

Proyecto Global Integrador: Control  
Semi-Automático Coordinado de Grúa Portuaria  
de Muelle tipo Pórtico

Marinsalda Pastor, Agustín  
Silva, Víctor David

*Autómatas y Control Discreto, Facultad de Ingeniería, Universidad  
Nacional de Cuyo, Mendoza, Argentina*

Mayo 2022

# Contents

<b>1</b>	<b>Introducción</b>	<b>3</b>
<b>2</b>	<b>Modelado del Sistema</b>	<b>3</b>
2.1	Traslación del Carro . . . . .	4
2.2	Izaje de Carga . . . . .	6
2.3	Análisis de la Carga . . . . .	8
2.3.1	Carga suspendida . . . . .	9
2.3.2	Carga apoyada . . . . .	11
<b>3</b>	<b>Autómata híbrido de control y protección</b>	<b>13</b>
3.1	Nivel 2: Controladores de Movimiento . . . . .	13
3.1.1	Controlador de Traslación del Carro . . . . .	13
3.1.2	Controlador de Izaje de Carga . . . . .	15
3.1.3	Controlador de Balanceo de Carga . . . . .	16
3.1.4	Lazo de realimentación de posición . . . . .	20
3.1.5	Nivel 2 en Codesys . . . . .	21
3.2	Nivel 1: Control Supervisor Global . . . . .	25
3.3	Nivel 0: Función de Seguridad o Protección . . . . .	35
<b>4</b>	<b>HMI</b>	<b>37</b>
<b>5</b>	<b>Resultados</b>	<b>40</b>
<b>6</b>	<b>Conclusiones</b>	<b>46</b>
<b>7</b>	<b>Anexo 1: Comunicaciones OPC UA</b>	<b>46</b>
<b>8</b>	<b>Referencias</b>	<b>49</b>

## Abstract

El presente informe detalla el desarrollo e implementación de un autómata híbrido con su sistema de protección para la operación semiautomática de una grúa porta contenedores portuaria de muelle tipo pórtico. Para ello se plantea el diseño, modelado conceptual y análisis de desempeño a través de una simulación "Model-in-the-loop" del sistema físico y del autómata. Posteriormente se implementa dicho autómata en un entorno de programación en concordancia con la norma IEC 61131-3 luego de lo cual se realizan simulaciones "Software-in-the-loop" junto con una emulación del sistema.

## 1 Introducción

Las grúas portuarias se encuentran en todo puerto digno de actividad comercial. Es un dispositivo necesario y habilitante para la carga y descarga de contenedores. Hoy en día se busca retener a los barcos el menor tiempo posible en el proceso de cargado o descargado, empleando para ello todo tipo de mejoras en los sistemas presentes en las grúas. Desde automatismos hasta sistemas de control más rápidos y eficientes. En general lograr estos tipos de avances involucran simulaciones de los sistemas en algún punto del recorrido, esto permite la comparación entre distintos métodos de control u otras mejoras con una base de referencia. El presente informe logra la simulación de un sistema de estas características mediante el modelado del sistema físico junto con sus controladores de movimientos principales, siendo estos la traslación a lo largo del eje de la grúa y el izado de carga. Se toma en cuenta las características de los motores eléctricos de los accionamientos y los requisitos de funcionamiento del sistema. Además se incorpora un control automático de balanceo de la carga manteniendo el ángulo de la misma dentro de un rango permitido con el fin de optimizar la traslación de la carga. Estos tres controladores constituyen el nivel 2 del autómata.

El nivel 1 del autómata se encuentra constituido por el control supervisor, que es el encargado de la operación semiautomática del proceso. En el modo manual un operario realiza las maniobras delicadas y lentas mientras que el sistema de control genera trayectorias rápidas y coordinadas con los movimientos principales y del balanceo.

El nivel 0 y prioritario esta dado por un autómata a parte, reducido y confiable, encargado de la seguridad de operación y protección ante fallas críticas, de riesgo o de seguridad.

## 2 Modelado del Sistema

Se comienza modelando el sistema físico del carro + el sistema de izaje + la carga. Esto resulta en un modelo dinámico no lineal en tiempo continuo simplificado mediante las siguientes consideraciones:

- Estructura de pórtico totalmente rígida.

- Sistema de izaje de carga totalmente montado sobre el carro ("Machinery on Trolley"), se desplazan horizontalmente juntos.
- Un solo cable de acero equivalente elásticos, sin masa propia, trabajando a tracción (no soporta compresión).
- Carro soportado por la estructura rígida; desplazamiento horizontal sin resbalamiento.
- Resistencia aerodinámica frontal despreciable para el carro y la carga.

## 2.1 Traslación del Carro

Se considera el movimiento del carro en el eje X siendo el sentido positivo del mismo hacia la derecha o alejándose de la base de la grúa yendo hacia el mar.

Límites:

- Posición: [-30.0m; 50.0m].
- Velocidad:  $\pm 4.0\text{m/s}$ .
- Aceleración  $\pm 1.0\text{m/s}$ .

El conjunto mecánico del carro, consiste en el carro, el motor, la reducción y la rueda sobre la cual esta montado el sistema. La dinámica del motor está dado por:

$$J_m \dot{\omega}_m(t) = T_m(t) - T_l(t) - b_m \omega_m(t) \quad (1)$$

donde  $J_m$  es el momento de inercia del eje del motor,  $\dot{\omega}_m(t)$  es la aceleración angular del eje motor,  $\omega_m(t)$  es la velocidad angular del mismo,  $T_m(t)$  es el torque electromagnético,  $T_l(t)$  es el torque de carga y  $b_m$  es el coeficiente de rozamiento viscoso del motor.

La dinámica de la rueda del carro está dada por la siguiente ecuación:

$$J_w \dot{\omega}_w(t) = T_q(t) - T_w(t) - b_w \omega_w(t) \quad (2)$$

donde  $J_w$  es el momento de inercia de la rueda,  $\dot{\omega}_w(t)$  es la aceleración angular de la rueda,  $\omega_w(t)$  es la velocidad angular de la misma,  $T_q(t)$  es el torque motriz,  $T_w(t)$  es el torque de carga de la rueda y  $b_w$  es el coeficiente de rozamiento viscoso de la rueda.

La relación de transmisión entre el eje del motor y la rueda es:

$$r_t = \frac{\omega_m(t)}{\omega_w(t)} = \frac{\dot{\omega}_m(t)}{\dot{\omega}_w(t)} = \frac{T_q(t)}{T_l(t)} \quad (3)$$

Reemplazando (3) en (1) y despejando  $T_q(t)$ :

$$J_m \dot{\omega}_w(t) r_t = T_m(t) - \frac{T_q(t)}{r_t} - b_m \omega_w(t) r_t$$

$$T_q(t) = -J_m \dot{\omega}_w(t) r_t^2 - b_m \omega_w(t) r_t^2 + T_m(t) r_t \quad (4)$$

Reemplazando la ecuación (4) en (2) y reordenando se obtiene:

$$\begin{aligned} J_w \dot{\omega}_w(t) &= -J_m \dot{\omega}_w(t) r_t^2 - b_m \omega_w(t) r_t^2 + T_m(t) r_t - T_w(t) - b_w \omega_w(t) \\ (J_w + J_m r_t^2) \dot{\omega}_w(t) &= T_m(t) r_t - (b_w + b_m r_t^2) \omega_w(t) - T_w(t) \end{aligned} \quad (5)$$

Las relaciones entre la rotación de la rueda y la traslación del carro cuando no hay resbalamiento están dadas por:

$$\begin{aligned} x_c(t) &= R_w \theta_w(t) \\ \theta_w(t) &= \frac{x_c(t)}{R_w} \end{aligned} \quad (6)$$

$$\begin{aligned} \dot{x}_c(t) &= R_w \omega_w(t) \\ \omega_w(t) &= \frac{\dot{x}_c(t)}{R_w} \end{aligned} \quad (7)$$

$$\begin{aligned} \ddot{x}_c(t) &= R_w \dot{\omega}_w(t) \\ \dot{\omega}_w(t) &= \frac{\ddot{x}_c(t)}{R_w} \end{aligned} \quad (8)$$

donde  $x_c(t)$  es la posición del carro en el eje x y  $R_w$  es el radio de la rueda.

La relación entre el torque de carga en la rueda y la fuerza tangencial ( $F_c(t)$ ) que mueve el carro es:

$$T_w(t) = R_w F_c(t) \quad (9)$$

Reemplazando (7), (8) y (9) en (5) y despejando  $F_c(t)$ :

$$\begin{aligned} (J_w + J_m r_t^2) \frac{\ddot{x}_c(t)}{R_w} &= T_m(t) r_t - (b_w + b_m r_t^2) \frac{\dot{x}_c(t)}{R_w} - R_w F_c(t) \\ F_c(t) &= -\frac{(J_w + J_m r_t^2)}{R_w^2} \ddot{x}_c(t) + \frac{T_m(t) r_t}{R_w} - \frac{(b_w + b_m r_t^2)}{R_w^2} \dot{x}_c(t) \end{aligned} \quad (10)$$

La ecuación de movimiento del carro es:

$$m_c \ddot{x}_c(t) = F_c(t) + F_l(t) - b_c \dot{x}_c(t) \quad (11)$$

donde  $m_c$  es la masa de la carga,  $F_l(t)$  es la fuerza de carga y  $b_c$  es el coeficiente de rozamiento viscoso de carro.

La fuerza de carga en función del ángulo de balanceo  $\theta(t)$  es:

$$F_l(t) = F_w(t) \sin \theta(t) \quad (12)$$

donde  $F_w(t)$  es la fuerza elástica del cable sobre el carro.

Reemplazando las ecuaciones (7) y (10) en (11) y reordenando:

$$\left(m_c + \frac{J_w + J_m r_t^2}{R_w^2}\right) \ddot{x}_c(t) = F_w \sin \theta(t) + \frac{T_m(t) r_t}{R_w} - \left(b_c + \frac{b_w + b_m r_t^2}{R_w^2}\right) \dot{x}_c(t) \quad (13)$$

Se pueden representar los parámetros equivalentes de la masa, coeficiente de rozamiento y fuerza como:

$$M_t = m_c + \frac{J_w + J_m r_t^2}{R_w^2} \quad (14)$$

$$b_t = b_c + \frac{b_w + b_m r_t^2}{R_w^2} \quad (15)$$

$$F_t(t) = \frac{T_m(t) r_t}{R_w} \quad (16)$$

Reemplazando (14), (15) y (16) en (13) se obtiene la ecuación dinámica de traslación del carro:

$$M_t \ddot{x}_c(t) = F_t(t) - b_t \dot{x}_c(t) + F_w(t) \sin \theta(t) \quad (17)$$

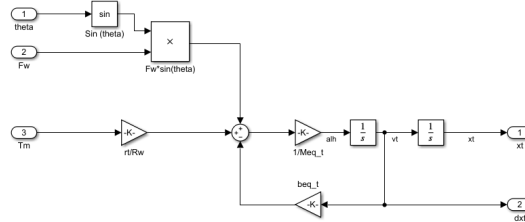


Figure 1: Representación en diagrama de bloques del sistema de traslación del carro

## 2.2 Izaje de Carga

El movimiento de Izaje se realiza sobre el eje vertical, siendo su sentido positivo hacia arriba. A la altura del muelle su valor es 0.

Limites:

- Altura de la viga:  $Y_{to} = 45\text{m}$
- Despeje mínimo sobre el borde el muelle "Sill Beam":  $Y_{sb} = 15\text{m}$
- Posición:  $[-20.0\text{m}; 40.0\text{m}]$ .

- Velocidad:  $\pm 1.5\text{m/s}$  (con carga)  $\pm 3.0\text{m/s}$  (sin carga).
- Aceleración  $\pm 1.0\text{m/s}$ .

La metodología a seguir es similar a la desarrollada para el sistema del carro. La ecuación correspondiente al motor de izaje es:

$$J_{mh}\dot{\omega}_{mh}(t) = T_{mh}(t) - T_{lh} - b_{mh}\omega_{mh}(t) \quad (18)$$

donde  $J_{mh}$  es el momento de inercia del eje del motor,  $\dot{\omega}_{mh}(t)$  es la aceleración angular del eje motor,  $\omega_{mh}(t)$  es la velocidad angular del mismo,  $T_{mh}(t)$  es el torque electromagnético,  $T_{lh}(t)$  es el torque de carga y  $b_m$  es el coeficiente de rozamiento viscoso del motor.

La ecuación del tambor está dada por:

$$J_d\dot{\omega}_d(t) = T_{qh}(t) - T_d(t) - b_d\omega_d(t) \quad (19)$$

donde  $J_d$  es el momento de inercia del tambor,  $\dot{\omega}_d(t)$  es la aceleración angular del tambor,  $\omega_d(t)$  es la velocidad angular del mismo,  $T_{qh}(t)$  es el torque motriz,  $T_d(t)$  es el torque de carga y  $b_m$  es el coeficiente de rozamiento viscoso del tambor.

Se puede relacionar a los dos sistemas anteriores a través de la relación de transmisión  $r_{th}$ :

$$r_{th} = \frac{\omega_{mh}(t)}{\omega_d(t)} = \frac{\dot{\omega}_{mh}(t)}{\dot{\omega}_d(t)} = \frac{T_{qh}(t)}{T_{lh}(t)} \quad (20)$$

Reemplazando (20) en (18) se refiere el eje del motor al eje del tambor y se despeja el torque  $T_{qh}$ :

$$J_{mh}r_{th}\dot{\omega}_d(t) = T_{mh}(t) - \frac{T_{qh}}{r_{th}} - b_{mh}r_{th}\omega_{mh}(t) \quad (21)$$

$$T_{qh} = T_{mh}(t)r_{th} - J_{mh}r_{th}^2\dot{\omega}_d(t) - b_{mh}r_{th}^2\omega_{mh}(t) \quad (21)$$

Reemplazando (21) en (19):

$$J_d\dot{\omega}_d(t) = r_{th}T_{mh} - J_{mh}r_{th}^2\dot{\omega}_d(t) - b_{mh}r_{th}^2\omega_{mh}(t) - T_d(t) - b_d\omega_d(t) \quad (22)$$

El torque resultante de el tambor es:

$$T_d(t) = r_{th}T_{mh} - J_{mh}r_{th}^2\dot{\omega}_d(t) - b_{mh}r_{th}^2\omega_{mh}(t) - J_d\dot{\omega}_d(t) - b_d\omega_d(t) \quad (23)$$

Los movimientos angulares del tambor están relacionados con la longitud sin enrollar del cable por:

$$\omega_d(t) = \frac{\dot{l}_h(t)}{R_d} \quad ; \quad \dot{\omega}_d(t) = \frac{\ddot{l}_h(t)}{R_d} \quad (24)$$

donde  $R_d$  es el radio del tambor,  $\dot{l}_h(t)$  es la velocidad a la que cambia la longitud del cable y  $\ddot{l}_h(t)$  es la aceleración a la que cambia la longitud del cable.

La fuerza tangencial en el tambor está dada por:

$$F_h(t) = \frac{-T_d(t)}{R_d} \quad (25)$$

Reemplazando (23) en (25)

$$F_h(t) = -\frac{r_{th}}{R_d} T_{mh} + \frac{J_{mh} r_{th}^2 + J_d}{R_d^2} \ddot{l}_h(t) + \frac{b_{mh} r_{th}^2 + b_d}{R_d^2} \dot{l}_h(t) \quad (26)$$

La ecuación correspondiente al accionamiento de izaje está dada por:

$$M_h \ddot{l}_h(t) = -F_h(t) - b_h \dot{l}_h(t) + F_w(t) \quad (27)$$

donde  $M_h$  es la masa del punto tangencial,  $b_h$  es el coeficiente de rozamiento viscoso entre el cable y el tambor,  $l_h(t)$  es la longitud del cable sin elongar y  $F_w(t)$  es la tensión del cable de izaje.

Reemplazando (26) en (27) se obtiene el modelo dinámico de izaje:

$$\left( \frac{J_{mh} r_{th}^2 + J_d}{R_d^2} + M_h \right) \ddot{l}_h(t) = \frac{r_{th}}{R_d} T_{mh} - \left( \frac{b_{mh} r_{th}^2 + b_d}{R_d^2} + b_h \right) \dot{l}_h(t) + F_w(t) \quad (28)$$

Al no considerar la masa del cable y el rozamiento que tiene con el tambor tanto  $M_h$  como  $b_h$  son igual a cero.

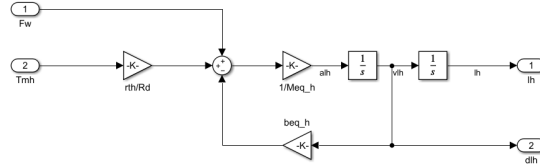


Figure 2: Representación en diagrama de bloques del sistema de izaje

## 2.3 Análisis de la Carga

El cable de acero posee masa despreciable, dispone en su extremo un gancho (denominado Spreader) cuya masa es igual a 15000kg. La carga total del sistema será la masa del gancho más la del contenedor siendo transportado.

La carga se encuentra apoyada al comienzo y al final de cada ciclo de trabajo. En esos momentos entran en cuenta los siguientes parámetros:

- Rigidez y fricción vertical (compresión):  $K_{cy} = 1,310^6 \frac{kN}{m}$ ;  $b_{cy} = 500 \frac{kN}{m}$
- Fricción horizontal (arrastre):  $b_{cx} = 1000 \text{ m/s}$



El cable de izaje, comprendido por 8 cables menor sección, puede ser representado como un único cable cuyos parámetros se especifican a continuación. Cabe destacar que el cable solo trabaja en tracción y en el caso de compresión se pandea.

- Rigidez a tracción:  $K_w = 1800 \frac{kN}{m}$
- Amortiguamiento (fricción interna de los hilos):  $b_w = 30 \frac{kN}{m \cdot s}$

$y_{c0}(x, t)$  es el perfil de elevaciones de obstáculos a lo largo del muelle, este evoluciona en el tiempo a medida que se desarrollen las maniobras de carga y descarga.

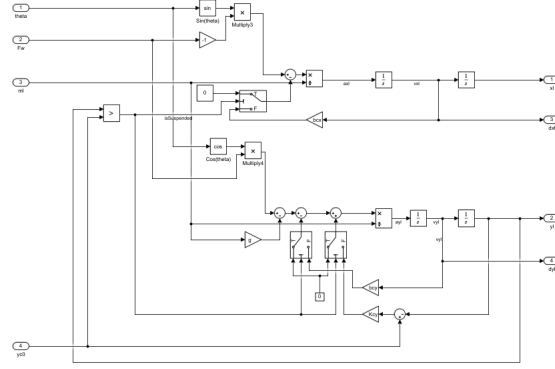


Figure 3: Representación en diagrama de bloques de la carga

### 2.3.1 Carga suspendida

Cuando la carga se encuentra suspendida se realiza un esfuerzo sobre el cable y este trabajara a tracción deformándose elásticamente aumentando su longitud en tracción  $l(T)$  respecto de su longitud en relajación  $l_h(t)$ , entonces:

$$l(t) \geq l_h(t) \quad (29)$$

$$l(t) = l_h(t) + \delta l \quad (30)$$

Considerando que las deformaciones del cable se mantienen dentro del límite elástico del material, se puede modelar la tensión del cable mediante la ley de Hooke.

$$F_w(t) = K_w(l(t) - l_h(t)) + b_w(\dot{l}(t) - \dot{l}_h(t)) \quad (31)$$

Luego las ecuaciones de movimiento en los ejes x e y, se modelan de la siguiente manera:

$$m_l \ddot{x}_l(t) = -F_w(t) \sin(\theta(t)) \quad (32)$$

$$m_l \ddot{y}_l(t) = F_w(t) \cos(\theta(t)) - m_l g \quad (33)$$

A su vez geométricamente las posiciones de la carga:

$$x_l(t) = x_t(t) + l(t) \sin(\theta(t)) \quad (34)$$

$$y_l(t) = y_{t0} - l(t) \cos(\theta(t)) \quad (35)$$

A su vez el ángulo de balanceo de la carga y la longitud del cable son dados por:

$$\theta(t) = \arctan\left(\frac{x_l(t) - x_t(t)}{y_{t0} - y_l(t)}\right) \quad (36)$$

$$l(t) = \sqrt{[x_l(t) - x_t(t)]^2 + [y_{t0} - y_l(t)]^2} \quad (37)$$

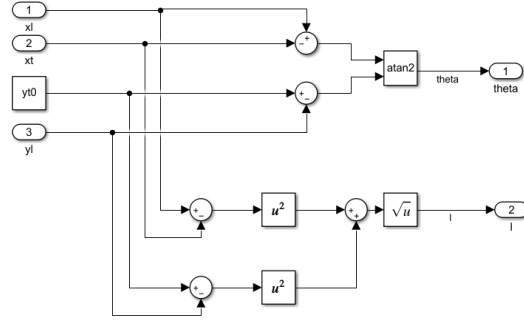


Figure 4: Cálculo de  $\theta$

Puesto que se considera la fricción interna del cable es necesario calcular su velocidad de deformación para obtener la componente viscosa de la tensión  $F_w$  para ello se deriva  $l(t)$  respecto del tiempo:

$$\dot{l}(t) = \frac{1}{\sqrt{(x_l(t) - x_t(t))^2 + (y_{t0} - y_l(t))^2}} [(x_l(t) - x_t(t))(\dot{x}_l(t) - \dot{x}_t(t)) - \dot{y}_l(t)(y_{t0} - y_l(t))] \quad (38)$$

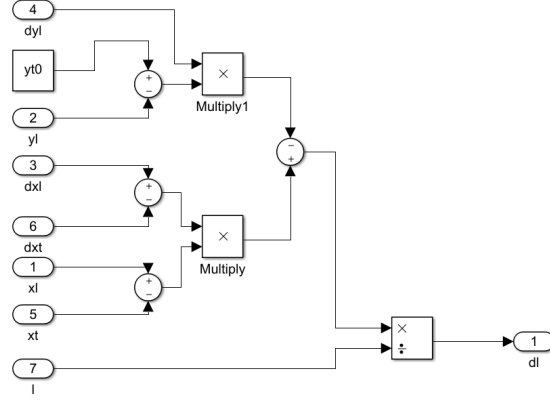


Figure 5: Derivación de l

### 2.3.2 Carga apoyada

Si la carga se encuentra apoyada se deben considerar los fenómenos de compresión a lo largo del eje y y la fricción de arrastre en el eje x. Cuando  $y_l(t) \leq y_{c0}(x, t)$  (carga apoyada), las ecuaciones de traslación pasan a ser.

$$m_l \ddot{x}_l(t) = -F_w(t) \sin(\theta(t)) - b_{cx} \dot{x}_l(t) \quad (39)$$

$$m_l \ddot{y}_l(t) = F_w(t) \cos(\theta(t)) - m_l g + K_{cy}(y_{c0}(x, t) - y_l(t)) - b_{cy} \dot{y}_l(t) \quad (40)$$

La transferencia de carga se producirá de forma progresiva hasta que  $F_w$  sea igual a cero, el cable pasará a estar “flojo”, por lo que al no haber contracción en el cable se pandeará y se cumplirá  $l(t) < l_h(t)$ , por lo que la tensión en el mismo pasa a ser:

$$F_w(t) = 0(l(t) - l_h(t)) + 0(\dot{l}(t) - \dot{l}_h(t)) \quad (41)$$

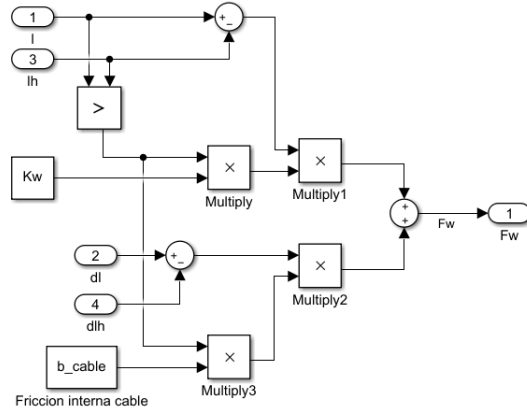


Figure 6: Representación en diagrama de bloques

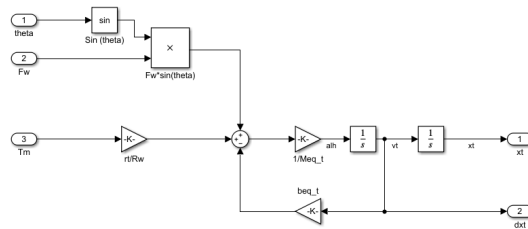


Figure 7: Representación en diagrama de bloques

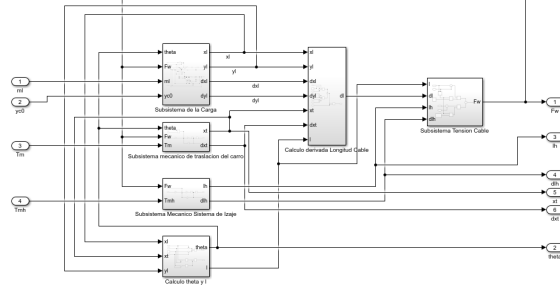


Figure 8: Representación en diagrama de bloques

### 3 Autómata híbrido de control y protección

El autómata contempla dos niveles de control: el primero es un control supervisor local de estados discretos activados por eventos (autómata secuencial), con estructura jerárquica y concurrencia, para operación suave y eficiente con coordinación y optimización de trayectorias, control de operación global del sistema y diagnóstico. El segundo nivel se trata de los controladores de movimiento de estados continuos en tiempo discretizado (control regulatorio de lazo cerrado), que reciben consignas de movimiento individuales del control supervisor para control directo de cada uno de los movimientos principales (izaje y traslación) y un control automático de balanceo de la carga, basados en accionamientos electromecánicos genéricos de 4 cuadrantes (Control PID de movimiento con modulador de torque). Además, se debe incorporar separadamente otro autómata con un nivel 0 para seguridad y/o protección ante fallas críticas o situaciones de riesgo.

#### 3.1 Nivel 2: Controladores de Movimiento

A continuación se detalla el planteo e implementación de los controladores PID de traslación del carro, izaje de carga y balanceo.

##### 3.1.1 Controlador de Traslación del Carro

El control de movimiento del carro se implementa mediante un controlador PID cuyos polos fueron asignados mediante el método ubicación de polos. Los polos del sistema se obtienen a través de su función de transferencia obteniendo el siguiente polinomio característico:

$$P_c = s(M_t s + b_t) \quad (42)$$

Siendo entonces los polos:

- $p_1 = -0,4576 \frac{rad}{s}$
- $p_2 = 0 \frac{rad}{s}$

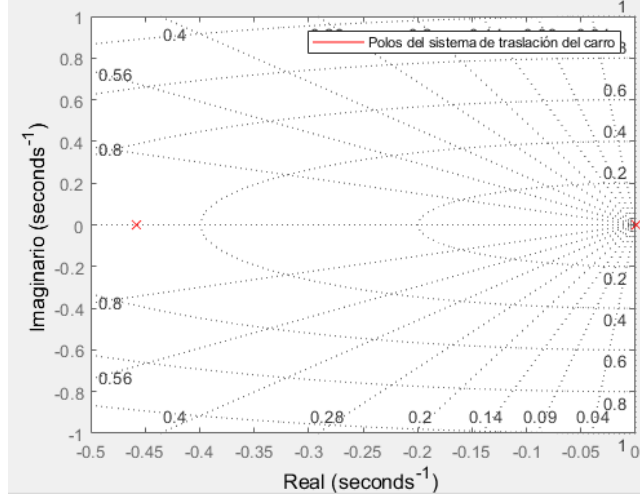


Figure 9: Polos del sistema de traslación del carro

A partir del polo más rápido del sistema ( $p_1$ ) se ubican los polos del controlador 5 veces a la izquierda del mismo, entonces  $w_{pos} = 5w_{p1}$ , además se añade un factor  $n$  que dictaminara el tipo de respuesta del controlador. En este caso se elige con sobreamortiguamiento con  $n = 4$ . De esta manera las ganancias del controlador quedan definidas como:

$$ba_t = (nw_{pos}M_t) \frac{R_W}{r_t} = 540000 \quad (43)$$

$$ksa_t = (nw_{pos}^2M_t) \frac{R_W}{r_t} = 1,2354e + 06 \quad (44)$$

$$ksia_t = (nw_{pos}^3M_t) \frac{R_W}{r_t} = 7,0661e + 05 \quad (45)$$

Para evitar los inconvenientes numéricos resultado de usar un bloque derivador se toma como entrada del controlador la derivada del error, es decir la diferencia entre velocidades deseadas y reales de traslación.

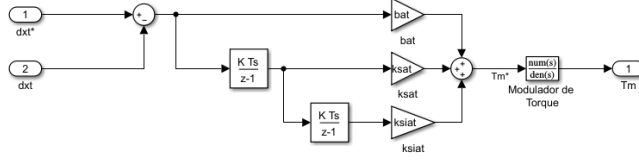


Figure 10: Nivel 2: Lazo de posición

### 3.1.2 Controlador de Izaje de Carga

El procedimiento para diseñar el controlador del sistema de izaje es análogo al anterior. Se obtiene el polinomio característico:

$$Pc = s(M_h s + b_h) \quad (46)$$

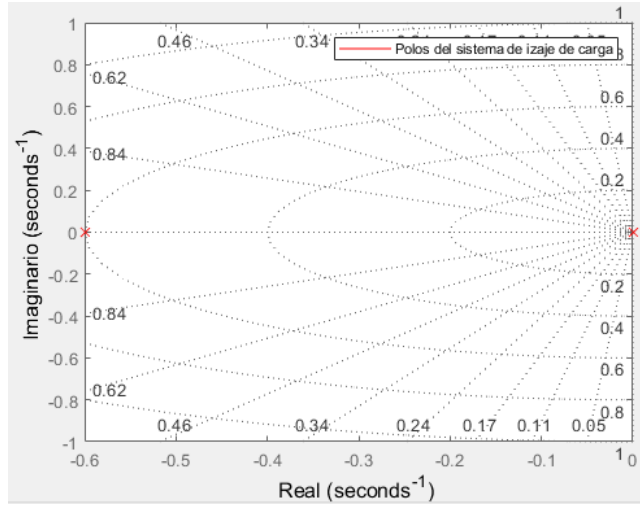


Figure 11: Polos del sistema de izaje de carga

Siendo entonces los polos del sistema  $p_1 = -0.5998 \frac{rad}{s}$  y  $p_2 = 0 \frac{rad}{s}$ . Los polos del controlador se ubican 10 veces más a la izquierda del polo más rápido del sistema ( $w_{pos} = 10w_{p1}$ ) y se configura con  $n = 3$  para obtener una respuesta críticamente amortiguada resultando en las siguientes ganancias:

$$ba_h = (nw_{pos} M_t) \frac{R_W}{r_t} = 864000 \quad (47)$$

$$ksa_h = (nw_{pos}^2 M_t) \frac{R_W}{r_t} = 5,1825e + 06 \quad (48)$$

$$ksia_h = (nw_{pos}^3 M_t) \frac{R_W}{r_t} = 1,0362e + 07 \quad (49)$$

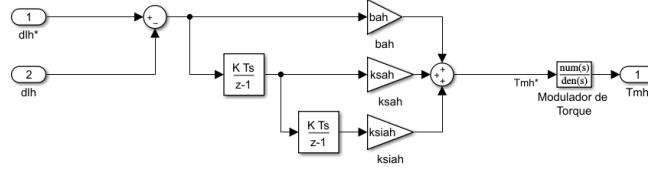


Figure 12: Nivel 2: Lazo de posición

### 3.1.3 Controlador de Balanceo de Carga

La carga al estar suspendida se comporta como un péndulo, realizando balanceos no deseados cuando se desarrollen movimientos horizontales de traslación. Para restringir esta oscilación a ángulos seguros se implementa un controlador adicional complementario al de traslación. Para ello se emplea un sensor instantáneo de ángulo presente en el sistema. El controlador debe satisfacer los siguientes requerimientos:

- Ángulo máximo durante aceleración/desaceleración:  $\pm 20^\circ$ .
- Ángulo máximo durante la trayectoria a velocidad "constante":  $\pm 5^\circ$ .
- Ángulo residual al completar movimiento y detener el carro:  $\pm 1^\circ$ .

Este controlador es del tipo PD, con un sistema adaptativo de acuerdo a la longitud del péndulo equivalente, la masa del mismo y el tipo de movimiento siendo desarrollado.

Se plantea el sistema como un péndulo y se obtiene su ecuación de movimiento mediante la formulación de Lagrange:

$$\begin{aligned} v_t &= \dot{x}_t; v_\theta = \dot{\theta}l \\ v_l^2 &= (\dot{x}_t + \dot{\theta}l \cos \theta)^2 + (\dot{\theta}l \sin \theta)^2 \\ v_l^2 &= \dot{x}_t^2 + 2\dot{x}_t\dot{\theta}l \cos \theta + \dot{\theta}^2 l^2 \\ v_l^2 &= \dot{x}_t^2 + 2\dot{x}_t\dot{\theta}l \cos \theta + \dot{\theta}^2 l^2 \end{aligned}$$

La energía cinética total del sistema es:

$$K = \frac{1}{2} M_t v_t^2 + \frac{1}{2} m_l v_l^2$$



$$K = \frac{1}{2}[M_t v_t^2 + m_l(\dot{x}_t^2 + 2\dot{x}_t\dot{\theta}l \cos \theta + \dot{\theta}^2 l^2)]$$

Y la energía potencial:

$$U = m_l g l (1 - \cos \theta)$$

Entonces el Lagrangiano queda:

$$L = K - U = \frac{1}{2}[M_t v_t^2 + m_l(\dot{x}_t^2 + 2\dot{x}_t\dot{\theta}l \cos \theta + \dot{\theta}^2 l^2)] - m_l g l (1 - \cos \theta)$$

Se toman la posición del carro y el ángulo desarrollado como coordenadas generalizadas. Se comienza con  $q = x_t$ :

$$\frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{x}_t} \right) - \frac{\partial \mathcal{L}}{\partial x_t} = F_t - b_t v_t$$

$$(M_t + m_l) \ddot{x}_t + m_l l \ddot{\theta} \cos \theta - m_l l \sin \theta \dot{\theta}^2 = F_t - b_t v_t$$

y para la coordenada  $\theta$  la ecuación es:

$$\frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{\theta}} \right) - \frac{\partial \mathcal{L}}{\partial \theta} = 0$$

$$\ddot{x}_t \cos \theta + l \ddot{\theta} + g \sin \theta = 0$$

Si se considera que el ángulo de balanceo es pequeño, se pueden realizar las siguientes aproximaciones:

$$(M_t + m_l) \ddot{x}_t = m_l l \ddot{\theta} = F_t - b_t v_t \quad (50)$$

$$\ddot{x}_t + l \ddot{\theta} + g \theta = 0 \quad (51)$$

Despejando la aceleración del carro de 51 se obtiene:

$$\dot{v}_t = \ddot{x}_t = \frac{F_t - b_t v_t}{m_t}$$

Reemplazando en 51 y despejando  $F_t$  resulta:

$$F_t = -m_t (l \ddot{\theta} + g \theta) + b_t v_t \quad (52)$$

Despejando  $F_t$  de 50, igualando con 52 se obtiene:

$$(M_t + m_l) \ddot{x}_t = -(M_t + m_l) l \ddot{\theta} - M_t g \theta \quad (53)$$

Aplicando la transformada de Laplace a la ecuación anterior resulta:

$$-(M_t + m_l) s V_t(s) = [(M_t + m_l) l s - M_t g] \Theta(s)$$

De donde se puede obtener la función de transferencia para una entrada  $V_t(s)$  y salida  $\Theta(s)$ :

$$G_p(s) = \frac{\Theta(s)}{V_t(s)} = \frac{-(M_t + m_l)s}{(M_t + m_l)ls^2 - M_tg} \quad (54)$$

De la función de transferencia se puede obtener la frecuencia natural del péndulo:

$$\omega_n = \sqrt{\frac{M_tg}{(M_t + m_l)l}}$$

El efecto producido por el controlador de balanceo puede ser considerado como una perturbación en la señal del controlador de traslación del carro. La función de transferencia del sistema respecto a la entrada de perturbación es entonces:

$$G_{yd} = \frac{Y(s)}{D(s)} = \frac{G_p}{1 + G_cG_p} \quad (55)$$

Donde  $G_p$  es la función de transferencia en 54 y  $G_c$  es la función de transferencia del controlador de balanceo (tipo PD) expresado a continuación:

$$G_c(s) = K_p + K_d s$$

Reemplazando en 55 se obtiene:

$$\frac{\Theta(s)}{t(s)} = \frac{G_p(s)}{1 + G_c(s)G_p(s)} = \frac{\frac{-(M_t + m_l)s}{(M_t + m_l)ls^2 + M_tg}}{1 + \frac{-(M_t + m_l)s}{(M_t + m_l)ls^2 + M_tg}} (K_p + K_d s) \quad (56)$$

De donde se puede despejar el siguiente denominador:

$$Denominador(s) = s^2 - \frac{K_p}{l - K_d}s = \frac{M_tg}{(M_t + m_l)(l - K_d)} \quad (57)$$

Para asignar los polos se compara dicho denominador con el siguiente polinomio característico de un sistema críticamente amortiguado ( $= 1$ )

$$(s - \omega_n) = s^2 - 2\omega_n s + \omega_n^2 \quad (58)$$

Comparando se obtienen las ganancias del controlador.

$$\begin{aligned} \omega_n^2 &= \frac{M_tg}{(M_t + m_l)(l - K_d)} \\ K_d &= l - \frac{M_tg}{(M_t + m_l)\omega_n^2} \end{aligned} \quad (59)$$

$$\begin{aligned} -2\omega_n &= -\frac{K_p}{l - K_d} \\ K_p &= 2\omega_n(l - K_d) \end{aligned} \quad (60)$$

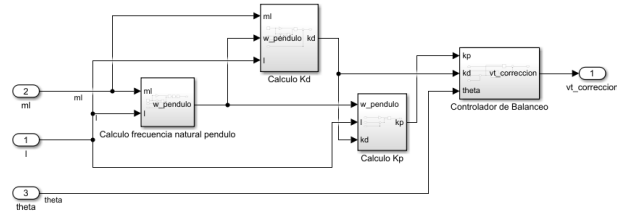


Figure 13: Nivel 2: Lazo de posición

La condición de cargado o vacío del Spreader, y el peso total suspendido, deben tenerse en cuenta para definir la velocidad máxima de izaje (operación a potencia constante en función de la carga, para aprovechar la potencia instalada en motores y reducir tiempo de ciclo con cargas menores a la nominal. Este tipo de comportamiento se ve ejemplificado en la siguiente gráfica.

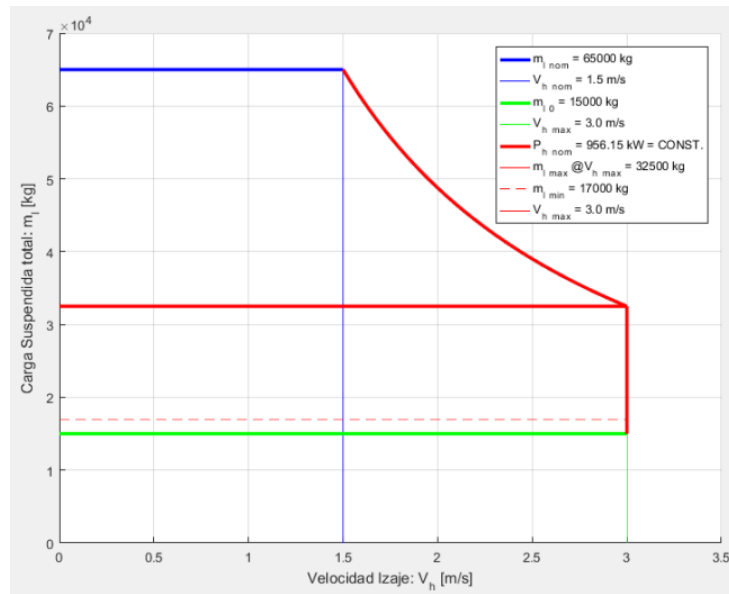


Figure 14: Característica de potencia constante - carga suspendida vs. velocidad izaje

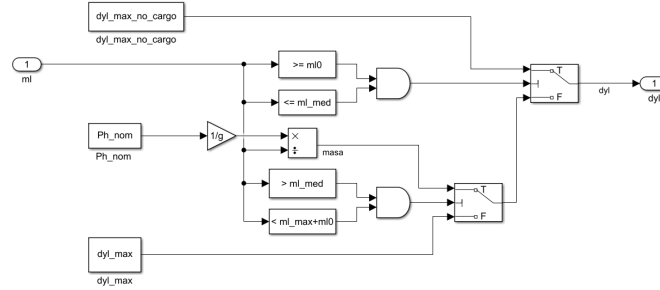


Figure 15: Modelado de la característica de potencia constante

A continuación se puede ver el modelo del sistema físico junto con el nivel 2 completo, modelado en simulink.

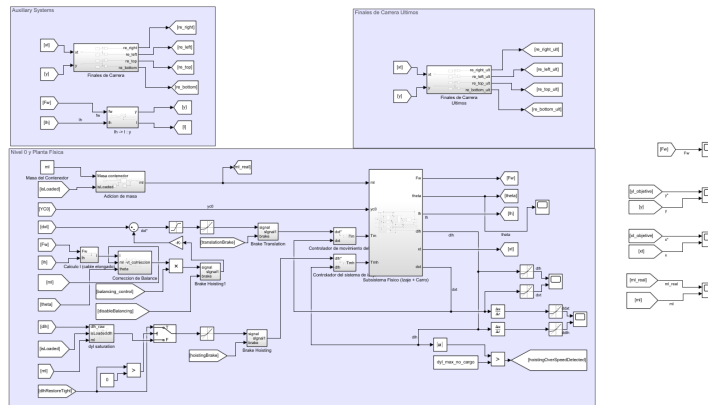


Figure 16: Nivel 2 y Planta Física

### 3.1.4 Lazo de realimentación de posición

Para aumentar la precisión del sistema se incorporó un lazo de realimentación de posición del tipo Proporcional con una ganancia baja para actuar en armonía con el controlador de balanceo. De esta manera se alcanza la posición de destino con ángulos y velocidades de oscilación en la carga cercanas a 0. Minimizando la oscilación final del sistema que pasa a actuar como críticamente amortiguado. Esto minimiza el malestar ocasionado por los ciclos de aceleración y desaceleración al operario presente en la cabina de control.

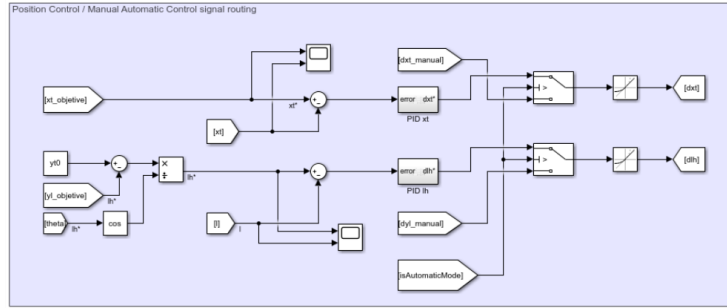


Figure 17: Nivel 2: Lazo de posición

Como mejora para prevenir picos repetitivos de aceleración/desaceleración se añadieron filtros que suavizaron la señal previniendo el sacudidas o jerk y la fatiga innecesaria de materiales y personal a bordo de la cabina de mando.

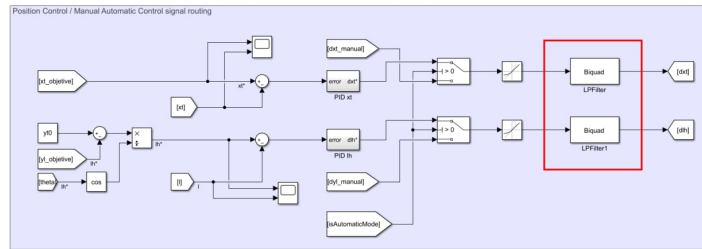


Figure 18: Filtros

### 3.1.5 Nivel 2 en Codesys

Para realizar el nivel 2 en Codesys se lo escribió en texto estructurado (ST) ya que presentaba mayor facilidad para realizar ciertos cálculos, como por ejemplo integración a través del método de Tusting. De este modo se escribieron los controladores PID correspondientes para cada caso en este lenguaje. En las siguientes figuras se observan los function blocks de los controladores PID de traslación e izaje, también se observa el cálculo de dlh en este nivel y el control adaptativo del controlador de balanceo.

1	FUNCTION_BLOCK Level2_TranslationPID
2	VAR_INPUT
3	err : REAL;
4	END_VAR
5	VAR_OUTPUT
6	Tm : REAL;
7	END_VAR
8	VAR
9	ksia : REAL;
10	ksa : REAL;
11	ba : REAL;
12	prevErr : REAL;
13	prevP : REAL;
14	h : REAL := 0.002;
15	p : REAL;
16	i : REAL;
17	d : REAL;
18	END_VAR

1	ksia := 54000;
2	ksa := 1.2354 * 1000000;
3	ba := 7.0661 * 100000;
4	p := (err + prevErr) * h / 2;
5	i := (p + prevP) * h / 2;
6	d := err;
7	Tm := p + i + d;
8	
9	prevErr := err;
10	prevP := p;

Figure 19: Nivel 2: Controlador PID de traslación

1	FUNCTION_BLOCK Level2_HoistingPID
2	VAR_INPUT
3	err : REAL;
4	END_VAR
5	VAR_OUTPUT
6	Tmh : REAL;
7	END_VAR
8	VAR
9	ksia : REAL;
10	ksa : REAL;
11	ba : REAL;
12	prevErr : REAL;
13	prevP : REAL;
14	h : REAL := 0.002;
15	p : REAL;
16	i : REAL;
17	d : REAL;
18	END_VAR

---

1	ksia := 86400;
2	ksa := 5.1825 * 1000000;
3	ba := 1.0362 * 10000000;
4	p := (err + prevErr) * h / 2;
5	i := (p + prevP) * h / 2;
6	d := err;
7	Tmh := p + i + d;
8	
9	prevErr := err;
10	prevP := p;
11	

Figure 20: Nivel 2: Controlador PID de izaje

```

39 //Calculo de dlh*
40 IF GVL.hoistingBrake > 0 THEN;
41     error_dlh := 0;
42 ELSE
43     IF GVL.dlhRestoreTight > 0 THEN;
44         error_dlh:= GVL.dlhRestoreTight;
45     ELSE
46         IF GVL.ml >= GVL.ml0 AND GVL.ml <= GVL.ml_med THEN;
47             dyl:= GVL.dyl_max_no_cargo;
48         ELSIF GVL.ml > GVL.ml_med AND GVL.ml<(GVL.ml_max+GVL.ml0) THEN;
49             dyl := GVL.Ph_nom/GVL.g/GVL.ml;
50         ELSE
51             dyl := GVL.dyl_max;
52         END_IF
53     IF GVL.dlh>dyl THEN
54         error_dlh := dyl;
55     ELSIF GVL.dlh<-dyl THEN
56         error_dlh := -dyl;
57     ELSE
58         error_dlh := GVL.dlh;
59     END_IF
60 END_IF
61 END_IF
62 rate_lim(u:=error_dlh,Rising:=GVL.ddyl_max,Falling:=-GVL.ddyl_max);
63 error_dlh := rate_lim.out;

```

Figure 21: Nivel 2: Calculo de dlh

```

20 //Control de balanceo
21 Dtheta := (GVL.theta - theta_prev) / GVL.dt2;
22 theta_prev := GVL.theta; //Se guarda ese valor para el próximo paso
23 IF GVL.disableBalancing = 0 THEN
24     IF GVL.balancingControl THEN //Activado o no activado el control de balanceo
25         w_pendolo := SQRT(ABS((GVL.Meq_t*GVL.g)/((GVL.Meq_t+GVL.ml)*GVL.l)));
26         Kdb := GVL.l - GVL.Meq_t * GVL.g / ((GVL.Meq_t + GVL.ml) * w_pendolo * w_pendolo); //Ecuación de la ganancia
27         Kpb := 2.0 * w_pendolo * (GVL.l - Kdb); //Ecuación de la ganancia Kpb
28         theta_derivada := 0 - Dtheta; //theta deseado es 0
29         vt_correccion := GVL.theta * Kdb + theta_derivada * Kpb;
30         vt_correccion := vt_correccion * 0.25; //Se lo multiplica por una ganancia a la salida
31     ELSE
32         vt_correccion := 0;
33     END_IF
34 ELSE
35     vt_correccion :=0;
36 END_IF
37

```

Figure 22: Nivel 2: Controlador de balanceo adaptativo



### 3.2 Nivel 1: Control Supervisor Global

Se desarrolló la máquina de estados del control supervisor. Para agilizar el diseño, simulación y verificación del mismo se procedió a prototipar en Simulink y posteriormente realizar una implementación de producción en Codesys.

Este autómata se compone de varios estados concurrentes: supervisión de control (input del usuario, finales de carrera, verificación de sujeción del contenedor, activación de control de balanceo, prevención de pandeo del cable y emisiones de señal de pulso de vida), así como los estados de operación manual y automático.

El input del usuario recibe información del operador y configura el autómata de forma acorde. Los finales de carrera en este nivel señalan los finales razonables de traslación hasta los que se puede llegar en la normal operación del sistema. Estos son anteriores a los finales de carrera últimos monitoreados en el nivel 0. La verificación de sujeción de contenedores emplea los 4 sensores de contacto presentes en el Spreader para verificar el correcto contacto con el contenedor. El control de balanceo de la carga permite al operario anular dicho controlador durante las operaciones manuales. El sistema de prevención de pandeo del cable monitorea las tensiones en el cable y la longitud de cable desenrollado evitando desenrollar cable demás para prevenir el afloje de las poleas del sistema de sujeción e izaje. El sistema de emisión de señal y pulsos de vida informan al nivel 0 del correcto funcionamiento del sistema.

Como se cuenta con un sensor de fuerza en la sujeción del Spreader es posible medir la tensión a la que se encuentra sometido el cable. De esta manera se puede determinar la masa de la carga como  $ml = \frac{F_w}{9,81}$  (figura 23) cuando el sistema se encuentra estable (figura 24). El proceso de toma de carga, consiste en adicionar una masa aleatoria comprendida entre los límites de masa del contenedor en el momento que la traba del spreader es cerrada y bloqueada. Luego, se limita la velocidad de izaje de forma tal de comenzar una elevación suave y lenta de la carga, la cual no se despegará del suelo hasta que la tensión de la cuerda no se iguale el peso de la carga. Analizándolo desde la forma que va tomando la fuerza  $F_w$ , en el momento que la misma deje de crecer y se vuelva estable en un valor constante independiente de la posición vertical, la carga habrá despegado. De esta forma, analizando la señal de  $F_w$ , es posible determinar en que momento la misma deja de crecer y obtener su valor numérico para luego calcular la masa que se usará en el cálculo de los perfiles de velocidad.



Figure 23: Nivel 1: Sensado de masa

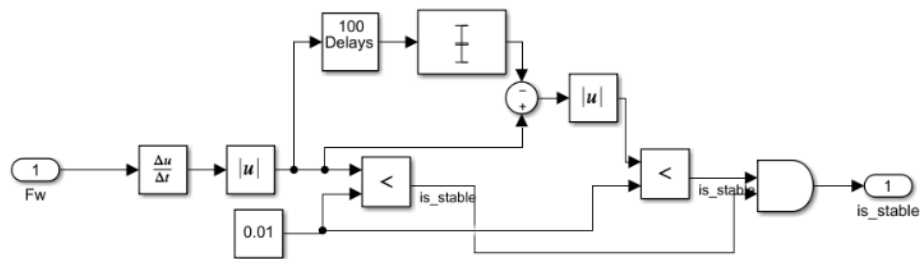


Figure 24: Nivel 1: Detección de masa

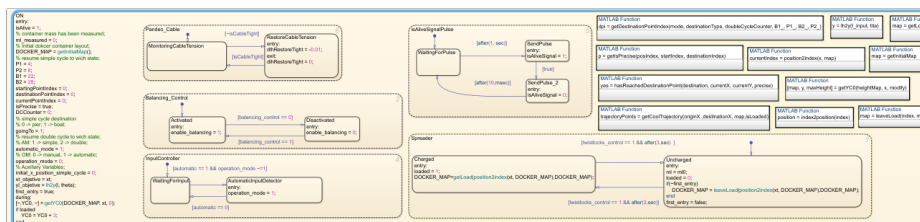


Figure 25: Nivel 1: Control Supervisor

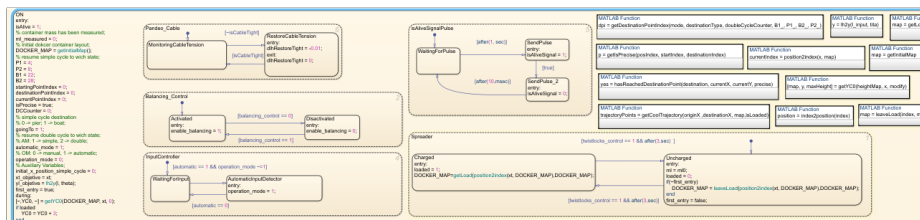


Figure 26: Nivel 1: Control Supervisor

El modo de operación por otro lado consisten en los estados manual y automático. Se intercambia entre ellos de acuerdo a las consignas de parte del usuario o para prevenir incidentes detectados por este nivel de control supervisor. El modo manual se emplea para las maniobras de aproximación e inicios de trayectorias. Estas maniobras pueden ser con o sin carga. Una vez completada la maniobra se puede pasar al modo automático de operación. Por otro lado, el Modo Automático está a cargo de las maniobras principales de izaje y traslación coordinadas entre muelle y barco a la máxima velocidad posible, con el objetivo de reducir tiempos de operación, aumentar la productividad y disminuir la fatiga del operador. Estas trayectorias generadas automáticamente

pueden reclasificarse en Ciclo Simple y Ciclo Doble. El Ciclo Simple (de ahora en adelante CS) se activa ante una consigna del operador una vez terminada la maniobra inicial de izaje. El CS calcula una trayectoria segura y eficiente para trasladar la carga a la posición de destino, dejando la maniobra final de posicionado al operario. El Ciclo Doble (de ahora en adelante CD) se compone de 3 ciclos simples que permiten realizar en un mismo viaje de ida y vuelta, al momento de cargar o descargar el contenedor se hace de forma manual. De esta manera se logra compartir bastante lógica entre ambos modos de operación simplificando la implementación del autómata y minimizando la superficie de fallas.

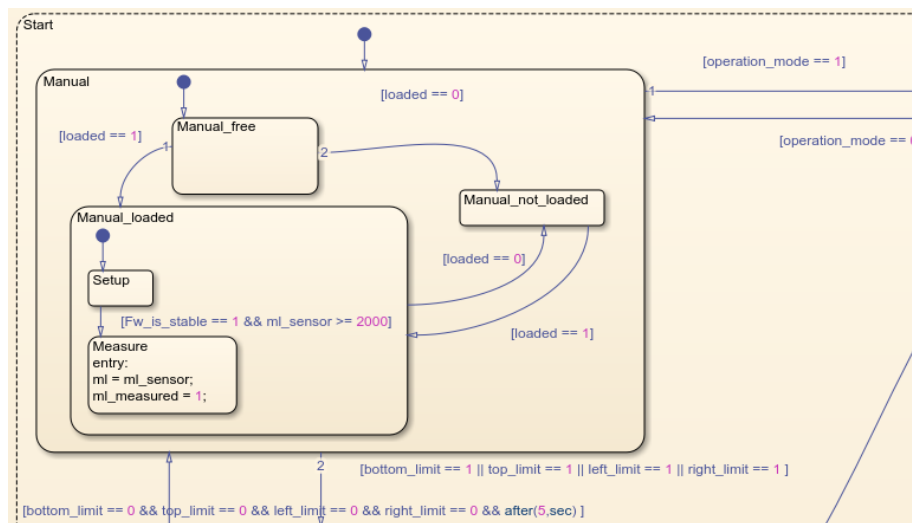


Figure 27: Nivel 1: Manual

Además el Modo Automático implementa un algoritmo de generación de trayectorias seguras y eficientes. Estas consisten en una serie de puntos por los cuales debe pasar el centro del spreader. La trayectoria se encuentra discretizada en puntos cada 2.5m (el ancho de cada contenedor) o filas de contenedores. Para cada uno de estos X discretos se asigna un valor de altura correspondiente. Las alturas se calculan de la siguiente manera:

- Se encuentra el punto más alto entre el punto de origen y el punto de destino.
- Se inicializan los valores
- Se itera desde el origen hasta la posición del punto más alto donde en cada punto de la trayectoria se elige la altura del punto sobre el cual se está iterando y si es más alta que la del punto anterior de la trayectoria, en

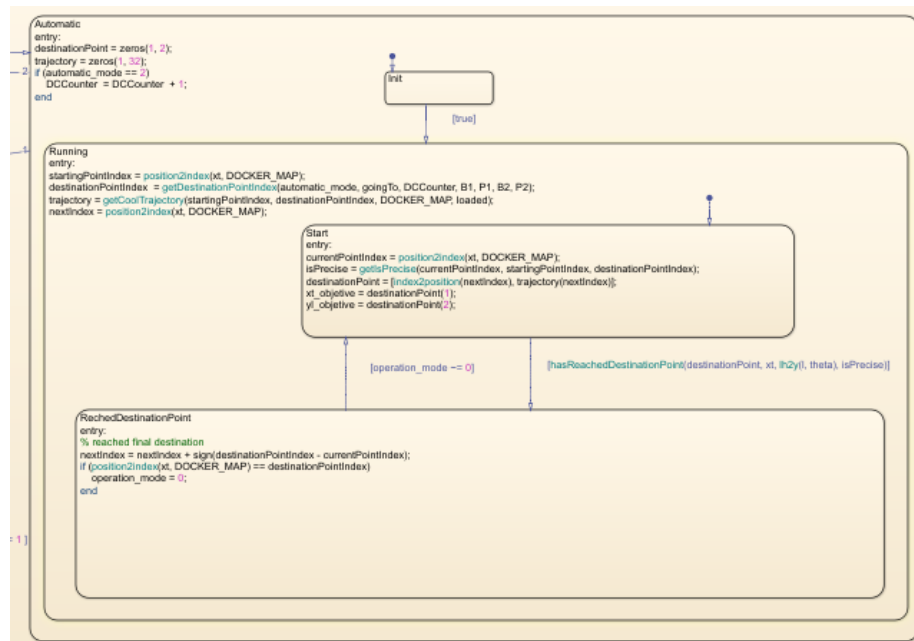


Figure 28: Nivel 1: Automático

caso contrario la altura elegida corresponde a la interpolación lineal entre el punto anterior y el más alto de la trayectoria.

- Se itera desde el destino hasta la posición del punto más alto calculando la trayectoria de la misma manera que en el punto anterior.

Esta trayectoria se calcula cuando se ingresa al modo Automático, y se ha implementado de tal manera que no se emplea memoria dinámica (heapless), siguiendo las recomendaciones standard de programación de sistemas embebidos. A su vez se ha optimizado el código reutilizando estructuras de datos eficientes para minimizar el consumo de memoria estática (stack) permitiendo ejecutar el autómata en procesadores más simples, confiables y económicos.

Posteriormente el sistema fue re implementado en la plataforma Codesys empleando los lenguajes gráficos SFC (Sequential Function Chart) y procedurales ST (Structured Text) que permiten exportar el autómata en formatos ejecutables por controladores industriales tipo PLC de acuerdo a la norma IEC 61131-3. De esta manera se emplea Simulink como Model in the Loop y Codesys como Software in the Loop. La comunicación entre ambos programas se logra mediante el protocolo de comunicación industrial OPC UA (Unified Architecture).

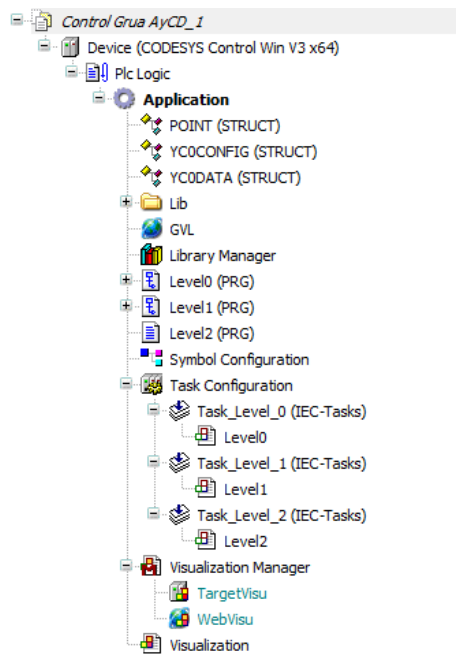


Figure 29: Árbol de proyecto general

```

1 FUNCTION_BLOCK checkFW
2   VAR_INPUT
3     Fw: LREAL;
4   END_VAR
5   VAR_OUTPUT
6     isTight: BOOL;
7     isStable: BOOL;
8     ml: LREAL;
9   END_VAR
10  VAR
11    previousFw: LREAL;
12    dFw: LREAL;
13    deltaT: LREAL := 0.002;
14    dFwHistory: ARRAY [0..99] OF LREAL := [00(0)];
15    i: DINT;
16  END_VAR
17
18  IF (Fw >= 0) THEN
19    isTight := TRUE;
20  END_IF
21
22  ml := Fw / 5.016;
23
24  dFw := ABS((Fw - previousFw) / deltaT);
25
26  dFwAccumulate := 0;
27  // update Fw History;
28  FOR i := LOWER_BOUND(dFwHistory, 1) TO UPPER_BOUND(dFwHistory, 1) - 1 DO:
29    dFwHistory[i] := dFwHistory[i + 1];
30    dFwAccumulate := dFwAccumulate + dFwHistory[i];
31  END_FOR
32  dFwHistory[99] := dFw;
33  dFwAccumulate := dFwAccumulate + dFw;
34  dFwMean := dFwAccumulate / 100;
35
36  IF (ABS(dFw - dFwMean) < tolerance AND dFw < tolerance) THEN;
37    isStable := TRUE;
38  END_IF

```

Figure 30: Nivel 1: Ejemplo de un Function Block

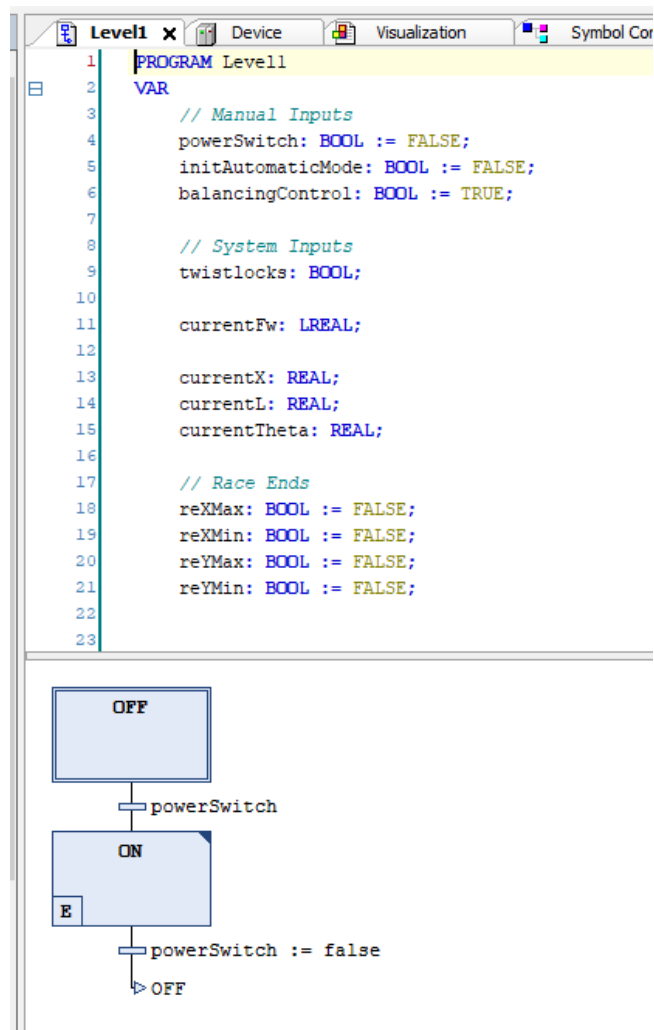


Figure 31: Nivel 1: On/Off

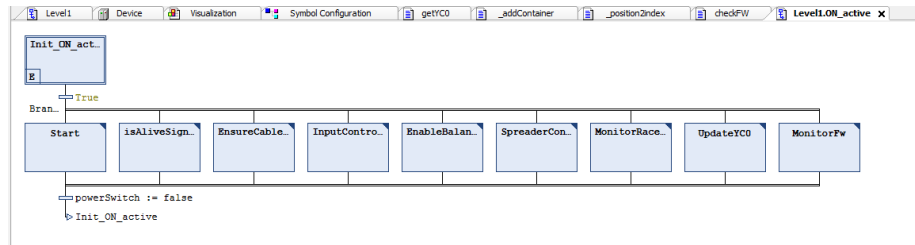


Figure 32: Nivel 1: On State

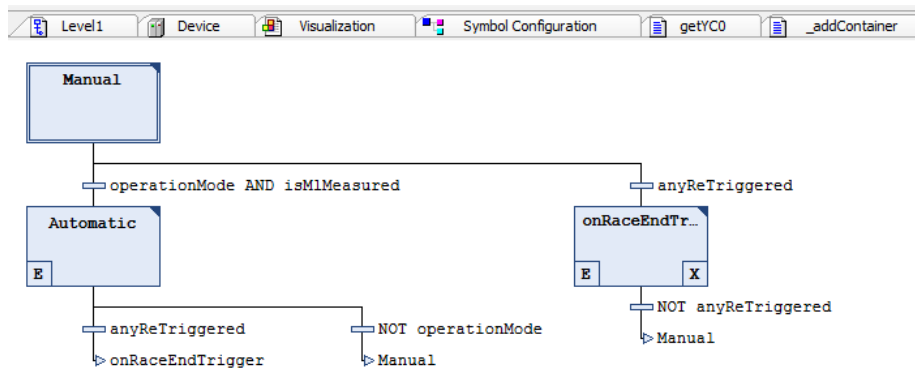


Figure 33: Nivel 1: Start

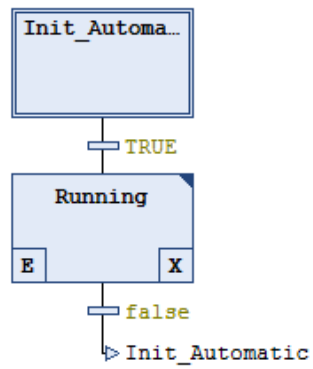


Figure 34: Nivel 1: Modo Automático

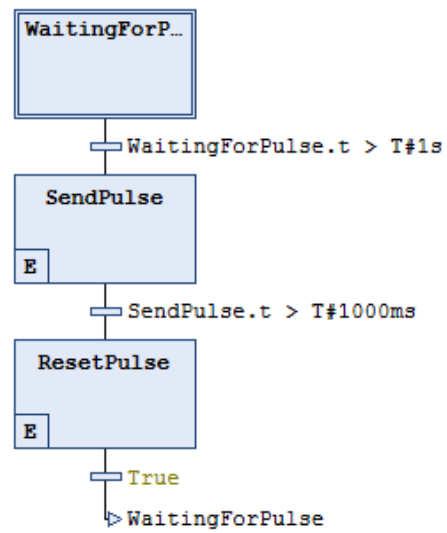


Figure 35: Nivel 1: ON - is Alive Pulse



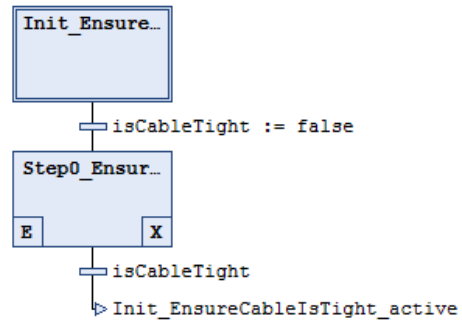


Figure 36: Nivel 1: ON - Ensure Cable Is Tight

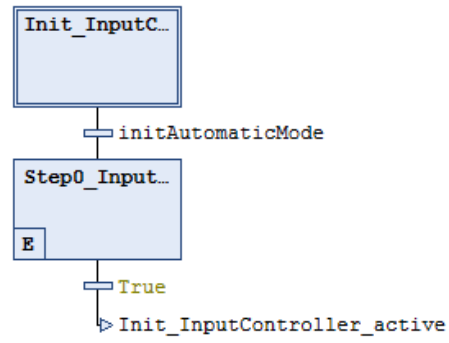


Figure 37: Nivel 1: ON - Input Controller active

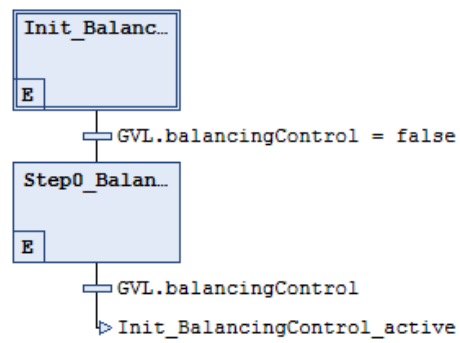


Figure 38: Nivel 1: ON - Enable Balancing Control

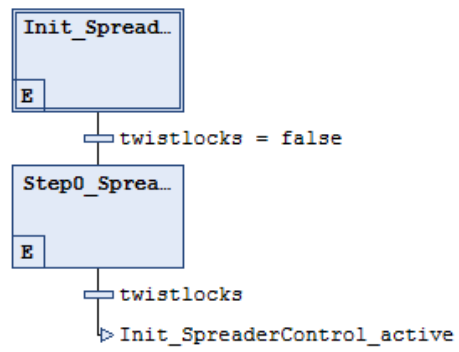


Figure 39: Nivel 1: ON - Spreader Control

### 3.3 Nivel 0: Función de Seguridad o Protección

Se realizó el nivel cero en otro autómata separado del nivel 1. Este autómata tiene como función el control ante fallas. Se observa en la figura 40 que tenemos un estado que al energizar el sistema pasa al macro estado "ON" donde se inicializan algunas variables.

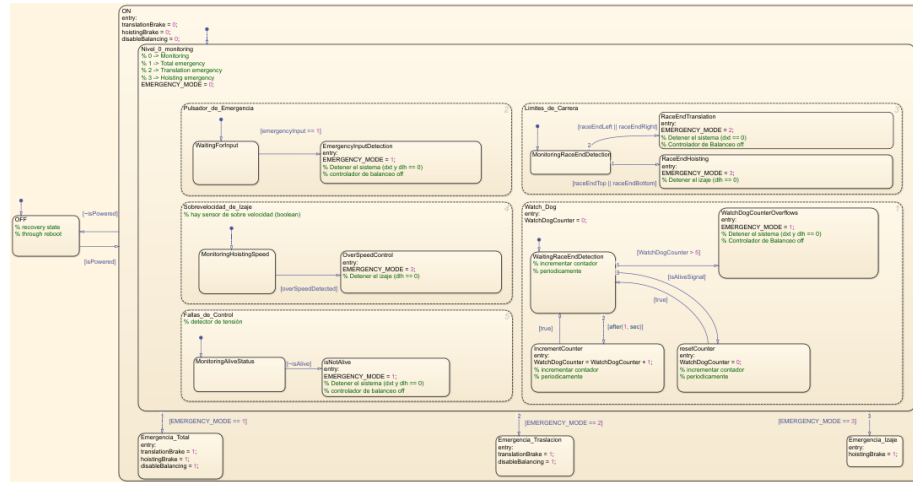


Figure 40: Nivel 0

Luego pasa a un estado donde se monitorizan las posibles fallas, en el se encuentran 5 estados concurrentes, según cual sea el que se active pasará a determinado estado de emergencia ya que este autómata está jerarquizado, con selectividad según el origen y criticidad de la falla. Entre los estados concurrentes (figura 41) tenemos:

**Pulsador de emergencia:** en caso de que el operador lo pulse entra en estado de emergencia total aplicando los frenos deteniendo tanto el izaje como el carro y deshabilitando el control de balanceo del carro.

**Sobrevelocidad de izaje:** En caso de que se supere la máxima velocidad de izaje el autómata aplicará los frenos en el izaje únicamente, ya que luego del producirse la falla debe ser revisado por personal de mantenimiento y para ello debería moverse el carro y no debería estar frenado.

**Fallas de control:** si no se recibe una señal que siempre debe tener una tensión constante en un valor determinado también se detiene el sistema completo.

**Límites de carrera:** la planta cuenta con ocho fines de carrera, de los cuales cuatro son del nivel 1 y los otros cuatro de este nivel se usan en este estado luego de los cuatro primeros, donde si se acciona el del límite derecho o izquierdo del carro entrará en emergencia de carro solamente y apagará el control de balanceo del carro, mientras que si se acciona el inferior o superior del izaje entrará en estado de emergencia del izaje únicamente.

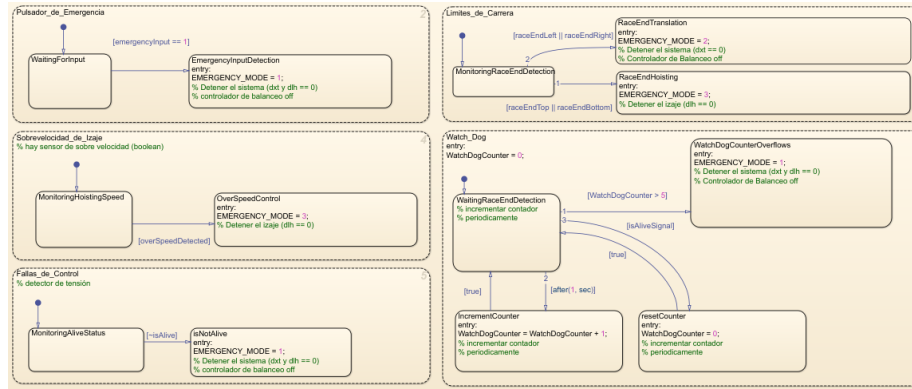


Figure 41: Nivel 0 detectores de emergencia

Watchdog: el autómata de nivel 1 le enviará pulsos cada 1s al autómata de nivel 0, mientras los reciba se incrementará un contador del watchdog y cuando supere los 5 pulsos pasará a estado de emergencia total. El contador se reiniciará mientras se tenga una tensión de vida evitando que se acumule el contador.

Luego de su implementación en simulink se replicó su funcionamiento en Codesys. Se comenzó por lo estado ON y OFF, que al energizarse se produce el flaqueo como se observa en la figura 42.

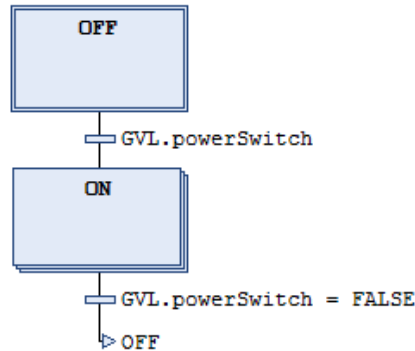


Figure 42: Nivel 0

Dentro del macro estado ON se tiene un macro estado que monitorea los casos de emergencia y en la parte inferior de la figura 43 se observan los posibles estados de emergencia.

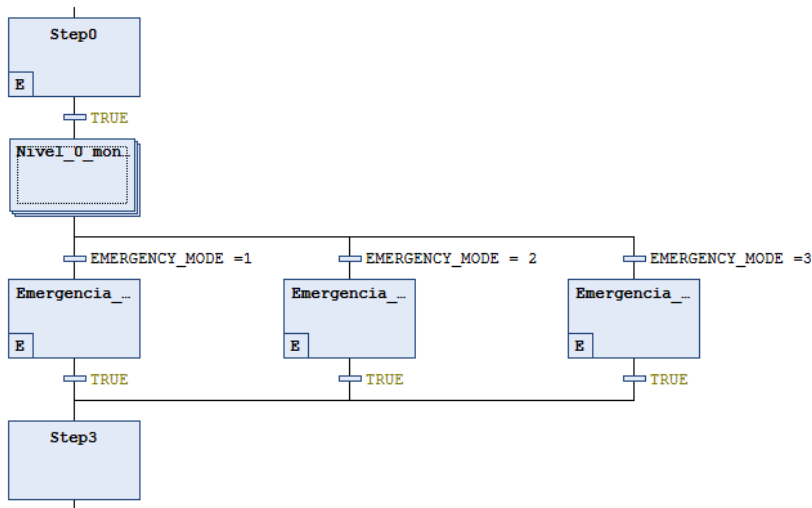


Figure 43: Nivel 0: ON

Dentro del estado Nivel 0 Monitoring se tienen macro estados concurrentes de las posibles fallas (44), que según cual sea ocurra será el estado de emergencia a activar.

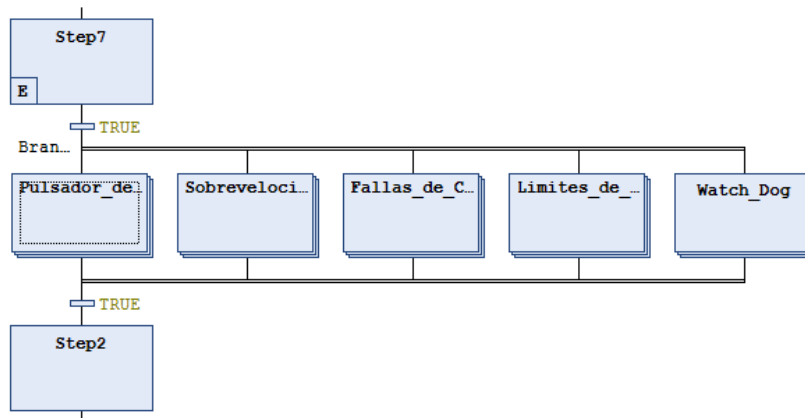


Figure 44: Nivel 0: Monitoring

## 4 HMI

En la figura 45 se observa la cabina del operario de la grúa. En esta se encuentran la interfaz hombre máquina (HMI) para permitir el control de la grúa.

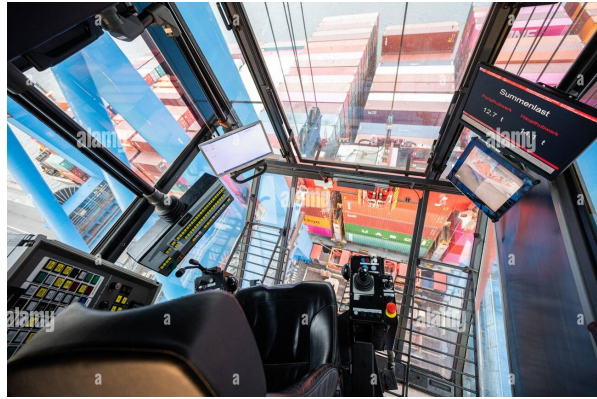


Figure 45: Operario de grúa

Por lo que se diseñó una interfaz hombre-máquina (HMI) en Codesys, que se observa en la figura 46, donde el operario puede elegir si va a realizar un ciclo doble o simple, el encendido, el paro de emergencia, la selección del lugar donde desea ir (muelle o barco), la selección de posibles posiciones en el barco o muelle mediante sliders, un display para saber el ángulo de balanceo, y las posiciones de en los ejes x e y.

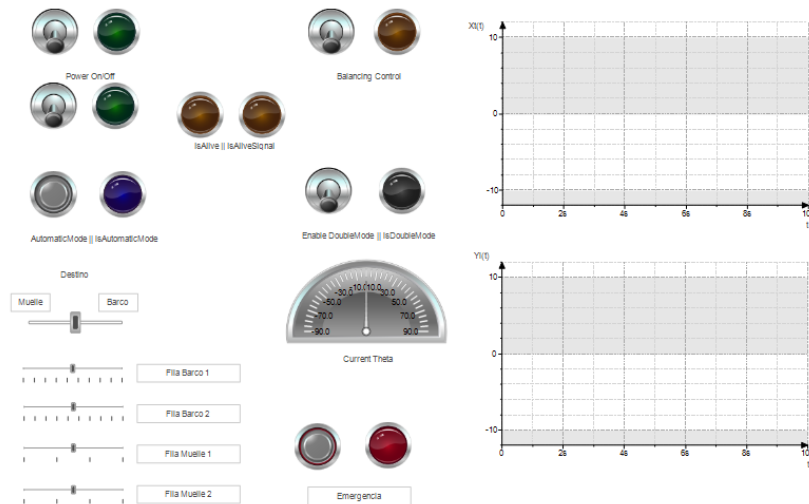


Figure 46: HMI

Para simular el control manual se hizo uso de un joystick de Playstation (figura 47) el cual fue configurado en simulink (figura 48) para usar las palancas de análogos para la traslación vertical y horizontal, y los botones para el inicio del modo automático y el control de los twistlocks.



Figure 47: Joystick

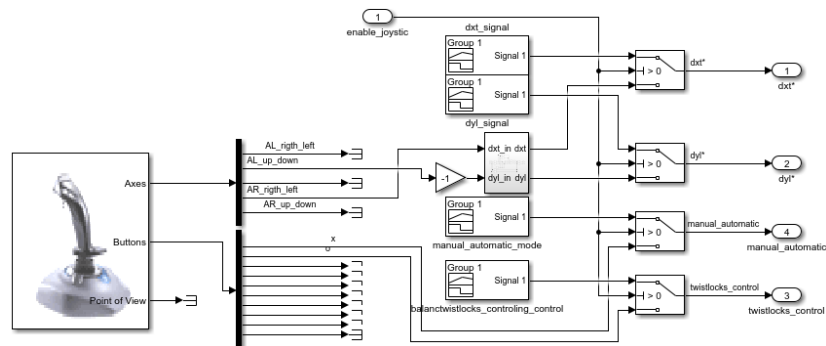


Figure 48: Manual input

## 5 Resultados

Se realizó la simulación de un ciclo simple para llevar una carga del muelle a el barco y volver al muelle. En la imagen 49 se tiene el estado inicial antes de comenzar el modo simple, donde los contenedores están representados por el perfil del color rojo en el muelle y en el barco.

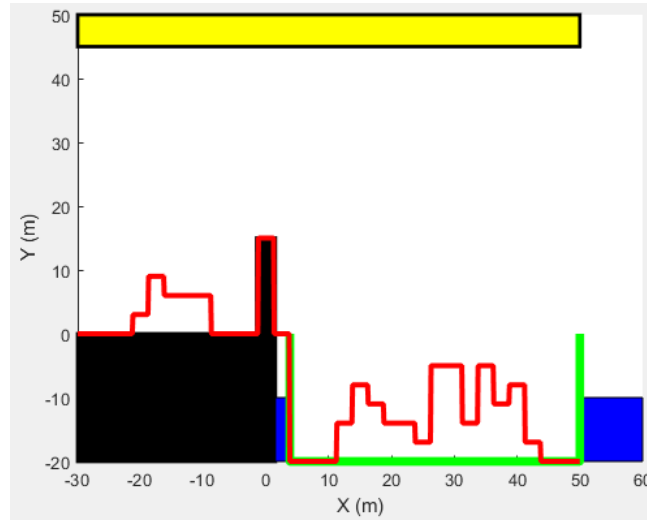


Figure 49: Estado inicial antes de comenzar el modo simple

Una vez realizado el modo simple se observa en la imagen 50 como se quitó un contenedor de lado del muelle, también se observa que una vez se tiene la carga se dibuja la línea de la parte de abajo del contenedor, de este modo se tiene un área de seguridad para disminuir el riesgo de impacto, como en este caso se tuvo en cuenta al pasar por encima de la sill beam. Una vez se traslada al barco luego se desciende manualmente a una altura prudencial hasta dejar el contenedor en su lugar de destino, actualizando el perfil rojo en el barco, luego se movió manualmente hacia arriba y se activa el modo automático para terminar el ciclo simple volviendo al muelle a una posición cercana a la del comienzo, se observa que al estar descargado en la vuelta este no tiene la línea de color naranja que representa la parte inferior del contenedor.



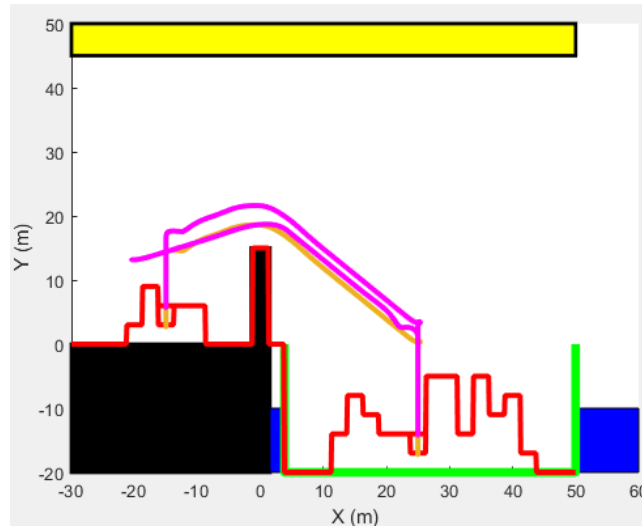


Figure 50: Trayectoria realizada para el modo simple (Rosa: Centro Spreader, Naranja: Base del Contenedor)

En la figura 51 se puede apreciar la fuerza de tiro del cable de izaje ( $F_w$ ) la cual tiene a un valor estable cuando se deja de mover la carga. Se descende hasta apoyar suavemente por lo que la fuerza pasa a valer cero, luego se toma el contenedor y se eleva manualmente un poco, y una vez se estabilice la fuerza se puede obtener el valor de la masa del contenedor como se observa en la figura 52 donde la masa medida (color azul) iguala a la masa real (color amarillo), una vez se haya medido la masa se puede acceder al modo automático.

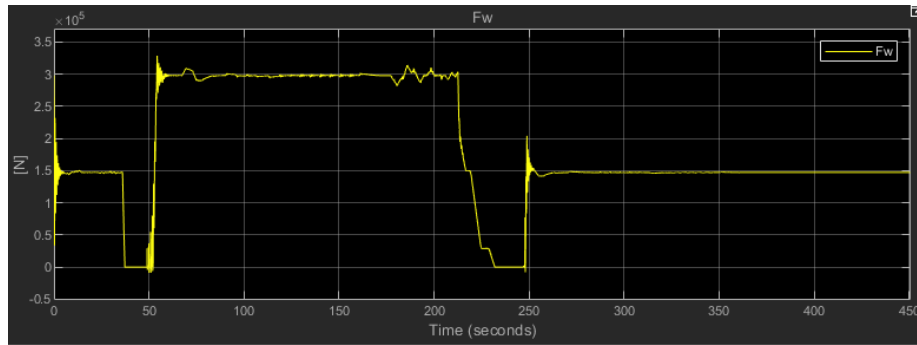


Figure 51: Ciclo simple  $F_w$

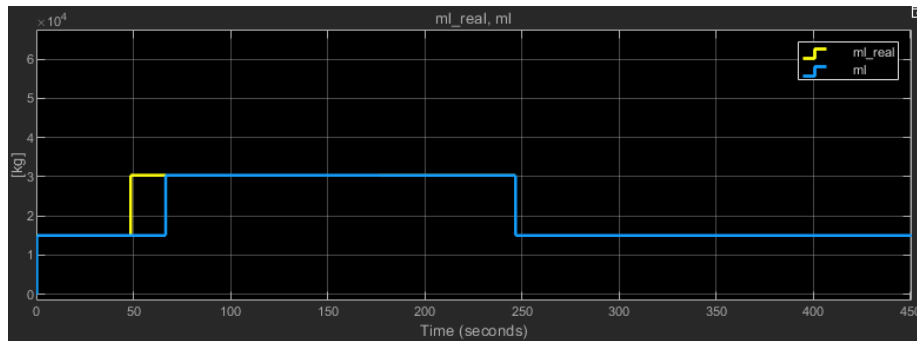


Figure 52: Ciclo simple ml

En las figuras 53 y 54 se observa como se siguen las consignas dadas por el modo automático. También se observa el movimiento en el eje y al inicio debido a el descenso para tomar la carga y luego otro descenso manual de mayor longitud. El eje x solo se controla de forma automática en ese ejemplo.

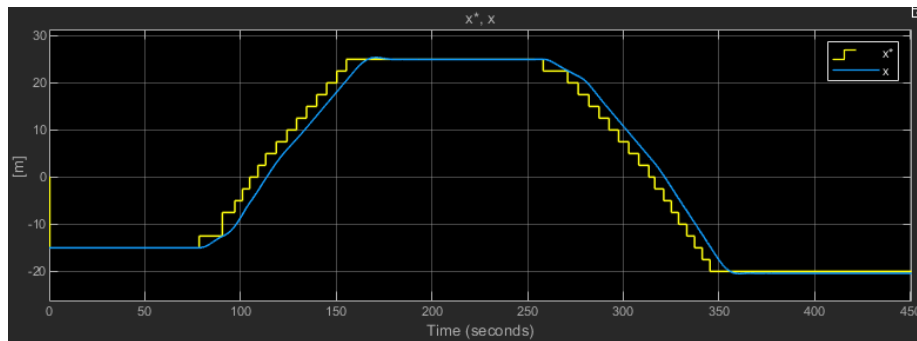


Figure 53: Ciclo simple x

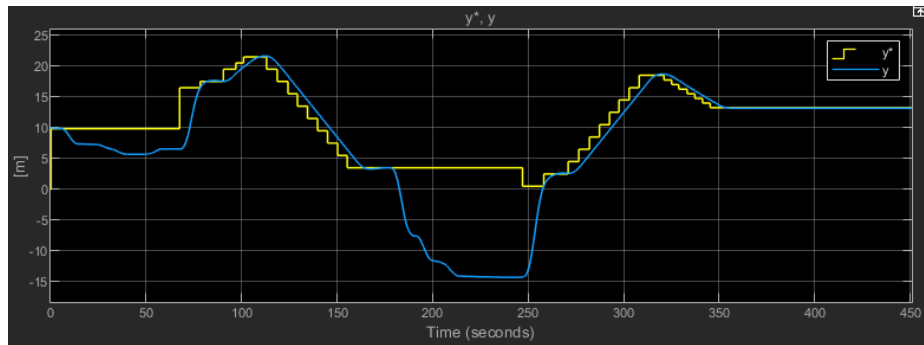


Figure 54: Ciclo simple  $y$

En las figuras 55 y 56 se observan las velocidades y aceleraciones respectivamente, tanto para el izaje como para la traslación del carro. Se aprecia que las velocidades y aceleraciones se mantuvieron por debajo de sus valores máximos.

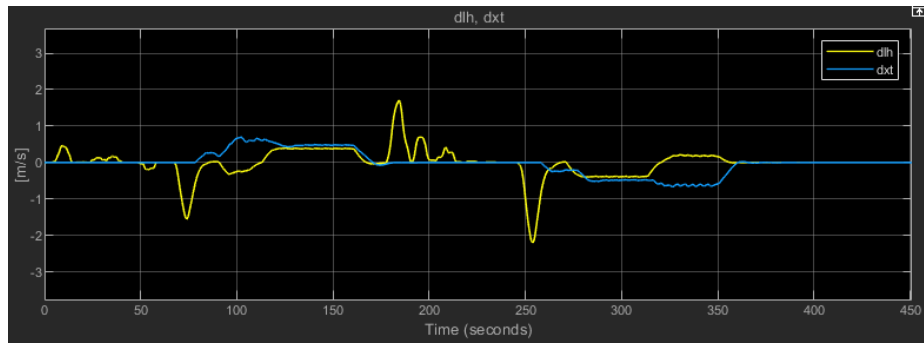


Figure 55: Ciclo simple velocidades

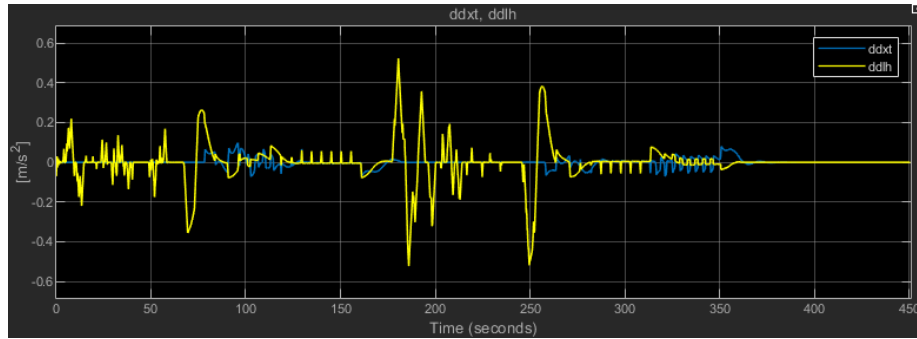


Figure 56: Ciclo simple aceleraciones

Para el caso del modo doble (figura 57) se planteó que inicialmente comienza en el muelle descargado pero debe ir a buscar la primera carga que estaría del lado muelle más a la izquierda, una vez que se encuentra sobre esta es operado manualmente hasta llegar al contenedor y se toma el mismo cerrando los twist-locks y sube un poco, una vez que la fuerza es estable se realiza la medición de la masa y en este momento se puede pasar a modo automático. Mueve la carga hasta el barco y la deposita con asistencia nuevamente del operario en modo manual, luego el modo automático lo lleva sin carga hasta otro contenedor encima del barco, se desciende y toma manualmente, se levanta un poco, y al estabilizarse al fuerza le permite nuevamente pasar al modo automático siendo así el último punto de la trayectoria del lado del muelle cerca de la sill beam, donde una vez encima de ese punto se puede descender manualmente para depositar la carga.

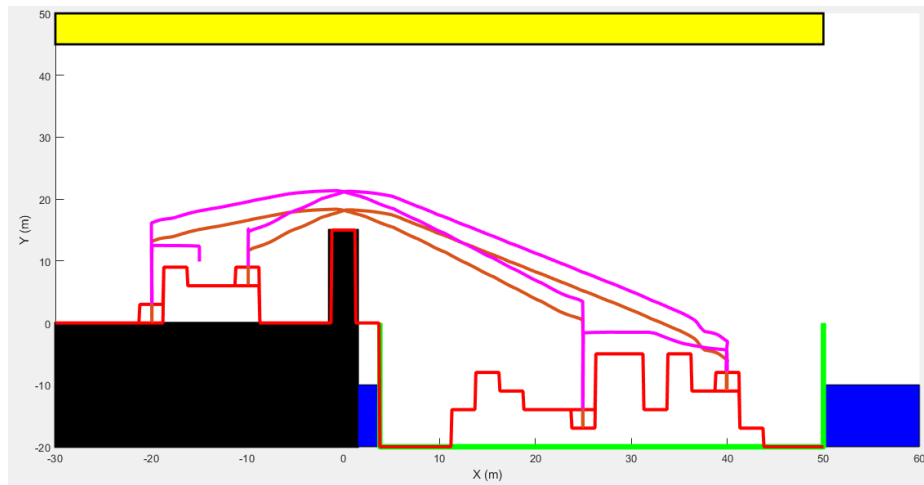


Figure 57: Ciclo doble

También se realizó la simulación utilizando Codesys. Para ello se monto únicamente el sistema físico y el Nivel 2 (por limitaciones de simulación) en Simulink. El nivel 1 y 0 fueron ejecutados por sus respectivos autómatas en Codesys. La comunicación se logra mediante el protocolo abierto OPC UA. En la imagen 58 se observa en la HMI las gráficas obtenidas del sistema físico para la posición en x y y del spreader.

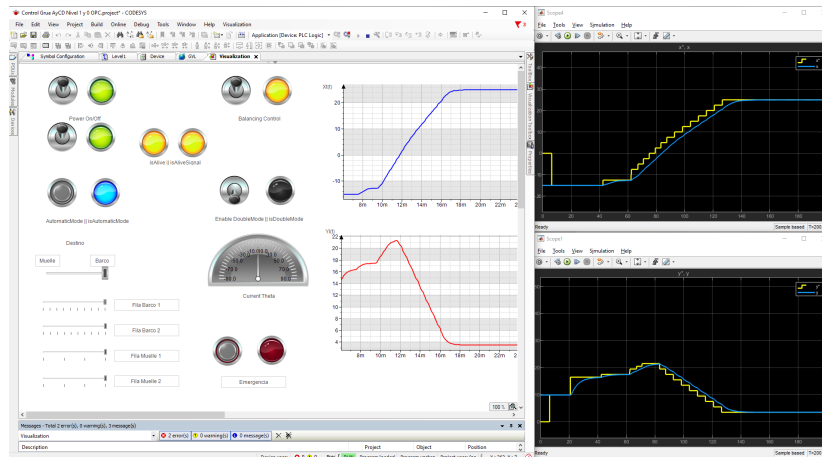


Figure 58: HMI: Con el autómata funcionando

## 6 Conclusiones

Partiendo del planteo de una aplicación real simplificada con fines académicos, en el presente proyecto se diseñó e implementó un modelo físico del sistema propuesto así como los tres niveles de control explicados anteriormente, logrando un sistema dinámico híbrido que combina la interacción de subsistemas continuos y discretos.

El objetivo del proyecto fue alcanzado satisfactoriamente, logrando trayectorias eficientes y seguras ante fallas, con un control efectivo de movimientos y posicionamiento final correcto y preciso, minimizando y prácticamente anulando oscilaciones en los finales de la trayectoria evitando pérdidas de tiempo innecesarias y mejorando el confort del operario que se ve sometido a menos aceleraciones / minuto.

El empleo del protocolo de comunicación OPC UA abre muchas puertas entre ellas el procesamiento distribuido como fue comprobado en este proyecto al emplear 2 ordenadores simultáneamente uno corriendo el autómata y el otro corriendo el modelo. Esto logra aligerar las cargas en cada dispositivo aumentando la responsividad del sistema. Esto se conoce como "Escalamiento Horizontal".

## 7 Anexo 1: Comunicaciones OPC UA

En el presente anexo se detallan las comunicaciones entre Codesys (software in the loop) y Simulink (model in the loop) a través del protocolo OPC UA. Adicionalmente se incluye el script genérico empleado para dichas comunicaciones.

OPC UA es un estándar de la norma IEC62541 para el intercambio de datos entre sensores y aplicaciones en la nube. Es un protocolo interplataforma y opensource desarrollado por la OPC Foundation.

Codesys provee emulación de PLC y logra la comunicación entre la aplicación el PLC simulado mediante el protocolo OPC UA, para asegurar esta comunicación es recomendable generar un certificado que habilita el cifrado seguro de la misma. Además se puede gestionar la creación de un usuario con permisos de acceso al PLC (más seguro) o se puede habilitar el acceso anónimo (más conveniente durante el desarrollo). En caso de tener comunicaciones cifradas se deben incorporar los certificados generados por los clientes externos (Matlab, UAExpert) al listado de certificados confiables. Estos por defecto se ubican en cuarentena al no compartir la misma firma digital que los generados por Codesys.

A continuación se muestra el script generalizado para comunicaciones OPCUA ¡- Simulink. Dicha función se puede llamar desde un "Interpreted Matlab Function Block" en simulink, alimentando a través de un Mutex todas las señales a escribir en OPC y leyendo mediante un Demux las variables obtenidas del autómata.

Posteriormente se implementó un mecanismo de throttling para limitar las operaciones de I/O de red, aumentando la performance del modelo. Por motivos de latencia, esta técnica debe emplearse con moderación.

---

```

function output = OPCCommunications(input)
    ReadKeys = ["XTarget" "YTarget" "operationMode" "YCO" "ml"
                "enableBalancing" "dlhRestoreTight" "isLoaded"];
    WriteKeys = ["currentX" "currentL" "currentTheta" "currentFw"
                 "twistlocks" "reXMax" "reXMin" "reYMax" "reYMin"];

    if (size(input) ~= size(WriteKeys))
        input = zeros(size(WriteKeys));
    end

    % connection Variables
    persistent initServer;
    persistent initNodes;
    persistent client;
    % nodes
    persistent ReadNodes;
    persistent WriteNodes;

    % initialize variables
    if isempty(initServer)
        initServer = 0;
        initNodes = 0;
    end

    % attempt connection to opcua server
    if (initServer == 0)
        server = opcuaserverinfo('localhost');
        client = opcua(server);
        client.connect();
        initServer = 1;
        disp("Connected to OPCUA server succesfully!");
    end

    % init network get/set
    if (client.isConnected && initNodes == 0)
        % read
        % Init ReadNodes type
        ReadNodes = client.getNamespace().findNodeByName(ReadKeys(1),
            "-once");
        for i = 1: size(ReadKeys, 2)
            ReadNodes(i) =
                client.getNamespace().findNodeByName(ReadKeys(i),
                    "-once");
        end
        disp("Initialized OPCUA ReadNodes!");
        % write
        % Init WriteNodes type
        WriteNodes =
            client.getNamespace().findNodeByName(WriteKeys(1),

```

```

        "-once");
    for i = 1: size(WriteKeys, 2)
        WriteNodes(i) =
            client.getNamespace().findNodeByName(WriteKeys(i),
            "-once");
    end
    disp("Initialized OPCUA WriteNodes!");
    % success
    initNodes = 1;
end

if (client.isConnected == 1 && initNodes == 1)
    % read data
    output = zeros(size(ReadNodes));
    for i = 1: size(ReadNodes, 2)
        output(i) = double(ReadNodes(i).readValue());
    end
    % write data
    for i = 1: size(WriteNodes, 2)
        WriteNodes(i).writeValue(input(i));
    end
end
end
end

```

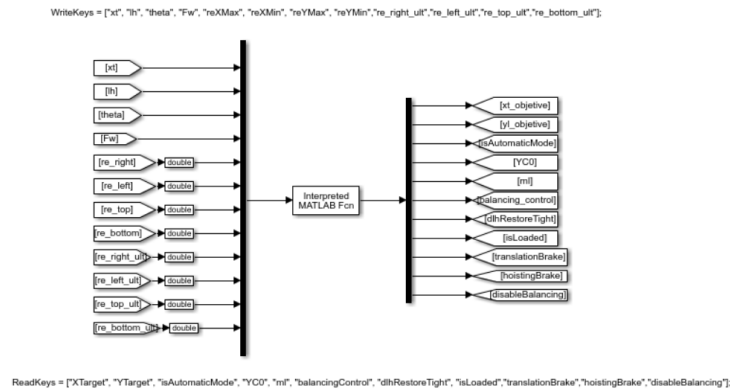


Figure 59: Bloque de función comunicaciones OPC UA

Además se comprobó la utilidad de este protocolo para lograr el procesamiento distribuido de ambas simulaciones aligerando la carga computacional en cada ordenador. Esto es conocido en la práctica como "Escalamiento Horizontal".



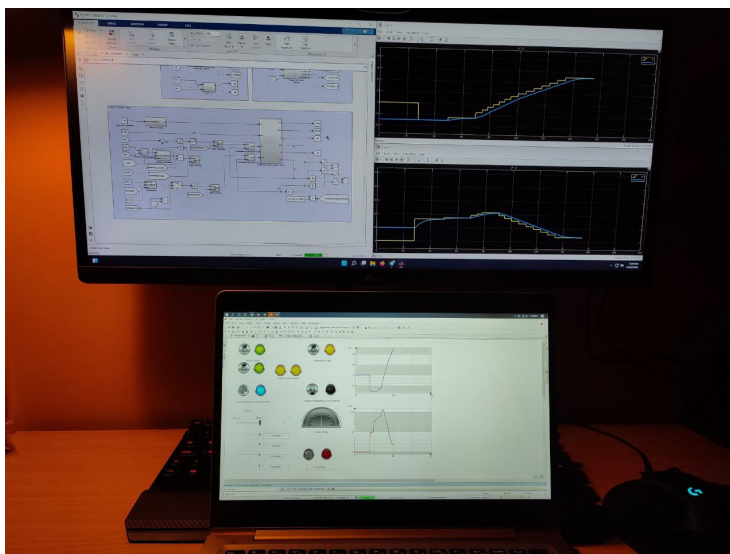


Figure 60: OPC UA - Procesamiento distribuido

## 8 Referencias

1. Guía de Trabajo - Proyecto Global Integrador AyCD - 2020 - Ing. Gabriel L. Julián
2. Apuntes de cátedra de “Autómatas y Control Discreto” y “Automática y Máquinas Eléctricas” - Ing. Gabriel L. Julián
3. Documentación de MATLAB
4. Documentación de CODESYS
5. Documentación Siemens, comunicación OPCUA - Simulink/Matlab