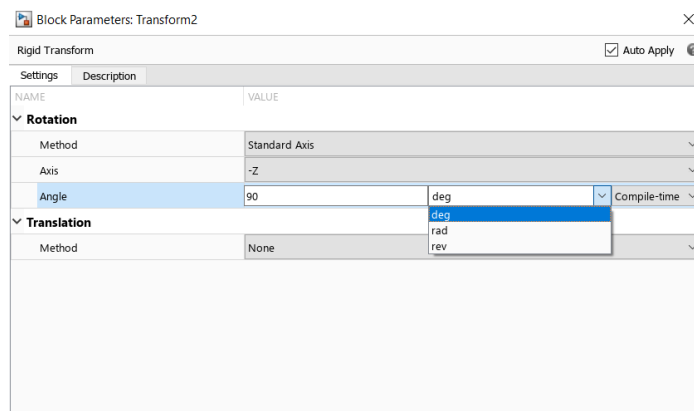
Este bloque nos permite realizar una rotación o traslación de un eje de referencia, estas rotaciones y traslaciones los hace en base a un eje que elijamos (X Y Z), por ejemplo, en la imagen de abajo definimos que queremos una rotación en el eje z.

NAME	VALUE
<b>Rotation</b>	
Method	Standard Axis
Axis	-Z
Angle	90 deg
<b>Translation</b>	
Method	None

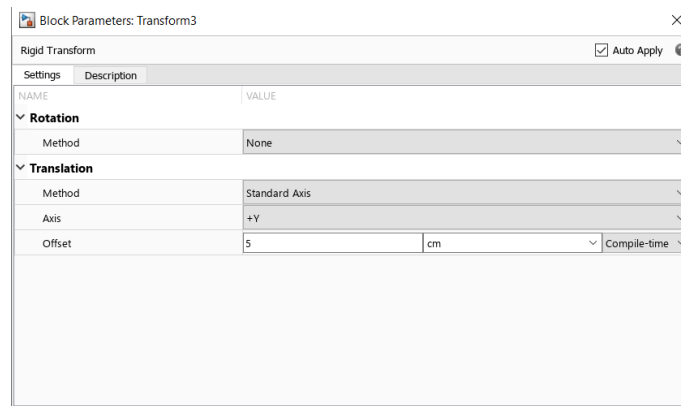
Para lograr una rotación en un eje debemos seleccionar en el apartado method la opción de Standard Axis, para que nos permita elegir un eje específico, se nos habilitara el apartado axis donde nosotros podemos seleccionar el eje y su dirección de rotación, si es Z rotara positivamente en sentido contrario a las manecillas del reloj, si seleccionamos -Z este rotara en sentido de las manecillas del reloj.

NAME	VALUE
<b>Rotation</b>	
Method	Standard Axis
Axis	-Z
Angle	-Z
<b>Translation</b>	
Method	-Z

En el apartado Angle nosotros definimos los grados de rotación que deseamos, este valor puede ser en grados, radianes o revoluciones, en el ejemplo se decide rotar 90 grados negativos en el eje Z.



Para realizar una traslación debemos dar click en la flecha junto a la palabra Traslacion para que podamos ver el menú, de igual forma nosotros debemos de elegir en la opción method < standar Axis, deberemos elegir el eje en el que nos queremos desplazar, al igual que su dirección, en el apartado offset debemos colocar la distancia que se va a mover, podemos cambiar las unidades de medida, por ejemplo, cm, m, in, mm etc. En la imagen se selecciona un desplazamiento en Y positivo de 5 cm.



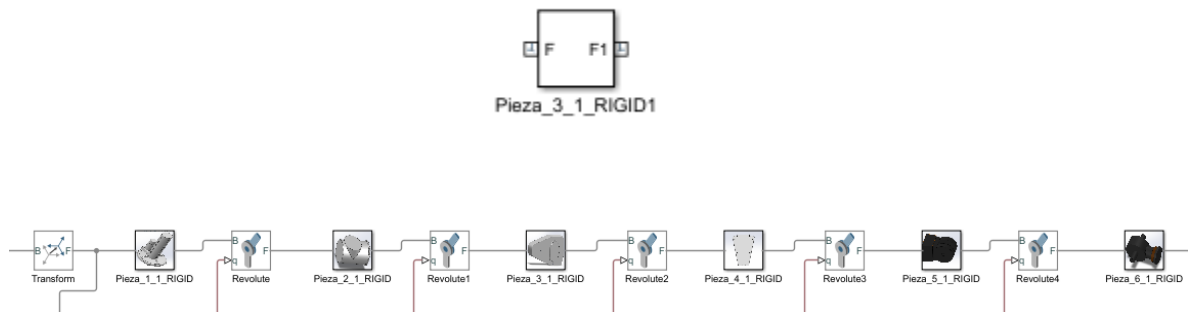
Por último, podemos unir varias transformaciones para llevar un eje de referencia a otro punto, esto quiere decir que de manera secuencial Simscape va a ir aplicando cada transformada al eje, hasta acabar el lazo, para que al final cree otro eje de referencia nuevo de salida.



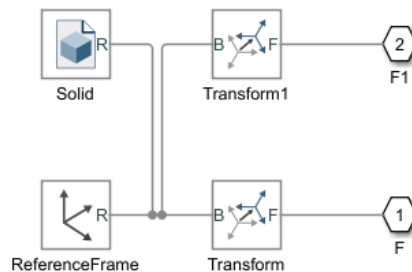
Recomendación: se recomienda no modificar las transformaciones espaciales creadas de la exportación del archivo CAD ya que se podría ubicar el eslabón en una posición imposible físicamente, por ejemplo, sobreponiéndose a otra pieza o volando en el espacio, solo modificar cuando sea estrictamente necesario.

## 5 BLOQUE DE PIEZA (ESLABON)

Este bloque puede crear un eslabón el cual será parte de nuestro robot, al importar el CAD se crearán automáticamente un numero de bloques equivalente al número de eslabones del robot, en este caso al usar el robot PUMA560 el cual contine 6 grados de libertad, contiene 6 eslabones por lo que aparecerá 6 piezas unidas por orden de ensamblaje.

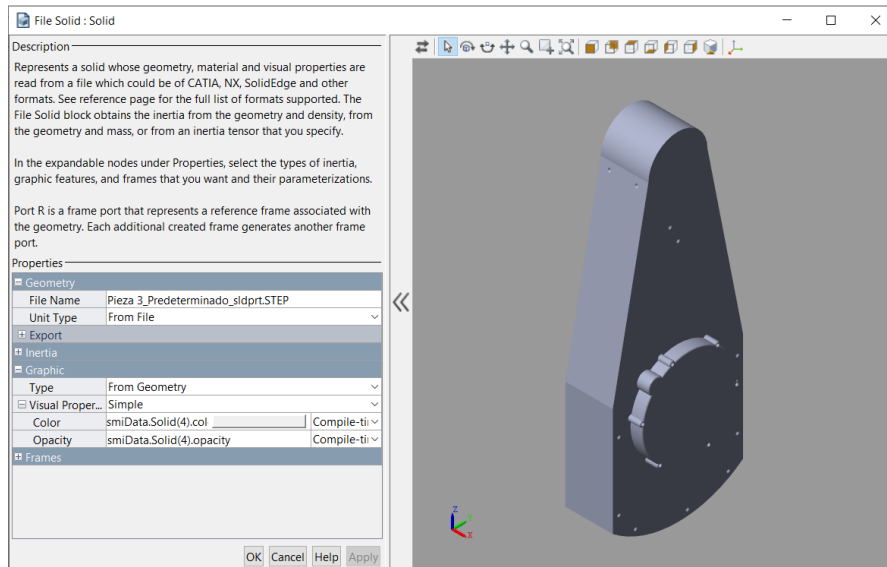


Dentro de la pieza podemos observar un bloque llamado ReferencerFrame, este define un marco al que se pueden hacer referencia otros marcos en una red o al que se pueden adjuntar bloques. Los marcos de referencia no son necesarios, pero sirven como una comodidad de modelado y diseño, este generalmente viene desde el archivo CAD.



De igual forma tenemos un bloque el cual se llama Solid, este contiene el CAD de la pieza, gracias a este podemos ver en nuestra simulación el eslabón dibujado en 3D, en este bloque se puede meter cualquier plano 3D, para ello debemos ingresar un archivo .STEP en el apartado file name, también se puede cambiar el color del eslabón abriendo la pestaña graphic y dando clic en color. En la parte inferior izquierda podemos ver nuestro eje de referencia el cual está representado por el bloque ReferenceFrame.

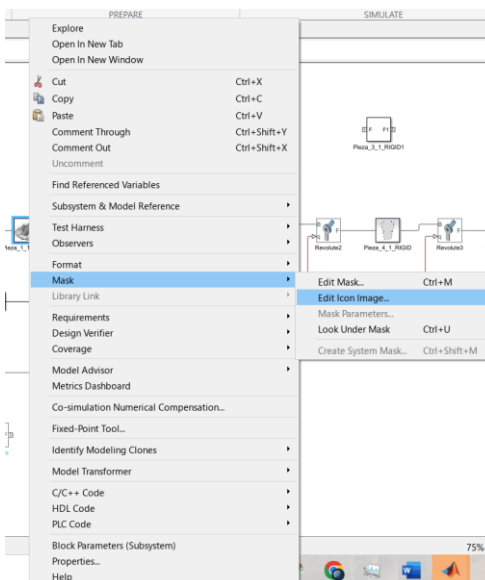




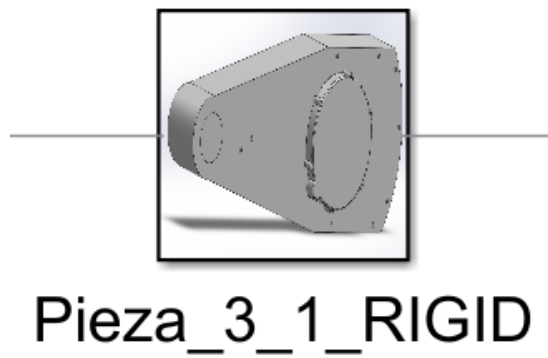
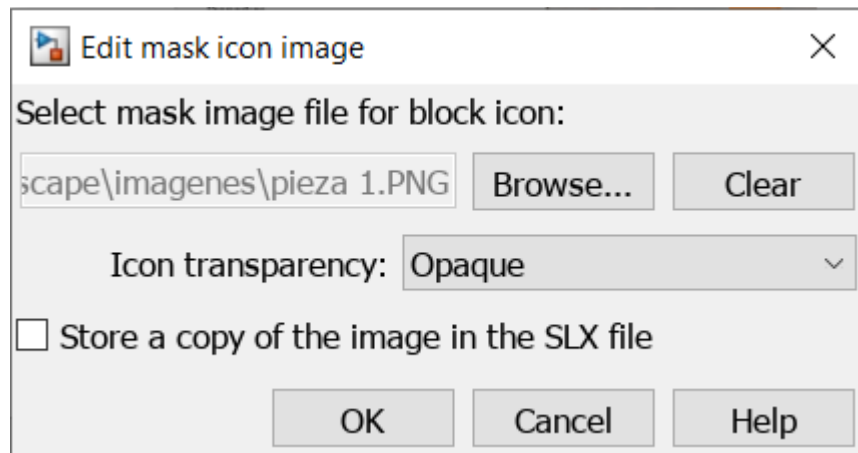
Dentro del bloque de pieza también tenemos el segmento llamado F y F1 los cuales nos indican el eje de referencia de donde empieza la pieza (F) y donde acaba esta (F1), es por ello que antes aparecen transformaciones espaciales, ya que el eje inicial del CAD no necesariamente debe estar en el inicio del eslabón, por lo que debemos de llevar estos puntos a nuestras posiciones finales.

F va a ser la entrada de la pieza por lo que anteriormente debe de ir un eje de referencia mundo o una unión, ya sea rotacional o prismática, F1 es nuestra salida por lo que conectaremos a esta una unión que dará paso a al siguiente eslabón.

De igual forma podemos ponerle una máscara nuestro bloque, para ello seleccionamos el bloque, clic izquierdo> mask> Edit icono image.



En la siguiente ventana seleccionamos nuestro archivo jpg y damos ok, inmediatamente esa imagen estará sobre nuestro bloque como se ve en la imagen de abajo, esto nos ayuda a organizar las piezas y saber que bloque corresponde a cada eslabón.

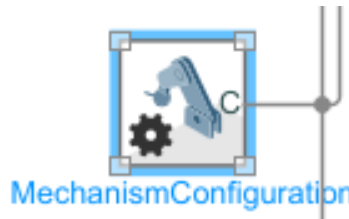


## 6 BLOQUE DE UNION ROTACIONAL

Una vez obtenido la conexión entre SOLID WORKS y SIMSCAPE, obtendremos el archivo del diagrama de bloques, cuyo archivo es un **.slx**.

En este archivo (diagrama de bloques) se empezará a configurar y agregar herramientas del mismo Simscape, con la finalidad de dar movimiento e instrucciones a las eslabones y articulaciones respectivamente para así realizar la manipulación correcta o buscada, en este caso se busca elaborar un pick and place con el brazo robótica “PUMA 560”.

Al principio de exportar el robot a Simscape y darle run (correr), se puede ver que el robot se moverá y actuara de forma extraña, se moverá como loco, esto se debe ya que los bloques de las articulaciones no tienen un número específico, sino que tiene un valor que solo el programa de MATLAB reconoce, este valor está escrito con letras y no números; es por ello que estas letras obedecerán la configuración que esté en ese momento, cuya configuración será el dato de gravedad (-9.80665), este dato aparecerá en el bloque “Mechanism Configuration”. Así mismo es importante mencionar que el valor de gravedad hace que se mueva incontroladamente el robot, de hecho, actúa como un sistema de caos.



Bloque Mechanism Configuration.

Para empezar a manipular el robot y darle orientación deseada o producir un movimiento ideal al mecanismo es importante configurar las articulaciones, de tal forma que el movimiento sea dado por el usuario o una entrada; así mismo también el torque o fuerza se pueda calcular de forma automática.

Para lograr lo antes mencionada se realiza lo siguiente:

#### PASO 1:

Se abre la revoluta a configurar.



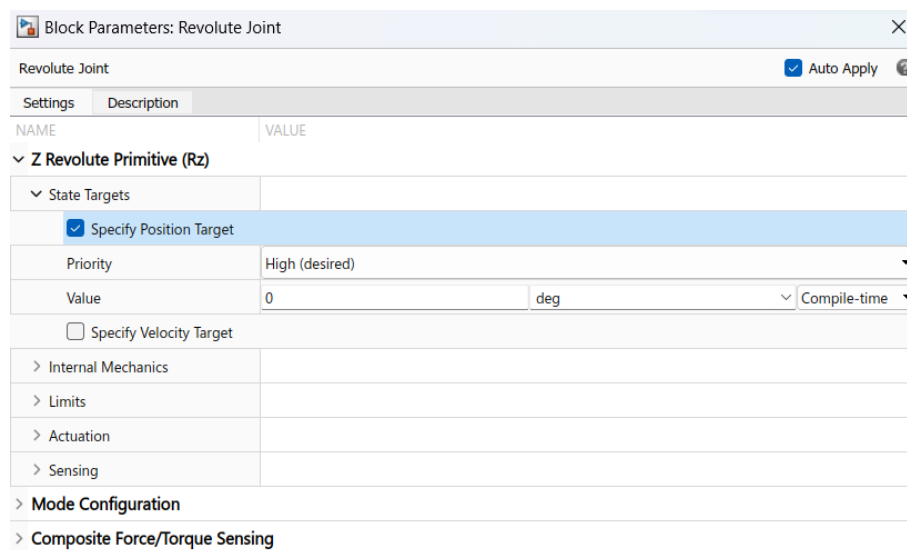
#### PASO 2:

Se abrirá una ventana de configuración.



### PASO 3:

Se selecciona la opción de State Targets y se selecciona la casilla Specify Position Target, y se en la opción de “Value” se configura con 0, esto se debe para que en la articulación seleccionada se coloque en posición 0 junto con el eslabón; así mismo es importante configurar en que parámetros se trabajara, en este caso se hará con grados, como no está la opción de grados, se escribirá la palabra “deg”.



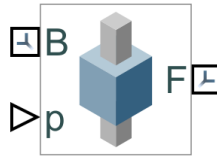
### PASO 4:

Al igual se deberá configurar la “Actuation”, esto como se mencionó para que sea calculada de forma automática.

### PASO 5:

Se selecciona la opción de “Actuation” y se seleccionara la casilla de “Torque”, y se escogerá “Automatically Computed”, esto sirve para obtener los cálculos de forma automática y sea configure de forma rápida y adecuada el brazo robótico.





## Prismatic Joint5

Dentro de él podemos observar varias opciones, se puede confirmar la afirmación de que esta union se mueve en el eje Z. para agregarle una entrada a la prismática debemos dar clic en Actuation, en el apartado de Force elegimos la opción de Automatically Computed para que Matlab nos calcule este valor de manera automática, en el apartado de Motion vamos a elegir la opción de Provided by input, esto hará que se agregue una entrada para poder conectar nuestra señal.

De igual forma podemos seleccionar los límites de esta unión, por ejemplo, si nuestro limite físico es 5cm con traslación positiva, en el apartado de upper limit colocaremos 5 y elegiremos la unidad en cm en lugar de metros, de la misma manera se hace para el limite inferior, ya que hay prismáticas que también pueden avanzar hacia atrás, en el apartado lower limit pondremos el valor máximo que recorre nuestra prismática en el eje Z negativo.

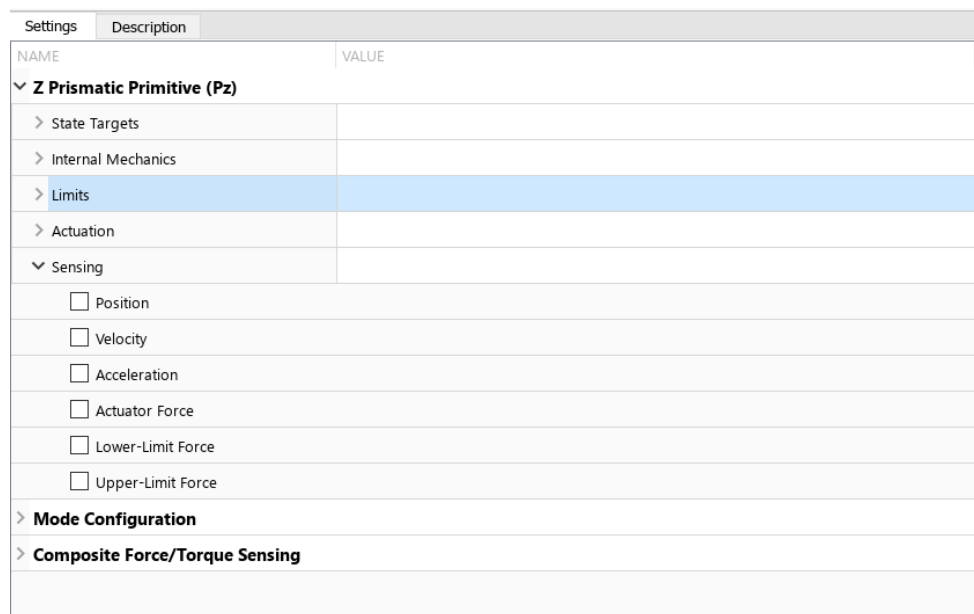
Block Parameters: Prismatic Joint5

Prismatic Joint ☒ Auto Apply

Settings	Description
NAME	VALUE
<b>▼ Z Prismatic Primitive (Pz)</b>	
> State Targets	
> Internal Mechanics	
▼ Limits	
<input checked="" type="checkbox"/> Specify Lower Limit	
Bound	-1 m Compile-time
Spring Stiffness	1e6 N/m Compile-time
Damping Coefficient	1e3 N/(m/s) Compile-time
Transition Region Width	1e-4 m Compile-time
<input checked="" type="checkbox"/> Specify Upper Limit	
Bound	1 m Compile-time
Spring Stiffness	1e6 N/m Compile-time
Damping Coefficient	1e3 N/(m/s) Compile-time
Transition Region Width	1e-4 m Compile-time
▼ Actuation	
Force	Automatically Computed
Motion	Provided by input
> Sensing	
> Mode Configuration	
> Composite Force/Torque Sensing	

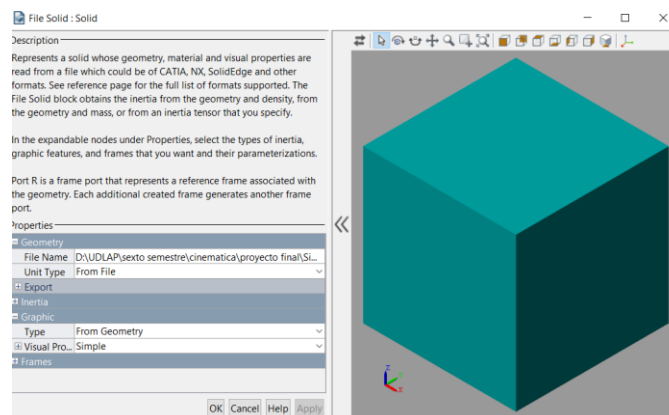
De igual forma podemos agregarle salidas al bloque, esto para que nos dé graficas de que está pasando en la articulación, nos puede arrojar posición, velocidad, aceleración,

torque, y los limites inferiores y superiores, al crear la salida debemos conectar un convertidor de variables PS a simulink, seguido de un scope para ver la graficas.

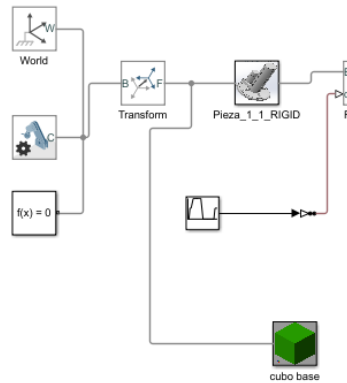


## 8 COLOCAR SOLIDOS EXTERENOS AL ROBOT

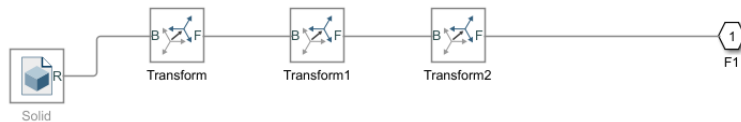
Para colocar solidos que no estén unidos al robot, se debe colocar un bloque de pieza, generalmente estos objetos se van a manipular con el brazo robótico, por lo que pueden ser cubos, esferas, piezas para ensamblar, palos, etc. En este proyecto se movió un cubo, para ello primero creamos el bloque pieza, dentro en el bloque Solid colocamos el CAD del cubo, además le asignamos un color azul.



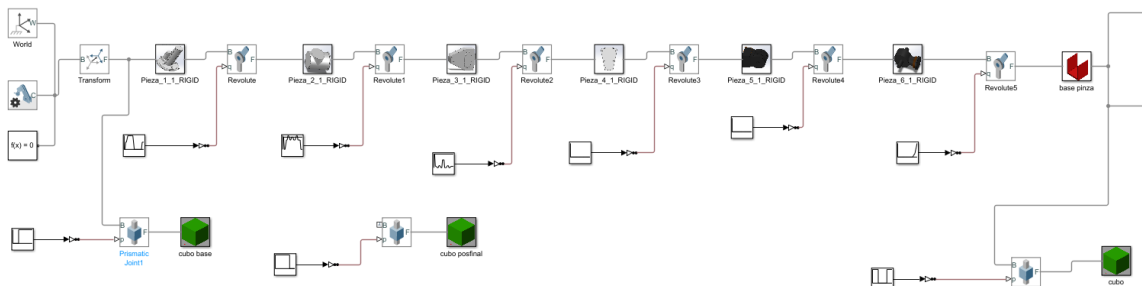
Este solido lo debemos conectar a un eje de referencia, ya sea el eje world o un eje de un eslabón, en este caso se decidió ponerlo con respecto a la base del robot Puma 560, por lo que se conceto a la entrada de esta pieza 1, ya que lo que deseamos es el eje de inicio.



En este momento el cubo aparecerá en el mismo eje que el de inicio de la base por lo que se van a sobreponer, para moverlo de posición debemos ingresar al bloque cubo, con el objetivo colocar transformadas espaciales que nos lleven al punto deseado, en este caso solo se utilizan traslaciones, debido a que en nuestro caso no importa que cara toque el piso, en el caso de otro objeto que si tenga una orientación también deberemos agregar rotaciones para girarlo.

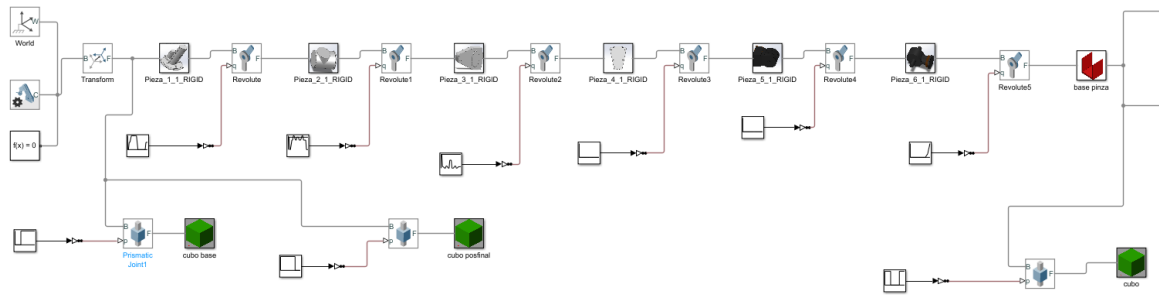


Ahora que ya logramos colocar un objeto externo, debemos hacer que este sea agarrado por el Puma 560 en el momento que este cierre su garra, para no meter bloques de física, crearemos 3 piezas cubos, estas aparecerán en nuestra simulación en diferentes tiempo, el primero aparecerá mientras el TCP del manipulador llega al punto de pick, a continuación este desaparecer y se colocara otro que aparezca justo en el centro de la herramienta para simular que este lo agarra, este segundo cubo estará ligado al eje de referencia del TCP, por lo que conectaremos su entrada a la salida de la brida, por ende seguirá su trayectoria hasta llegar a la posición place.

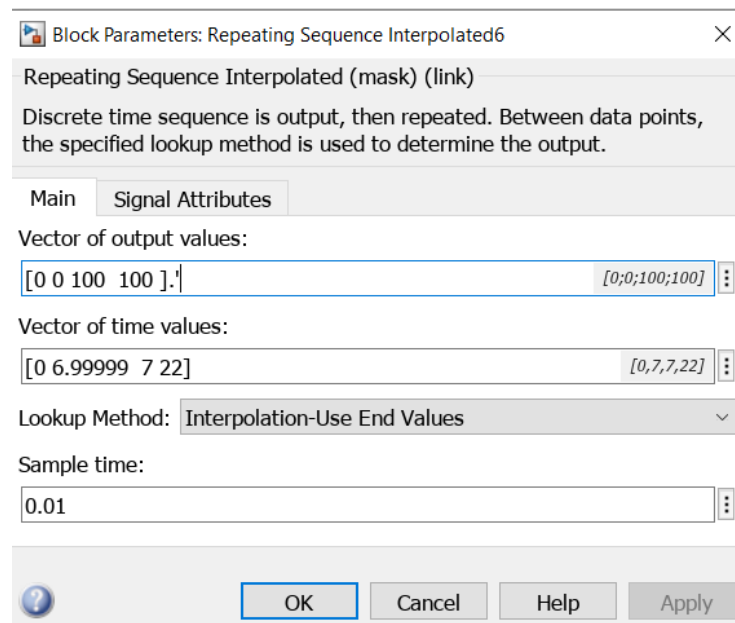




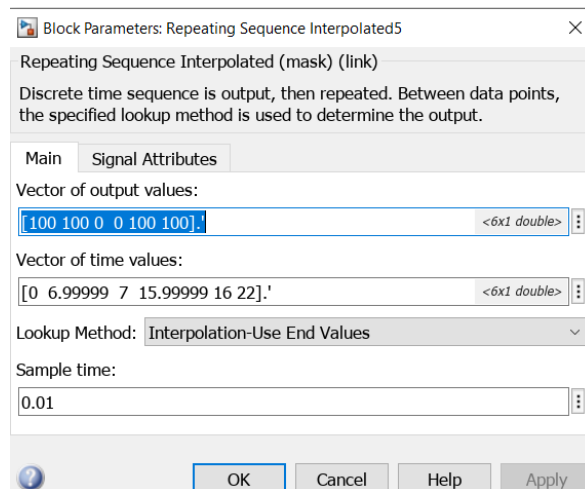
El tercer cubo, el cual es el del centro del diagrama de bloques , igualmente esta con respecto al eje de inicio de la base del robot, por lo que deberá ir conectado al mismo nodo que el primer cubo, dentro de este deberemos colocar transformaciones espaciales para llevarlo a la posición place, este aparecerá cuando el TCP llegue a esta posición, en el mismo segundo el cubo numero 2 desaparecerá y de esta forma se verá como si el robot lo soltara.



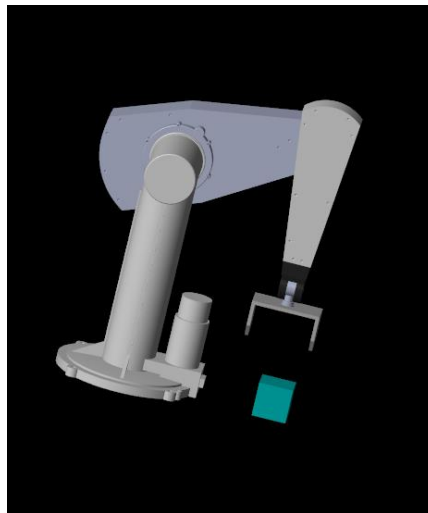
Para hacer que desaparezcan los cubos les colocaremos una unión prismática, esta nos ayudara a asignar un intervalo de tiempo para que este se mueva 0 m, por lo que se quedara estático, en el segundo que deseamos que desaparezca el sólido le asignaremos a la unión prismática que se mueva 100cm para que salga del campo de visión, en teoría no desaparece sino que simplemente se aleja una distancia muy grande para no ser vista, en la siguiente imagen se muestran los vectores de un bloque de señal el cual es el input de nuestra prismática.



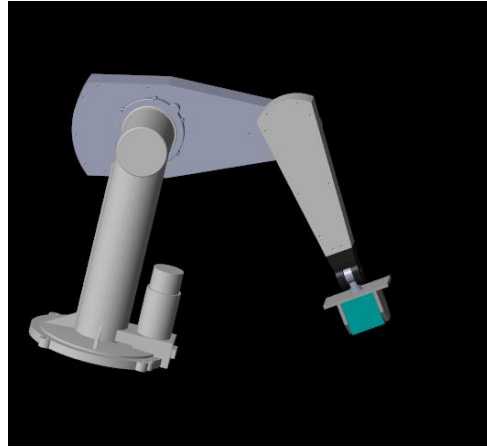
En este ejemplo podemos ver como del segundo 0 a 7 el cubo estará en el campo de visión, este cubo es el inicial por lo que es lógico que se vea, debido a que la herramienta del robot todavía no había llegado a la posición pick, en el según 6.999 una milésima de según antes de que la garra se cierre este activa su unión prismática para desplazarse 100cm, por lo que ya no se verá, como segundo paso la prismática del cubo 2, el cual es el que está en la punta del TCP y antes del segundo 7 estaba a 100 cm de distancia, por lo que con las gráficas decimos que regrese a la posición 0 y este sea agarrado por la garra.



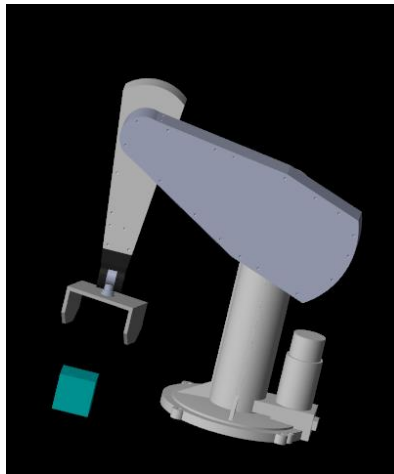
A continuación, se vera la simulación y como estos bloques van apareciendo en sus respectivos tiempos.



Cubo 1



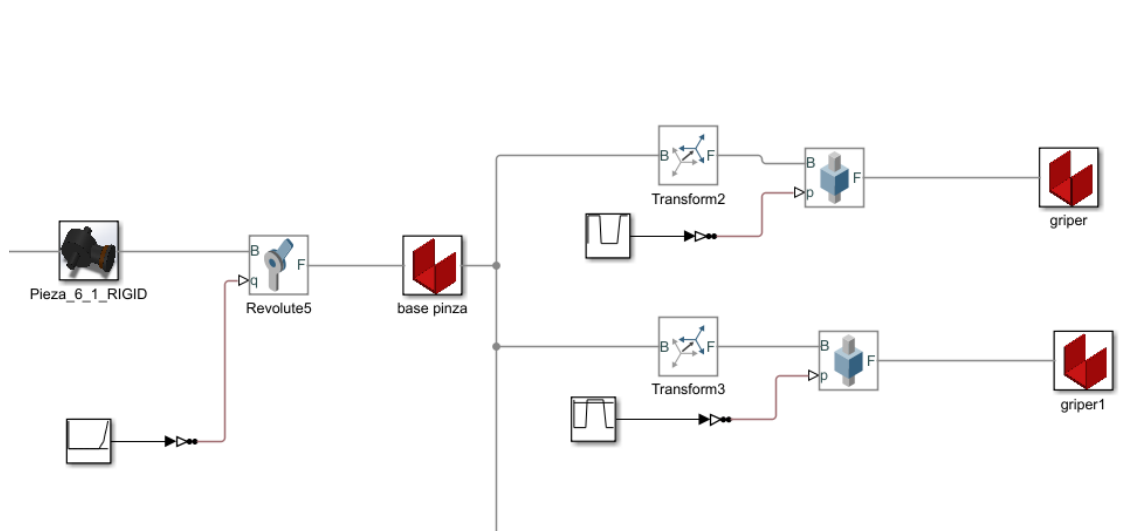
Cubo 2



Cubo 3

## 9 COLOCAR EN LA BRIDA UNA HERRAMIENTA

Para colocar una herramienta en la brida de cualquier robot, deberemos crear un bloque pieza el cual contenga dentro un bloque Solid que este ligado al CAD .step, el cual es el plano 3D de la herramienta a ocupar, en este caso se decidió poner una pinza, el objetivo de esta es que abre y cierre para tomar objetos, por ende, se deberán crear 3 partes de esta herramienta, la base y 2 grippers.



#### PASO 1:

Como primer paso debemos de indicarle a simulink que la última articulación, la cual sería la pieza número 6, tendrá una salida F1, que se desplazara por medio de una transformación espacial que solo se moverá en z ,debido a que es la distancia de la punta de la brida a TCP.

#### PASO 2:

Inmediatamente colocaremos una unión de revolución, para que la herramienta puede girar, de igual manera debemos agregarle una señal como input, para poder controlar los grados de rotación de esta en el tiempo.

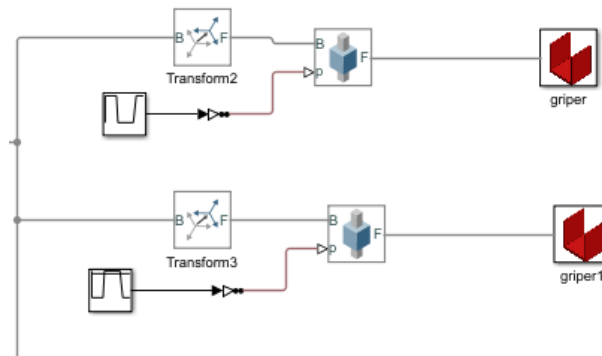
#### PASO 3:

Ahora conectaremos la base de la pinza a esta unión, como el eje de referencia del CAD no está en el centro de la pieza tenemos que realizar una transformación espacial de traslación en X y Y para poder centralarla con la brida.



#### PASO 4:

Conectaremos los 2 grippers a la base de la pinza, de igual forma debemos conectar una transformación de traslación, esto debido a que al momento de colocar la pieza esta aparecerá en el centro de la herramienta, y el gripper debe de estar al extremo de ambos lados, ya que son 2 grippers.

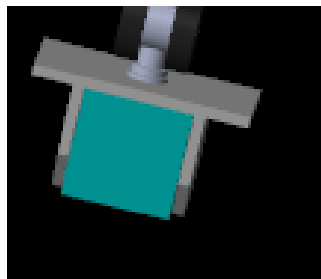


#### PASO 5:

Para que estos 2 gripper se muevan, los uniremos con una unión prismática, cada gripper va a tener su unión en dirección diferente para que se junten en el centro moviéndose un cierto número de cm, por ello también agregaremos señales para decirle que en la posición Pink y la posición place, realice el recorrido equivalente a la cantidad necesaria para agarrar el cubo.

#### PASO 6:

Ahora se mostrar la simulación de como la piza abre y cierra.



## Referencias

Link de GitHub: <https://github.com/JoanCarlosMonfilHuitle/Pick-and-place-Simscape-y-matlab->

“VisibleBreadcrumbs,” *Mathworks.com*, 2023.

<https://la.mathworks.com/help/sm/ref/prismaticjoint.html;jsessionid=995d920c3d69fa5356597535999d> (accessed May 05, 2023).

“VisibleBreadcrumbs,” *Mathworks.com*, 2023.

<https://la.mathworks.com/help/sm/ref/rigidtransform.html> (accessed May 05, 2023).

“VisibleBreadcrumbs,” *Mathworks.com*, 2023.

<https://la.mathworks.com/help/sm/ref/mechanismconfiguration.html> (accessed May 05, 2023).

“VisibleBreadcrumbs,” *Mathworks.com*, 2023.

<https://la.mathworks.com/help/sm/ref/revolutejoint.html> (accessed May 05, 2023).

“VisibleBreadcrumbs,” *Mathworks.com*, 2023.

<https://la.mathworks.com/help/simscape/ref/simulinkpsconverter.html> (accessed May 05, 2023).

“VisibleBreadcrumbs,” *Mathworks.com*, 2023.

<https://la.mathworks.com/help/simscape/ref/solverconfiguration.html> (accessed May 05, 2023).

“VisibleBreadcrumbs,” *Mathworks.com*, 2023.

<https://la.mathworks.com/help/simscape/ug/introducing-the-simscape-block-libraries.html> (accessed May 05, 2023).

Enrique Rocha Espinoza (enrique.rochaea@udlap.mx)

Joan Carlos Monfil Huitle (joan.monfilhe@udlap.mx)

José Antonio Tapia Tapia (jose.tapiata@udlap.mx)