

Universidad Nacional de Cuyo
Facultad de Ingeniería
Ing. en Mecatrónica

Control Semi-Automático Coordinado de Grúa Portuaria de Muelle tipo Pórtico

Autómatas y Control Discreto
Proyecto Global Integrador

Profesor:
Ing. Gabriel L. Julián

Alumnos:
Martín Gabriel Quiroga
Tomás Corteggiano

Resumen

En el presente informe, se explica el desarrollo y la implementación de un autómata híbrido y protección para operación semiautomática coordinada en una grúa portacontenedores portuaria de muelle tipo pórtico lo cual involucra, primeramente el diseño, modelado conceptual y análisis de desempeño mediante simulación “Model-in-the-loop” del sistema físico y del autómata (Utilizando Matlab, Simulink y Stateflow); y posteriormente la implementación en un entorno de programación de autómatas industriales según la norma IEC 61131 (Utilizando Codesys) y su simulación “Software-in-the-loop”, junto con una emulación o co-simulación del sistema.

Índice

1. Introducción	3
2. Modelado del Sistema	5
2.1. Traslación del carro	5
2.2. Izaje de Carga	8
2.3. Análisis de la carga	10
2.3.1. Carga suspendida	11
2.3.2. Carga apoyada	12
2.4. Detección de masa	12
3. Autómata híbrido de control y protección	14
3.1. Nivel 2: Controladores de Movimiento	14
3.1.1. Controlador de Traslación del Carro	14
3.1.2. Controlador de Izaje de Carga	15
3.1.3. Controlador de Balanceo de Carga	16
3.2. Nivel 1: Control Supervisor Global	22
3.2.1. Generación de Trayectorias	26
3.3. Nivel 0: Seguridad y Protección	29
3.4. Co-simulación y servidor OPC UA	30
4. Resultados	34
4.1. Trayectoria Ciclo Simple	34
4.2. Trayectoria Ciclo Doble	38
5. Conclusiones	41
6. Referencias	42
7. Anexos	43
7.1. Función de lectura y escritura OPC UA - Matlab	43

1. Introducción

Actualmente, en todo puerto comercial del mundo se cuenta con la presencia de las grúas portuarias, que resultan un elemento imprescindible al momento de las cargas y descargas de contenedores y de todo tipo de mercancías de los grandes buques cargueros.

En este tipo de actividades económicas el tiempo es apremiante, dado que cada segundo demorado equivale a dinero perdido para las partes involucradas. Por este motivo, se busca que la velocidad de los procesos de embarque y desembarque, y por tanto de la grúa portuaria, puedan incrementarse a su nivel óptimo según lo permita la tecnología disponible.



Figura 1: Grúa Portuaria de Muelle tipo Pórtico

Una parte del recorrido para lograr este objetivo es el diseño y simulación del sistema, para poder obtener aproximaciones de la realidad que permitan conocer ventajas y defectos de las distintas estrategias de control y del sistema en sí. Esto es lo que se realizó en el proyecto descrito por el informe a continuación.

Partiendo de un modelado matemático de sistema físico, se diseña el mismo en Simulink junto con los controladores de los movimientos principales, que son la traslación horizontal del carro y el izaje vertical de la carga impulsados por motores eléctricos y con restricciones de recorrido, velocidades y aceleraciones. Además se incorpora un control automático del balanceo de la carga manteniendo el ángulo en rangos definidos por la parte de la trayectoria en que se

encuentre el proceso, con el fin de disminuir su oscilación y acelerar el posicionamiento final del contenedor. Estos tres controladores consolidan el llamado Nivel 2 del autómata.

El control supervisor global conforma el Nivel 1 del autómata y es el encargado de la operación semi-automática del proceso; en el modo manual, un operador realiza las maniobras, generalmente, de aproximación, toma de carga, descarga e inicio de la trayectoria de forma suave y cautelosa, mientras que el modo automático genera las trayectorias gruesas de forma óptima con un control coordinado de los movimientos principales y del balanceo.

La prioridad del control del sistema la tiene el Nivel 0, que consiste en un autómata aparte, reducido y confiable, encargado de la seguridad y/o protección que toma el control ante fallas críticas y/o riesgo de seguridad. Tanto este nivel como el anterior fueron implementados en Stateflow en una primera instancia a fin de acelerar el proceso de búsqueda de su funcionamiento efectivo y eficiente. Sin embargo, una vez logrado dicho objetivo, se trasladó al entorno de Codesys en el lenguaje SFC (Sequential Function Chart) regido por la norma IEC 61131 y se procedió a co-simularlo con el sistema físico y el nivel 2 funcionando en Simulink y comunicando ambos programas mediante el protocolo de comunicación industrial OPC UA.

2. Modelado del Sistema

Como primer paso se busca obtener una representación matemática del sistema físico carro-izaje-carga (con y sin contacto vertical), que resultará en un modelo dinámico no lineal en tiempo continuo, simplificado a partir de 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 como el desplazamiento en el eje x , positivo hacia la derecha, referido al borde del muelle ($x=0$). Sus movimientos horizontales tienen los siguientes límites:

- Posición x : $[-30,0$ (sobre muelle) ... $0,0$... $+50,0$ (sobre barco)]m
- Velocidad máx.: $\pm 4,0 \frac{m}{s}$ (con o sin carga)
- Aceleración máx.: $\pm 1,0 \frac{m}{s^2}$ (con o sin carga)

En la Fig. 2 se puede ver un esquema del sistema de traslación de carro, que incluye motor, reducción y rueda equivalente. El comportamiento del motor está representado por la ecuación:

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

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

La rueda equivalente se comporta según la ecuación:

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

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

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

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

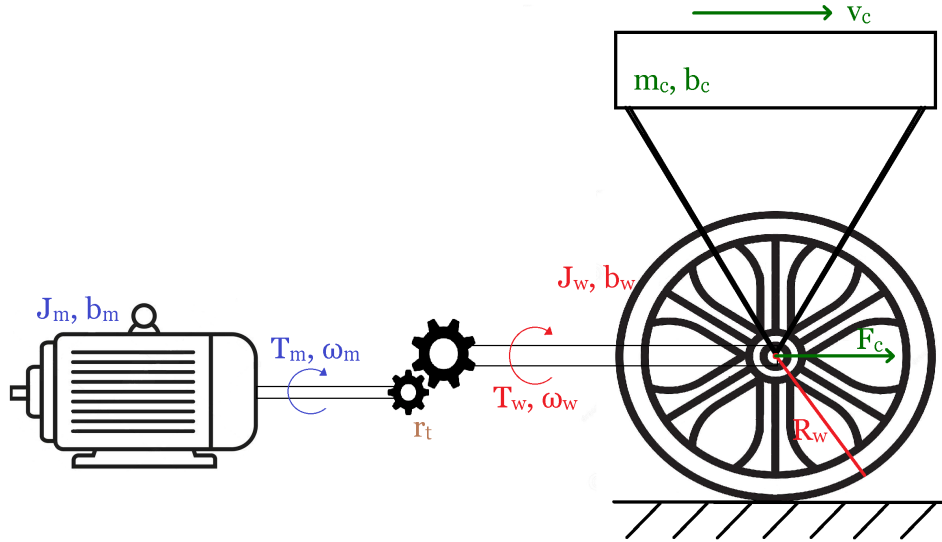


Figura 2: Modelo simplificado de traslación de carro

Reemplazando esta última relación en la Ec. (1) y despejando $T_q(t)$ se obtiene:

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

Despejando $T_q(t)$ en la Ec. (2), reemplazando en la Ec. (4) y operando se llega a:

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

Si se considera que no existe resbalamiento entre las ruedas y los rieles, se cumplen las siguientes relaciones entre la rotación de las ruedas y la traslación del carro:

$$\begin{aligned} x_c(t) = R_w \theta_w(t) &\implies \theta_w(t) = \frac{x_c(t)}{R_w} \\ \dot{x}_c(t) = R_w \omega_w(t) &\implies \omega_w(t) = \frac{\dot{x}_c(t)}{R_w} \\ \ddot{x}_c(t) = R_w \dot{\omega}_w(t) &\implies \dot{\omega}_w(t) = \frac{\ddot{x}_c(t)}{R_w} \end{aligned} \quad (6)$$

donde $x_c(t)$ es la posición del carro, R_w el radio de la rueda equivalente y $\theta_w(t)$ la posición angular de la rueda.

De forma similar se obtiene la relación entre el torque de carga en la rueda y la fuerza $F_c(t)$ que impulsa el carro:

$$T_w(t) = R_w F_c(t) \quad (7)$$

Reemplazando estas últimas relaciones en la Ec. (5) y despejando $F_c(t)$ se obtiene:

$$F_c(t) = -\frac{(J_w + J_m r_t^2)}{R_w^2} \ddot{x}_t(t) + \frac{T_m(t) r_t}{R_w} - \frac{(b_w + b_m r_t^2)}{R_w^2} \dot{x}_t(t) \quad (8)$$

Por otro lado, la ecuación de movimiento del carro está dada por:

$$m_c \ddot{x}_c(t) = F_c(t) + F_l(t) - b_c \dot{x}_c(t) \quad (9)$$

donde m_c es la masa de la carga, b_c el coeficiente de rozamiento viscoso del carro (por rodadura de la rueda respecto al riel) y $F_l(t)$ la fuerza de carga en función del ángulo de balanceo θ de la misma:

$$F_l(t) = F_w(t) \sin \theta(t) \quad (10)$$

donde $F_w(t)$ es la fuerza elástica del cable sobre el carro¹.

Reemplazando $F_l(t)$ en la Ec. (9), despejando $F_c(t)$ de esta y reemplazando en la Ec. (8) se llega a:

$$\left(m_c + \frac{J_w + J_m r_t^2}{R_w^2} \right) \ddot{x}_c(t) = F_w \sin \theta(t) + \frac{T_m(t) r_t}{R_w} - \left(b_c + \frac{b_w + b_m r_t^2}{R_w^2} \right) \dot{x}_c(t) \quad (11)$$

Se consideran los términos entre paréntesis como parámetros equivalentes, tal que:

$$M_t = m_c + \frac{J_w + J_m r_t^2}{R_w^2}$$

$$b_t = b_c + \frac{b_w + b_m r_t^2}{R_w^2} \quad (12)$$

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

Así, se obtiene una expresión de la forma:

$$M_t \ddot{x}_c(t) = F_t(t) - b_t \dot{x}_c(t) + F_w(t) \sin \theta(t) \quad (13)$$

que será la ecuación del modelo dinámico de la traslación del carro.

A continuación se presenta el diagrama de bloques correspondiente al modelo del carro:

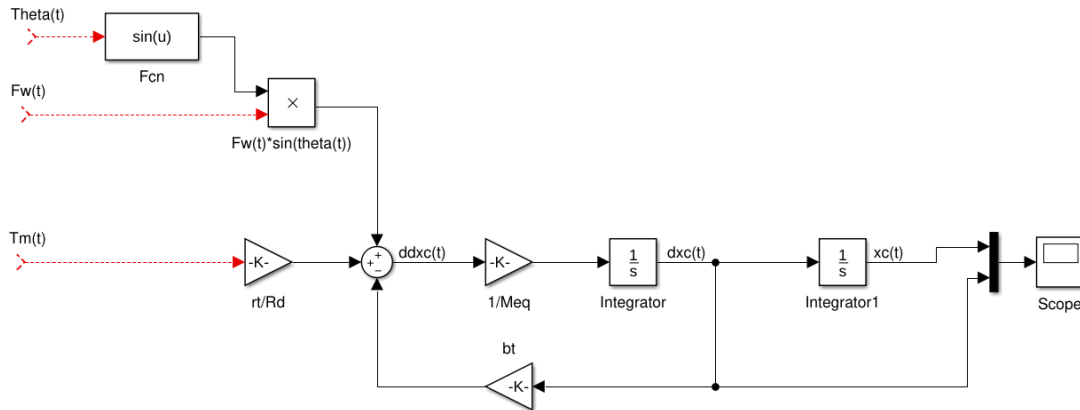


Figura 3: Diagrama de bloques de sistema carro

¹No relacionado con el subíndice de la rueda

2.2. Izaje de Carga

El movimiento de izaje se produce sobre el eje y , vertical, tomando como positivo el sentido hacia arriba y a la altura del muelle como el valor $y = 0$. Se consideran, en este movimiento, los siguientes límites y referencias:

- Posición y : $[-20.0$ (dentro del barco) ... 0.0 ... $+40.0$ (sobre barco/muelle)]
- Altura de Carro y Sistema de Izaje (fija): $Y_{t0} = +45m$
- Despeje mínimo sobre borde de muelle (viga testera o "sill beam"): $Y_{sb} = +15m$
- Velocidad máx: $\pm 1,5 \frac{m}{s}$ (Cargado con carga nominal); $\pm 3,0 \frac{m}{s}$ (Sin carga). Ver Fig. 4
- Aceleración máx: $\pm 1,0 \frac{m}{s^2}$

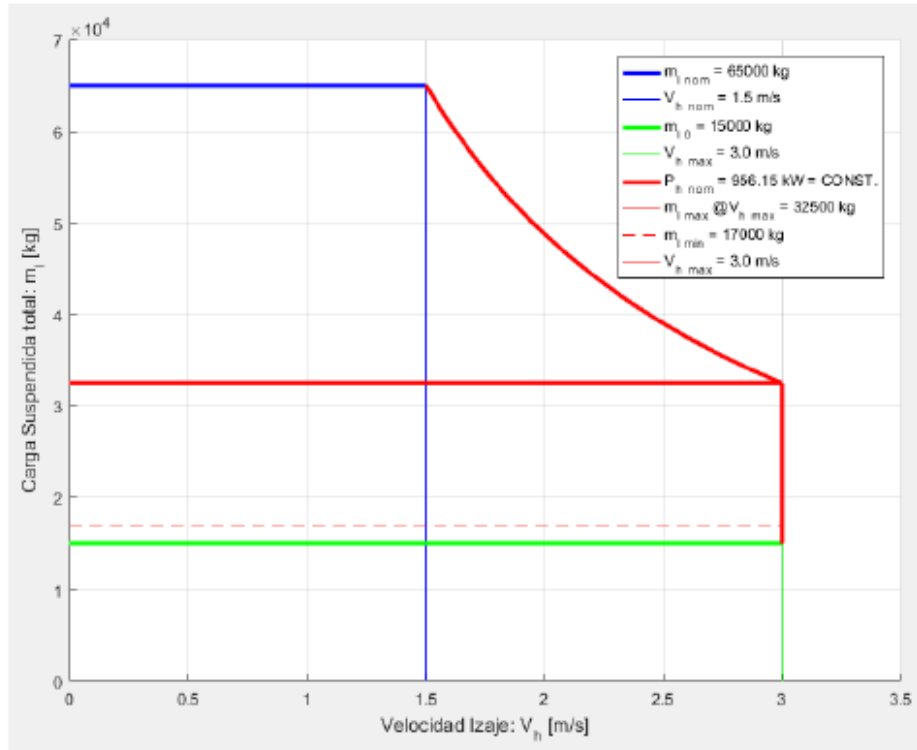


Figura 4: Característica de potencia constante - Carga suspendida vs. Velocidad de izaje

El procedimiento para encontrar el modelo del accionamiento de izaje es similar al anterior. La ecuación que representa el comportamiento del motor de izaje está dada por:

$$J_{mh}\dot{\omega}_{mh}(t) = T_{mh}(t) - T_{lh} - b_{mh}\omega_{mh}(t) \quad (14)$$

donde J_{mh} y b_{mh} son el momento de inercia y el amortiguamiento viscoso del motor respectivamente, $\dot{\omega}_{mh}(t)$ y $\omega_{mh}(t)$ la aceleración y la velocidad del eje rápido, $T_{mh}(t)$ el torque electromagnético que produce el motor y $T_{lh}(t)$ el torque de carga visto desde el eje rápido.

El cable equivalente de izaje se enrolla en un tambor cuya ecuación es:

$$J_d \dot{\omega}_d(t) = T_{qh}(t) - T_d(t) - b_d \omega_d(t) \quad (15)$$

donde J_d y b_d son el momento de inercia y el amortiguamiento viscoso del tambor respectivamente, $\dot{\omega}_d(t)$ y $\omega_d(t)$ la aceleración y velocidad del eje lento, T_{qh} el torque que impulsa al tambor y $T_d(t)$ el torque de carga que sufre el eje lento.

Los ejes de los dos sistemas anteriores están relacionados mediante una reducción con la siguiente relación de transmisión r_{th} :

$$r_{th} = \frac{\omega_{mh}(t)}{\omega_d(t)} = \frac{\dot{\omega}_{mh}(t)}{\dot{\omega}_d(t)} = \frac{T_{qh}(t)}{T_{lh}(t)} \quad (16)$$

El eje de mayor interés en este caso es el del tambor, por lo que refiere la Ec. (14) del motor a este eje, usando la Ec. (16):

$$J_{mh} r_{th} \dot{\omega}_d(t) = T_{mh}(t) - \frac{T_{qh}}{r_{th}} - b_{mh} r_{th} \omega_{mh}(t) \quad (17)$$

y se despeja el torque $T_{qh}(t)$:

$$T_{qh}(t) = r_{th} T_{mh} - J_{mh} r_{th}^2 \dot{\omega}_d(t) - b_{mh} r_{th}^2 \omega_d(t) \quad (18)$$

Reemplazando en la Ec. (15) del tambor:

$$J_d \dot{\omega}_d(t) = r_{th} T_{mh} - J_{mh} r_{th}^2 \dot{\omega}_d(t) - b_{mh} r_{th}^2 \omega_d(t) - T_d(t) - b_d \omega_d(t) \quad (19)$$

Por lo cual el torque útil en el tambor es:

$$T_d(t) = r_{th} T_{mh} - J_{mh} r_{th}^2 \dot{\omega}_d(t) - b_{mh} r_{th}^2 \omega_d(t) - J_d \dot{\omega}_d(t) - b_d \omega_d(t) \quad (20)$$

Los movimientos angulares del tambor están directamente relacionados con los movimientos de la carga y, por lo tanto, con la longitud sin enrollar del cable, tal que:

$$\omega_d(t) = \frac{\dot{l}_h(t)}{R_d} ; \quad \dot{\omega}_d(t) = \frac{\ddot{l}_h(t)}{R_d} \quad (21)$$

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

La fuerza tangencial en el tambor F_h puede expresarse como:

$$F_h(t) = \frac{-T_d(t)}{R_d} \quad (22)$$

$$F_h(t) = -\frac{r_{th}}{R_d} T_{mh} + \frac{J_{mh} r_{th}^2 + J_d}{R_d^2} \ddot{l}_h(t) + \frac{b_{mh} r_{th}^2 + b_d}{R_d^2} \dot{l}_h(t) \quad (23)$$

Por otro lado la ecuación de accionamiento de izaje, planteada para el punto tangencial entre la cuerda y el tambor es:

$$M_h \ddot{l}_h(t) = -F_h(t) - b_h \dot{l}_h(t) + F_w(t) \quad (24)$$

donde M_h es la masa del punto tangencial, F_h es la fuerza tangencial actuando en el tambor generada por un torque útil para subir o bajar la carga, l_h es la longitud del cable sin elongar y F_w es tensión del cable de izaje.

De esta forma, se obtiene la ecuación de izaje con parámetros equivalentes, remplazando la Ec. 23 en Ec. 24 y agrupando términos:

$$\left(\frac{J_{mh}r_{th}^2 + J_d}{R_d^2} + M_h\right)\ddot{l}_h(t) = \frac{r_{th}}{R_d}T_{mh} - \left(\frac{b_{mh}r_{th}^2 + b_d}{R_d^2} + b_h\right)\dot{l}_h(t) + F_w(t) \quad (25)$$

Considerando que la masa del cable es depreciable y no se consideran efectos de rozamiento entre el cable y el tambor entonces los parámetros M_h y b_h son nulos.

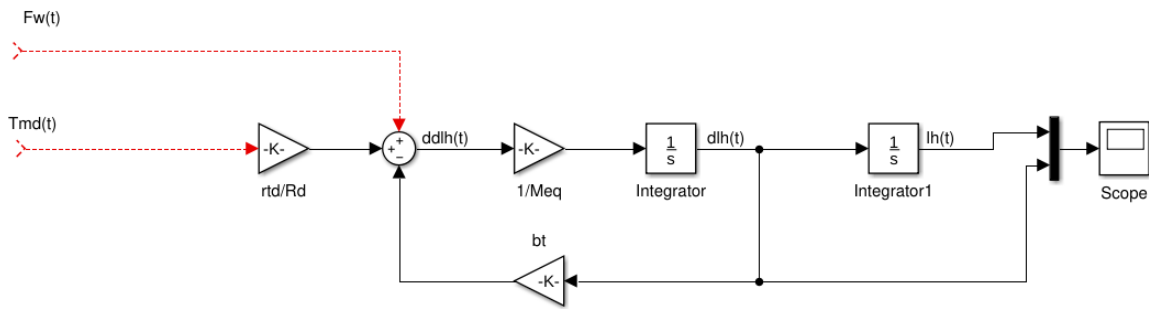


Figura 5: Diagrama de bloques de sistema izaje

2.3. Análisis de la carga

El cable de acero de izaje de masa despreciable, dispone en su extremo un gancho, cuya masa es de 15000 kg, al cual se acoplará un contenedor. La masa del contenedor podrá variar desde los 2000 kg, estando vacío, hasta una masa nominal de 50000 kg. La carga total (gancho + contenedor) se puede encontrar en dos estados, apoyada o suspendida, por lo cual, es necesario considerar los siguientes parámetros de contacto:

- Rigidez y fricción vertical (compresión): $K_{cy} = 1,3 \times 10^6 \frac{kN}{m}$; $b_{cy} = 500 \frac{kN}{s}$
- Fricción horizontal (arrastre): $b_{cx} = 1000 \frac{kN}{m/s}$

El cable equivalente (8 cables) seguirá un modelo elástico amortiguado trabajando a tracción y siempre tensado por la acción de la aceleración gravitatoria cuando el cabezal, con o sin contenedor, esté suspendido. El cable no soporta compresión; en tal caso, se flexiona o se padea. Por lo cual se consideran dos parámetros importantes del cable:

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

Se considera a $y_{c0}(x, t)$ como el perfil de obstáculos o de apoyo vertical para cada posición x , el cual se modificará en función del tiempo t cuando se desarrollen las maniobras de carga y descarga.

2.3.1. Carga suspendida

Como se mencionó anteriormente, si la carga se encuentra suspendida ($y_l(t) > y_{c0}(x, t)$), el cable trabajará siempre a tracción lo cual implica que el mismo tendrá una deformación elástica y, por tanto, la longitud del cable elongado desenrollado fuera del tambor $l(t)$, será mayor que la longitud desarrollada fuera del tambor sin elongar $l_h(t)$.

$$\begin{aligned} l(t) &\geq l_h(t) \\ l(t) &= l_h(t) + \Delta l \end{aligned} \quad (26)$$

Considerando la rigidez a tracción y el amortiguamiento y asumiendo que las deformaciones del cable se encuentran dentro del límite elástico del material, es posible hacer uso de la ley de Hook para determinar la tensión del cable de la siguiente manera.

$$F_w(t) = K_w(l(t) - l_h(t)) + b_w(\dot{l}(t) - \dot{l}_h(t)) \quad (27)$$

Luego, los movimientos de traslación en x e izaje en y de la carga suspendida, se determinan como:

$$m_l \ddot{x}_l(t) = -F_w(t) \sin(\theta(t)) \quad (28)$$

$$m_l \ddot{y}_l(t) = F_w(t) \cos(\theta(t)) - m_l g \quad (29)$$

El sistema carro-izaje-carga presenta restricciones geométricas que aportan ecuaciones algebraicas adicionales, las cuales se pueden determinar a partir de la Fig. 6.

$$x_l(t) = x_t(t) + l(t) \sin(\theta(t)) \quad (30)$$

$$y_l(t) = y_{t0} - l(t) \cos(\theta(t)) \quad (31)$$

O, equivalentemente, el ángulo de balanceo de la carga, θ , y la longitud del cable, $l(t)$, se definen como:

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

$$l(t) = \sqrt{[x_l(t) - x_t(t)]^2 + [y_{t0} - y_l(t)]^2} \quad (32)$$

Siendo que se considera el efecto de amortiguamiento en el cable por fricción de los hilos del mismo, es necesario calcular la velocidad de deformación del cable para obtener la componente $b_w(\dot{l}(t) - \dot{l}_h(t))$ de la tensión F_w . Para ello, se obtiene $\dot{l}(t)$ derivando la expresión (32) respecto al tiempo:

$$\dot{l}(t) = \frac{1}{\sqrt{(x_l(t) - x_t(t))^2 + (y_{t0} - y_l(t))^2}} [(x_l(t) - x_t(t))(\dot{x}_l(t) - \dot{x}_t(t)) - \dot{y}_l(t)(y_{t0} - y_l(t))] \quad (33)$$

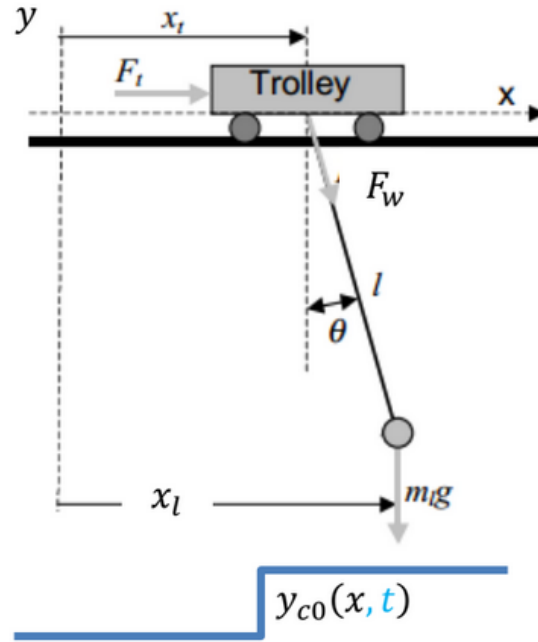


Figura 6: Modelo físico idealizado

2.3.2. Carga apoyada

En caso de que la carga se encuentre apoyada, es necesario incluir un efecto de compresión en la dirección del eje coordenado y , como también un efecto de arrastre en el eje coordenado x . Entonces, cuando $y_l(t) \leq y_{c0}(x, t)$, las ecuaciones de traslación de la carga se modelan como:

$$m_l \ddot{x}_l(t) = -F_w(t) \sin(\theta(t)) - b_{cx} \dot{x}_l(t) \quad (34)$$

$$m_l \ddot{y}_l(t) = F_w(t) \cos(\theta(t)) - m_l g + K_{cy}(y_{c0}(x, t) - y_l(t)) - b_{cy} \dot{y}_l(t) \quad (35)$$

Luego, la ecuación del cable elástico amortiguado, flojo al apoyar la carga, con flexión o pandeo, tal que $l(t) < l_h(t)$, es:

$$F_w(t) = 0(l(t) - l_h(t)) + b_w(\dot{l}(t) - \dot{l}_h(t)) \quad (36)$$

2.4. Detección de masa

Dado que se puede medir la flexión del eje del tambor de izaje y, por tanto, indirectamente la tensión del cable, es posible determinar la masa de la carga como $m_l = F_w/9,81$ cuando el sistema se encuentra estable. 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 gancho 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

```

graph LR
    Add_mass[1] --> Sum((+))
    Rand_mass[2] --> Sum
    Rand_mass --> Cond{rand > 0}
    Cond --> Sum
    Sum --> m[15000]
  
```

The diagram illustrates the 'add_random_mass' block. It takes two inputs: 'Add_mass' (value 1) and 'Rand_mass' (value 2). The 'Rand_mass' input is also connected to a conditional block labeled 'rand > 0'. The output of this conditional block is connected to the '+' input of a summing junction. The 'Add_mass' input is also connected to the '+' input of the summing junction. The output of the summing junction is 15000, which is the value of 'm'.

The diagram shows a control system for a mass. It starts with a reference input '1' (labeled 'Fw') entering a 'Peak_theta_detection1' block. This block has a 'Reset' input (a square wave icon) and two outputs: 'Amplitude' and 'Logical_output'. The 'Amplitude' output goes to a gain block '-K' with a value of '1/9.81'. The 'Logical_output' output goes to a 'Stable_Fw_flag' block (labeled '2'). The output of the gain block goes to a 'Mass' block (labeled '1').

13

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

En la subsección a continuación, se implementan los controladores PID para los movimientos de traslación del carro, del izaje de la carga y del balanceo de la misma.

3.1.1. Controlador de Traslación del Carro

El control de movimiento del carro, fue implementado mediante un controlador PID, cuyas ganancias fueron sintonizadas mediante el método de ubicación de polos. Los polos del sistema carro se obtuvieron a partir del polinomio característico del sistema. El mismo fue obtenido aplicando la transformada de Laplace al modelo matemático representado en la ecuación 13. El polinomio característico resultante fue:

$$P_c = s(M_t s + b_t) \quad (37)$$

Luego los polos del sistema carro se encuentran en $p_1 = -0,4576 \frac{rad}{s}$ y en $p_2 = 0 \frac{rad}{s}$.

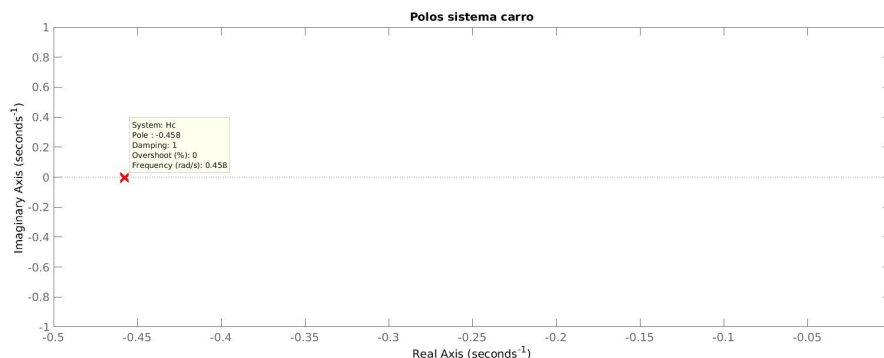


Figura 10: Polos del sistema carro

Conociendo el polo mas rápido del sistema p_1 , se ubican los polos del controlador entre 8 y 10 veces hacia la izquierda del mismo. Por lo cual, $\omega_{pos} = 10\omega_{p1}$. Para definir completamente el polinomio deseado, se define a $n = 2,5$ y luego las ganancias quedan definidas como:

$$ba_t = (n\omega_{pos}M_t)\frac{R_W}{r_t} = 22500$$

$$Ksa_t = (n\omega_{pos}^2M_t)\frac{R_W}{r_t} = 1,0295e+05$$

$$Ksia_t = (\omega_{pos}^3M_t)\frac{R_W}{r_t} = 1,8843e+05$$

A fin de evitar bloques derivativos en el sistema, cabe destacar que se hace uso del error entre la velocidad de translación deseada y real del carro.

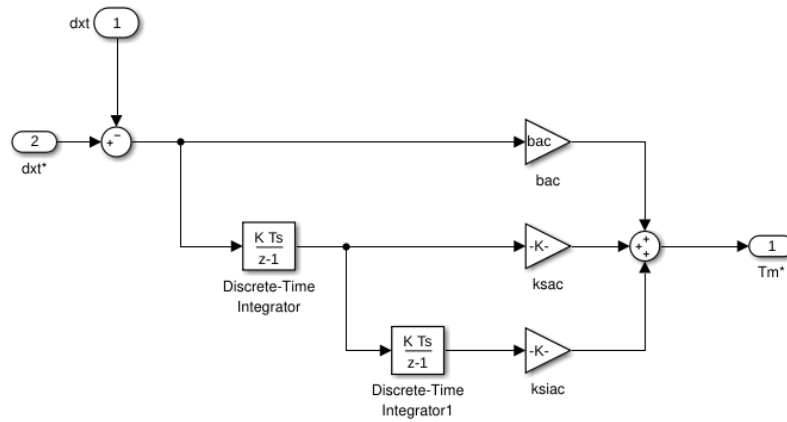


Figura 11: Diagrama de bloques del controlador discreto del sistema carro

3.1.2. Controlador de Izaje de Carga

Respecto al controlador del sistema de izaje, el procedimiento para la obtención de las ganancias del controlador PID fue similar al realizado anteriormente para el controlador del carro. El polinomio característico resultante del sistema izaje es:

$$P_c = s(M_h s + b_h) \quad (38)$$

Luego, los polos del sistema izaje se encuentran en $p_1 = -0,5998 \frac{rad}{s}$ y en $p_2 = 0 \frac{rad}{s}$.

Conociendo el polo mas rápido del sistema p_1 , se ubican los polos del controlador entre 8 y 10 veces hacia la izquierda del mismo. Por lo cual, $\omega_{pos} = 10 * \omega_{p1}$. Para definir completamente el polinomio deseado, se define a $n = 2,5$ y luego las ganancias quedan definidas como:

$$ba_{iz} = -(n\omega_{pos}M_h)\frac{R_d}{r_{th}} = -18000$$

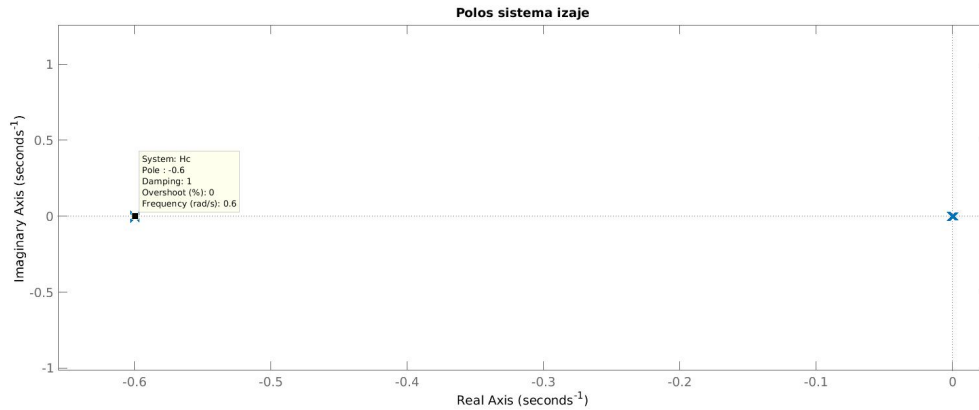


Figura 12: Polos del sistema izaje

$$Ksa_{iz} = -(n\omega_{pos}^2 M_h) \frac{R_d}{r_{th}} = -1,0797e+05$$

$$Ksia_{iz} = -(\omega_{pos}^3 M_h) \frac{R_d}{r_{th}} = -2,5905e+05$$

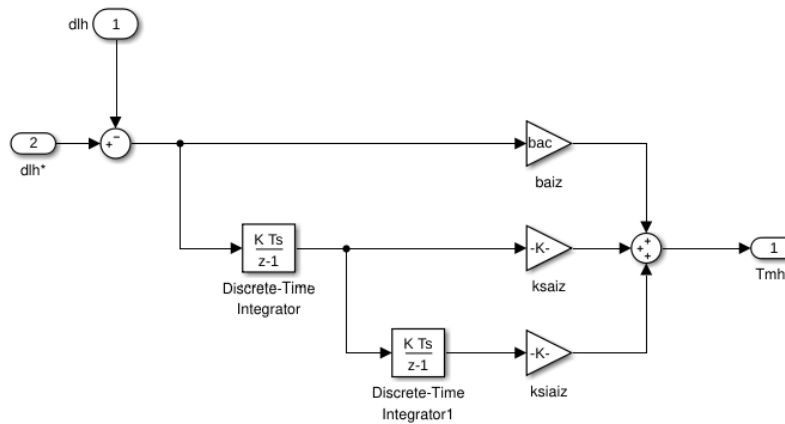


Figura 13: Diagrama de bloques del controlador discreto del sistema izaje

3.1.3. Controlador de Balanceo de Carga

La carga tendrá cierto balanceo no deseado en el plano vertical cuando empiece el movimiento horizontal, por lo que se implementa un controlador adicional al de traslación de carro con el fin de amortiguar la oscilación de la trayectoria alrededor del punto fijo de equilibrio dinámico y su oscilación remanente una vez detenido el carro en la posición final. Para ello, la máquina cuenta con un sensor de ángulo instantáneo del cable entre la carga y el carro, con respecto a la vertical. Las especificaciones de desempeño de dicho controlador son:

- 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: $\pm 1^\circ$.

La compensación del error de ángulo será del tipo PD con ganancias ajustadas en función de la altura de izaje, de la masa de la carga y del tipo de movimiento (Aceleración o Velocidad constante). Se parte de la formulación de Lagrange del sistema carro-péndulo, como el que se muestra en la Fig. 14.

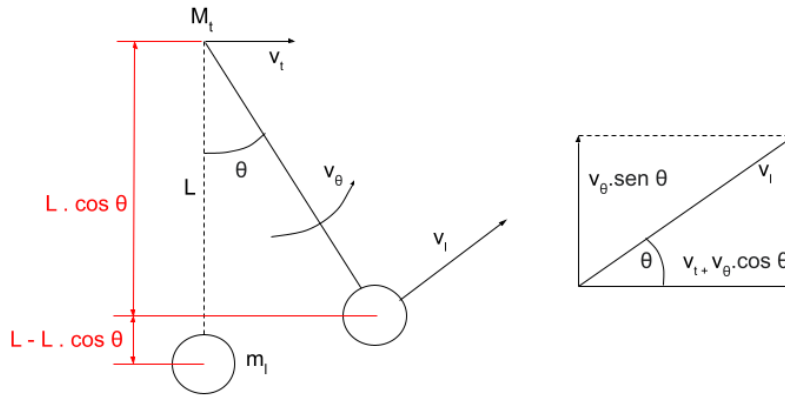


Figura 14: Sistema carro - péndulo

A dicho sistema le corresponden las siguientes ecuaciones:

$$v_t = \dot{x}_t ; v_\theta = \dot{\theta}l$$

$$v_l^2 = \left(\dot{x}_t + \dot{\theta}l \cos \theta\right)^2 + \left(\dot{\theta}l \sin \theta\right)^2$$

$$v_l^2 = \dot{x}_t^2 + 2\dot{x}_t\dot{\theta}l \cos \theta + \dot{\theta}^2l^2 \cos^2 \theta + \dot{\theta}^2l^2 \sin^2 \theta$$

$$v_l^2 = \dot{x}_t^2 + 2\dot{x}_t\dot{\theta}l \cos \theta + \dot{\theta}^2l^2$$

tal que la energía cinética total resulta:

$$K = \frac{1}{2}M_tv_t^2 + \frac{1}{2}m_lv_l^2$$

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

y la energía potencial:

$$U = m_lgl(1 - \cos \theta)$$

El Lagrangiano se obtiene como:

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

Se usan la posición del carro y el ángulo como coordenadas generalizadas, entonces haciendo la ecuación de Lagrange correspondiente a x_t es:

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

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

y para la coordenada θ 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 aproximan a las siguientes ecuaciones:

$$(M_t + m_l) \ddot{x}_t + m_l l \ddot{\theta} = F_t - b_t v_t \quad (39)$$

$$\ddot{x}_t + l \ddot{\theta} + g \theta = 0 \quad (40)$$

Despejando la aceleración del carro de la Ec. (13) se obtiene:

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

Reemplazando en la Ec. (40) y despejando F_t resulta:

$$F_t = -m_t (l \ddot{\theta} + g \theta) + b_t v_t \quad (41)$$

Despejando F_t de la Ec. (39), igualando con la Ec. (41) y operando un poco se llega a:

$$(M_t + m_l) \ddot{x}_t = -(M_t + m_l) l \ddot{\theta} - M_t g \theta \quad (42)$$

Aplicando la transformada de Lagrange 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 despejar la función de transferencia para una entrada $V_t(s)$ y salida $\Theta(s)$:

$$G_p(s) = \frac{\Theta(s)}{V_t(s)} = \frac{-(M_t + m_l) s}{(M_t + m_l) s^2 + M_t g} \quad (43)$$

Del denominador de la función de transferencia obtenida en la Ec. (43) se puede obtener la frecuencia natural del sistema péndulo:

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

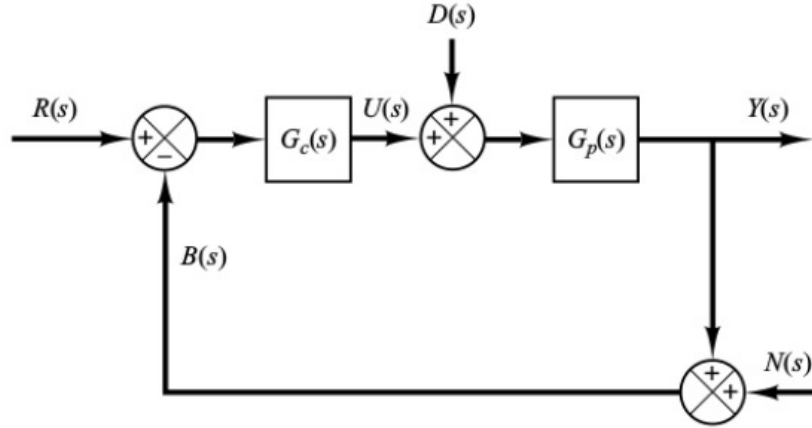


Figura 15: Diagrama de bloques con entrada de perturbación.

El efecto producido por el controlador de balanceo propuesto puede considerarse como una perturbación al sistema en la señal dada por el controlador de traslación de carro, similar a como se muestra en la Fig. 15, donde $D(s)$ representa dicha perturbación.

La función de transferencia de dicho sistema respecto a la entrada de perturbación es:

$$G_{yd} = \frac{Y(s)}{D(s)} = \frac{G_p}{1 + G_c G_p} \quad (44)$$

En el presente caso, G_p es la función de transferencia obtenida en la Ec. (43) y G_c es la función de transferencia del controlador PD que se muestra a continuación:

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

Reemplazando todo en la Ec. (44), resulta:

$$\frac{\Theta(s)}{V_t(s)} = \frac{G_p(s)}{1 + G_c(s)G_p(s)} = \frac{\frac{-(M_t + m_l)s}{(M_t + m_l)ls^2 + M_t g}}{1 + \frac{-(M_t + m_l)s}{(M_t + m_l)ls^2 + M_t g} (K_p + K_d s)}$$

Operando algebraicamente se obtiene el siguiente denominador:

$$Den(s) = s^2 - \frac{K_p}{l - K_d} s + \frac{M_t g}{(M_t + m_l)(l - K_d)} \quad (45)$$

Por otro lado, para aplicar el método de asignación de polos se propone el siguiente polinomio, dado por la frecuencia natural del sistema y amortiguamiento crítico ($\zeta = 1$):

$$(s - \omega_n) = s^2 - 2\omega_n s + \omega_n^2 \quad (46)$$

Comparando los términos de igual grado entre la Ec.(45) y la Ec. (46), se obtienen las ganancias para el controlador:

$$\begin{aligned} \omega_n^2 &= \frac{M_t g}{(M_t + m_l)(l - K_d)} \Rightarrow K_d = l - \frac{M_t g}{(M_t + m_l)\omega_n^2} \\ -2\omega_n &= -\frac{K_p}{(l - K_d)} \Rightarrow K_p = 2\omega_n (l - K_d) \end{aligned}$$

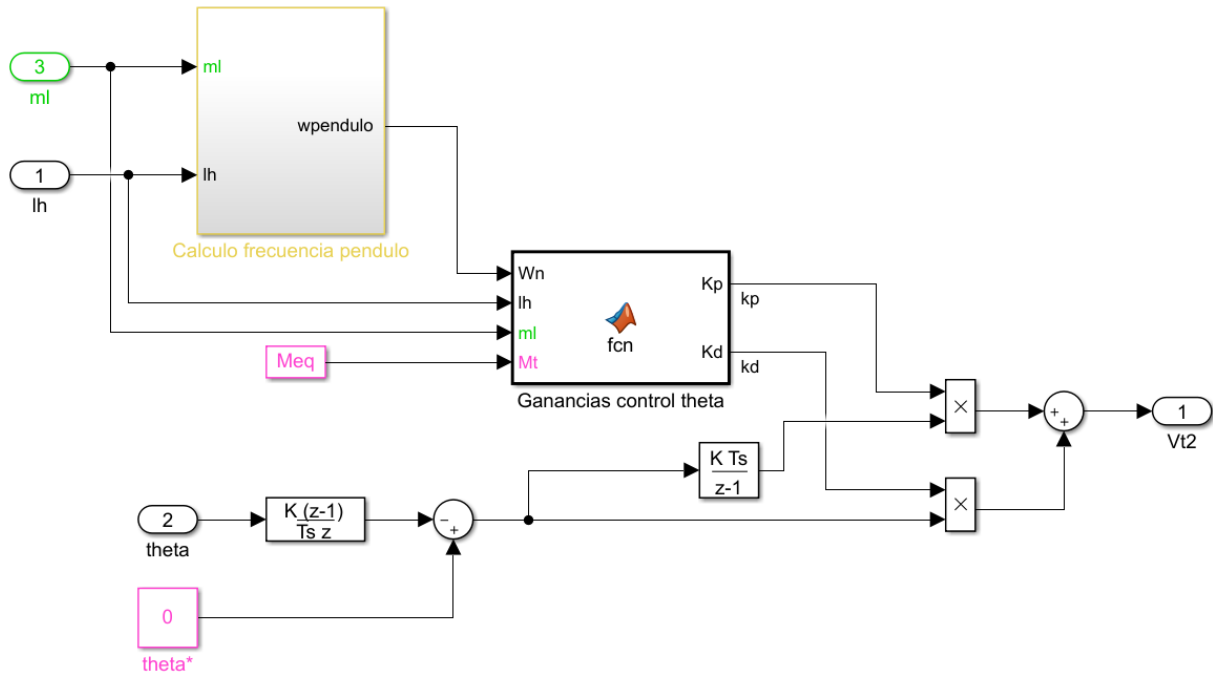


Figura 16: Diagrama de bloques del controlador de balanceo

Este sistemas de ecuaciones se implementó en Simulink, resultando en el diagrama de la Fig. 16.

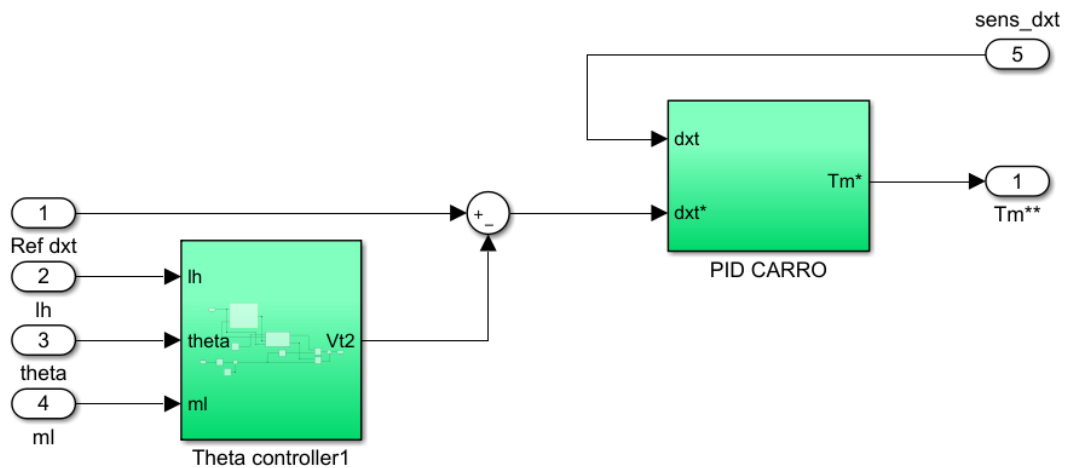


Figura 17: Control de balanceo y de traslación de carro.

Dado que la salida del controlador de balanceo, por la manera en que se obtuvo, es una señal de velocidad, se debe convertir en torque para que realmente se pueda ver como una perturbación. Por lo que se resolvió que resulta conveniente restarla a la señal de velocidad del carro medida de la planta y que esta diferencia sea la entrada al controlador de traslación como se puede ver en la Fig. 17.

En la Fig. 18 puede observarse el diagrama compuesto por el modelo del sistema físico junto con los controladores del nivel 2, donde distinguen según los colores azul, rojo y amarillo las señales correspondientes al control del carro, control de izaje y bloque de la planta respectivamente.

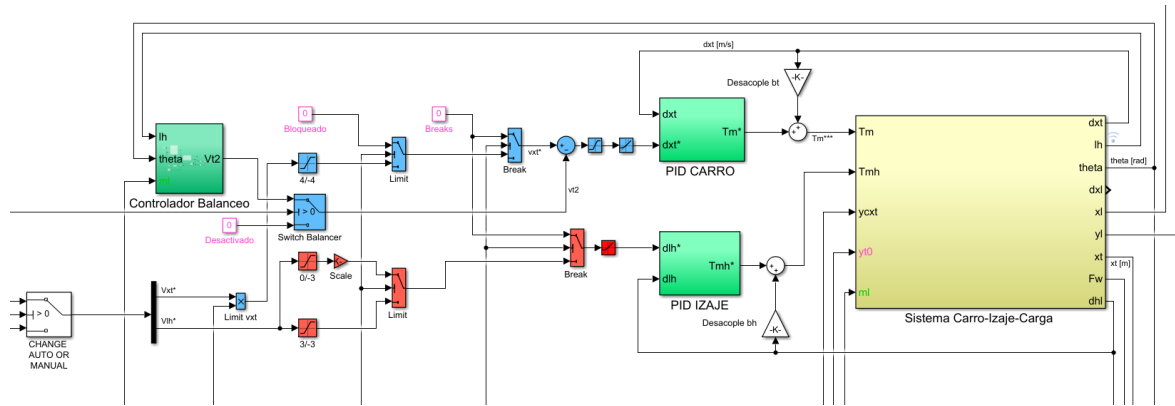


Figura 18: Sistema físico y controladores de movimiento.

3.2. Nivel 1: Control Supervisor Global

En esta subsección, se detalla el diseño del autómata que incluye los modos y ciclos de operación, toma de carga, generación de trayectorias, entre otros, además de manejar los comandos del operador y los límites normales de operación.

En un primer momento, se planteó el autómata en Stateflow para tener una mayor facilidad en realizar pruebas y simulaciones y acelerar el proceso de desarrollo, obteniéndose el diagrama general de la Fig. 19, que luego se mostrarán en mayor detalle en las figuras de Codesys.

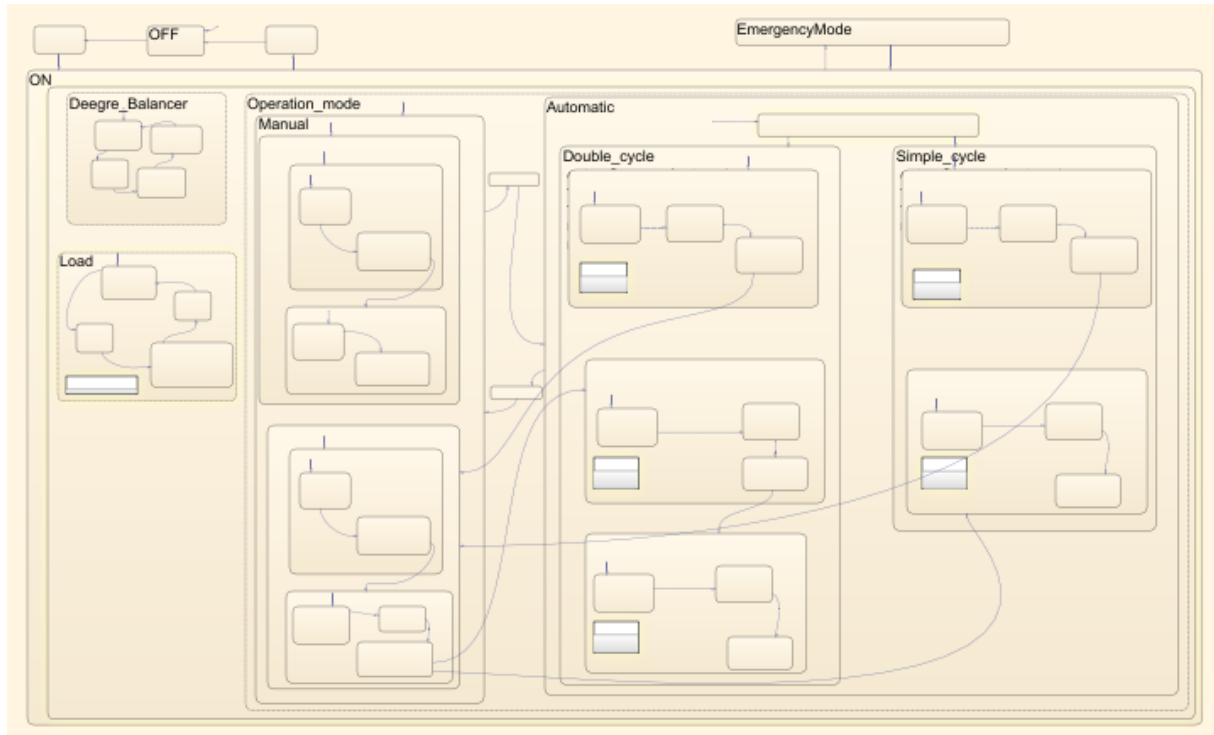


Figura 19: Autómata planteado en Stateflow

Una vez logrado el funcionamiento deseado, se procedió a migrar dicho autómata a Codesys en el lenguaje de programación gráfico SFC (Sequential Function Chart) especificado según la norma IEC 61131-3.

En este nivel del autómata se operan 3 etapas encapsuladas de forma paralela: Balanceo, Carga y Modo de Operación, como se puede ver en la Fig. 21. En el estado de **Balanceo** simplemente se controla de forma binaria si el control de ángulo está encendido o no. El de **Carga** funciona de forma similar con el estado del spreader, indicando si está cargado o vacío, pero además genera una masa aleatoria simulando un contenedor desconocido.

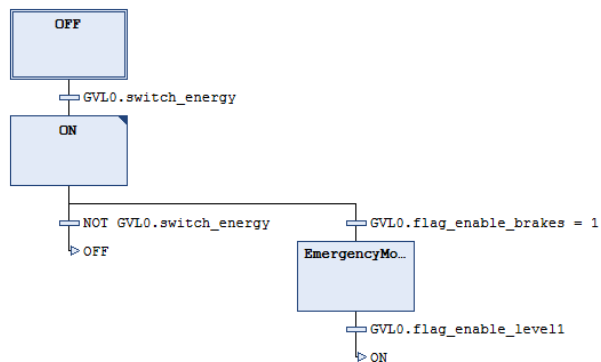


Figura 20: Nivel 1 en Codesys

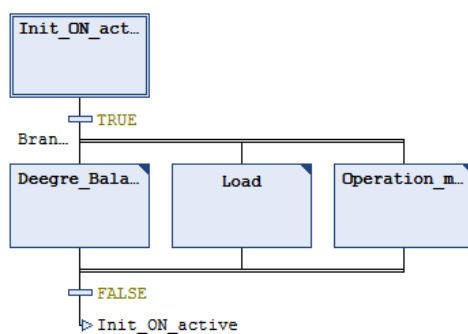


Figura 21: POU de Nivel 1 activo

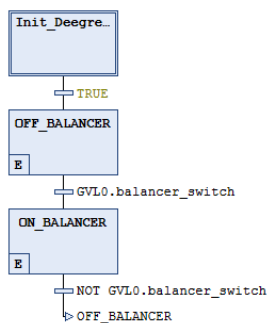


Figura 22: POU balanceo

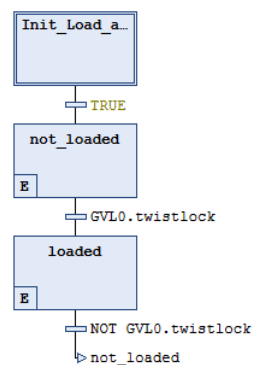


Figura 23: POU carga

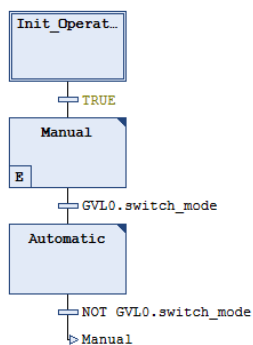


Figura 24: POU modo de operación

El estado de **Modo de Operación** selecciona alternadamente entre el modo Manual y el modo Automático para definir las trayectorias de la carga. El **Modo Manual** se emplea para maniobras de aproximación e inicios de trayectorias, por lo que dentro de esta etapa se encuentran dichas aplicaciones con y sin carga, como puede verse en las Fig. 25, 26 y 27.

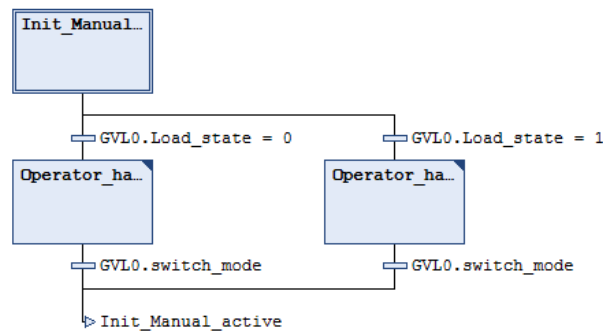


Figura 25: POU del Modo Manual

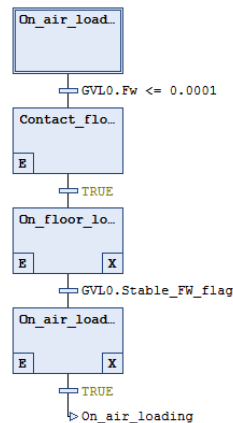


Figura 26: Secuencia de toma de carga.

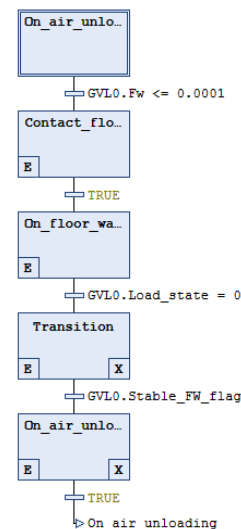


Figura 27: Secuencia de descarga

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 y aumentar la productividad, disminuyendo a su vez la fatiga del operador. Estas trayectorias generadas automáticamente pueden subclasificarse en Ciclo Simple y Ciclo Doble. Estas etapas se pueden observar en las Fig. 28, 29 y 30.

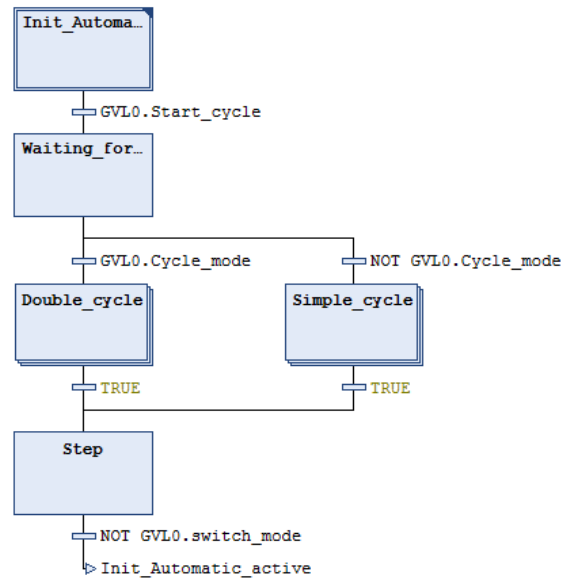


Figura 28: Encapsulado del Modo Automático

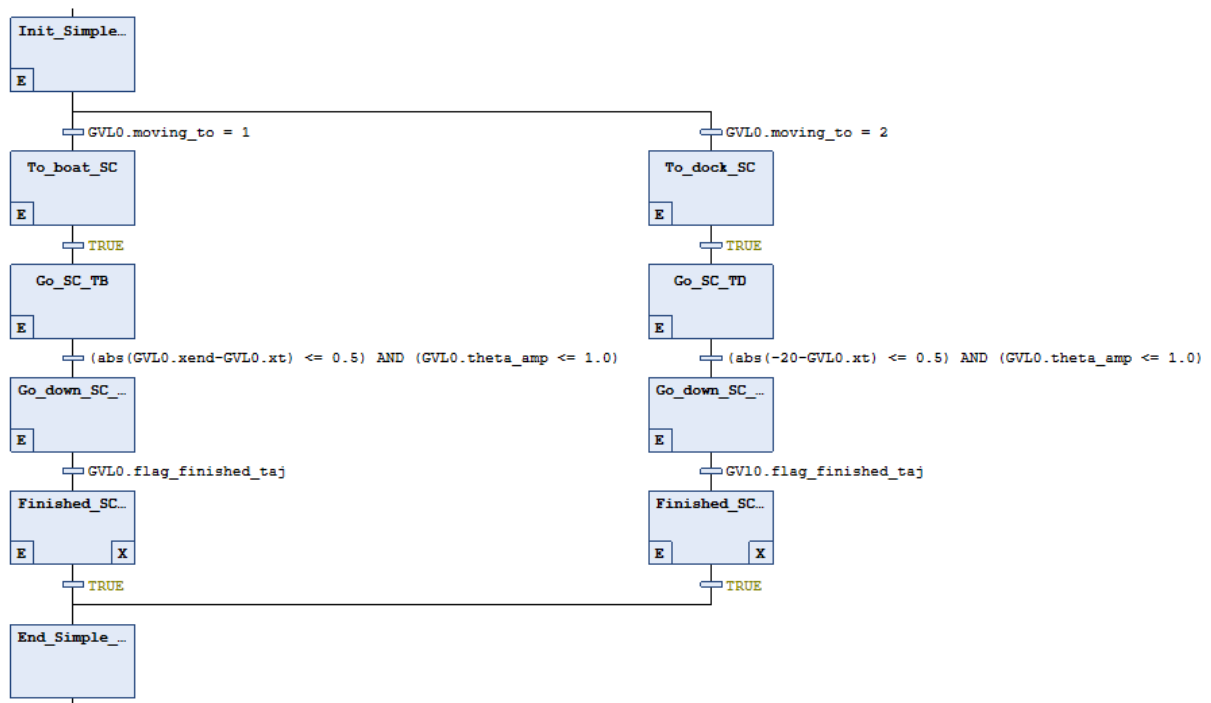


Figura 29: Secuencia de Ciclo Simple

Dentro de cada tipo de ciclo, se encuentran las funciones encargadas de la generación de trayectorias que dependen del estado actual del sistema y de las posiciones objetivos.

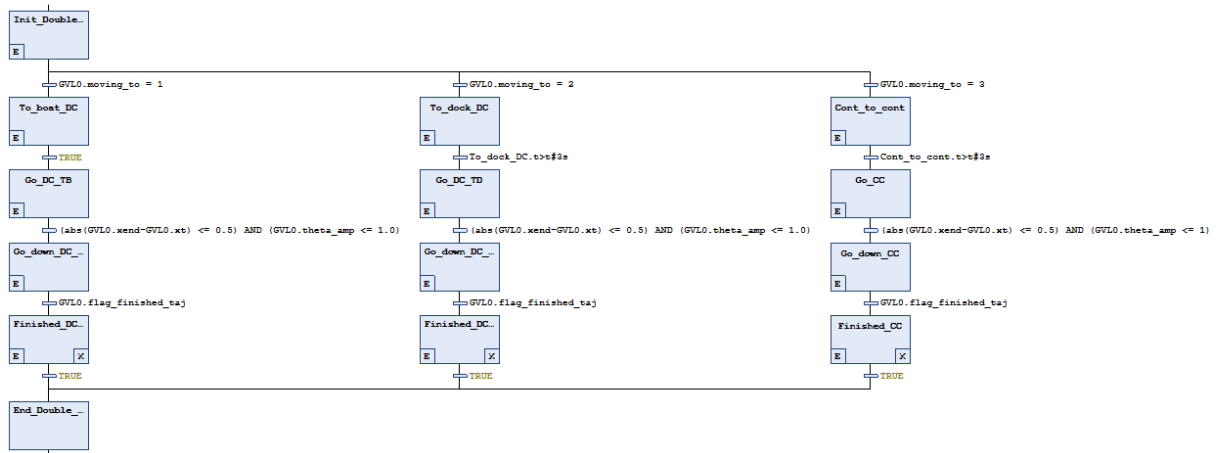


Figura 30: Secuencia de Ciclo Doble

3.2.1. Generación de Trayectorias

En la siguiente sección, se explica el algoritmo para la generación de trayectorias sin colisiones, tanto desde el muelle hacia el barco como del barco hacia el muelle, con o sin carga en cualquiera de ambas. Este algoritmo tiene como objetivo, además de evitar accidentes, optimizar el tiempo de carga o descarga del barco.

3.2.1.1 Trayectoria hacia el barco

Se define el punto cero como el punto de partida del gancho o de la base del contenedor apoyado sobre el muelle, dependiendo si la maniobra será de carga o de descarga

Se analizan las alturas de las columnas de contenedores a la izquierda de la columna objetivo y de la viga testera para determinar cual es la máxima altura a la que se debe subir, teniendo en cuenta también una distancia de seguridad. A partir de esta, teniendo en cuenta velocidades y aceleraciones máximas se calcula cuanto subir verticalmente (si fuese necesario) y la diagonal que va a trazar el gancho o la carga.

Luego, se calcula la trayectoria horizontal hasta el punto donde se puede empezar a descender en diagonal al gancho o la carga. Finalmente, se desciende verticalmente hasta la parte superior de la columna objetivo. Finalmente, aplicando las consideraciones de velocidades y aceleraciones máximas se obtiene la trayectoria que se muestra en la Fig. 31

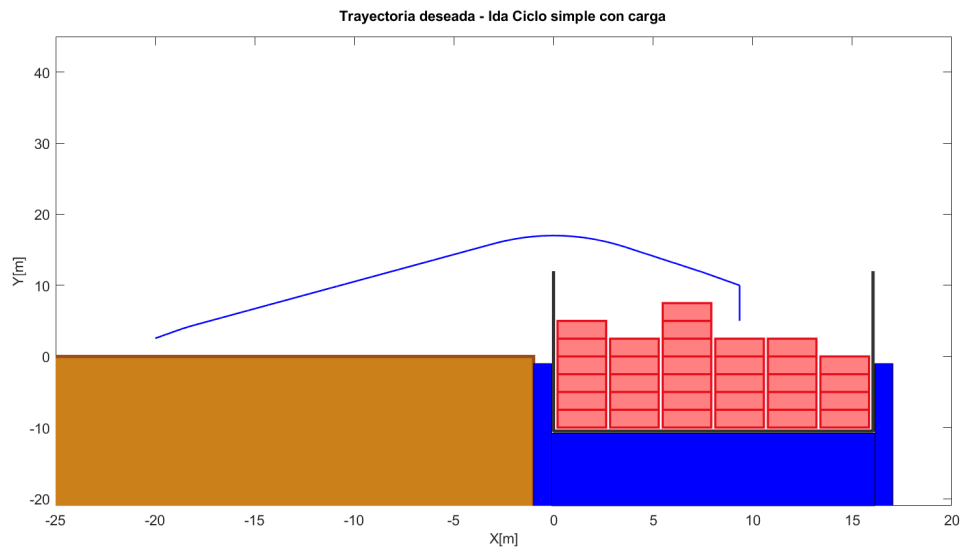


Figura 31: Trayectoria calculada de muelle a barco

3.2.1.2 Trayectoria hacia el muelle

El cálculo es similar al anterior, pero teniendo en cuenta las nuevas posiciones de inicio y fin, como se muestra en la Fig. 32.

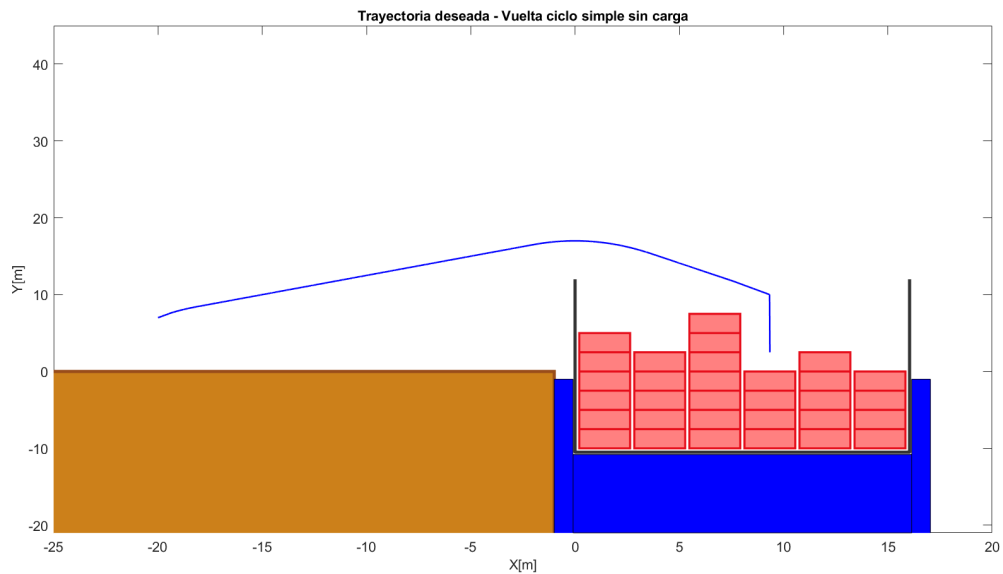


Figura 32: Trayectoria calculada del barco al muelle

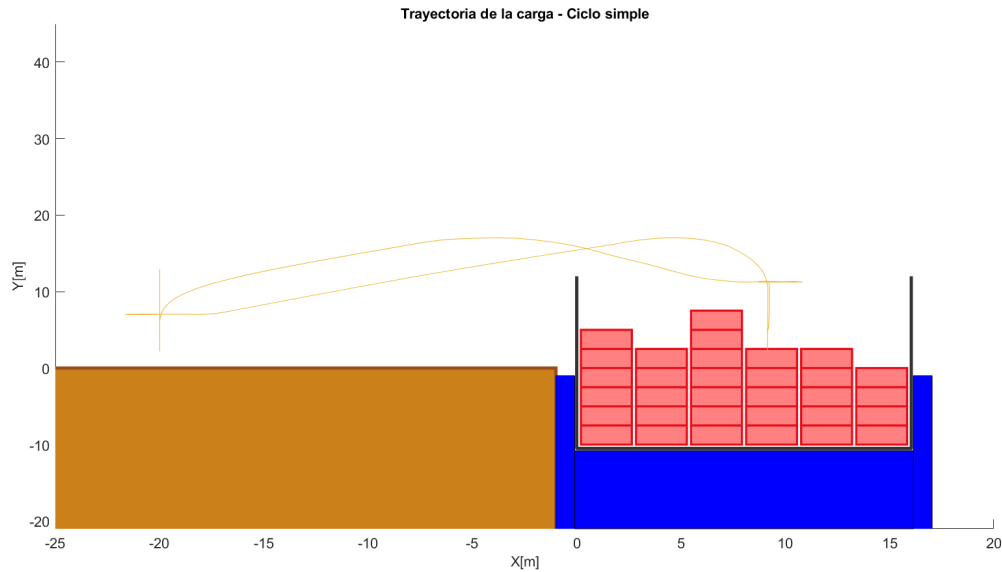


Figura 33: Trayectoria real de la carga

Las trayectorias anteriores se usan como consigna, pero la real se ve afectada por los controladores del sistema, resultando en la presentada en la Fig. 33. Además, antes de realizar la última parte de cualquier descenso, se espera a que la oscilación de la carga se reduzca a un valor determinado para evitar colisiones.

3.2.1.3 Ciclo doble

Con el mismo método de cálculo, se implementa un ciclo doble, es decir una trayectoria en la que se lleva un contenedor del muelle al barco, se lo descarga y se toma un nuevo contenedor de los que están sobre el barco para llevar al muelle. El cálculo de los movimientos de ida hacia el barco y hacia el muelle es igual que en el ciclo simple, pero se agrega, de forma intermedia, una trayectoria de contenedor a contenedor con el mismo procedimiento de cálculo para evitar una colisión con cualquier columna de contenedores que pudiera encontrarse entre ellos. En la Fig. 34 puede verse la trayectoria calculada para un ciclo doble y en la Fig. 35, se observa como queda dicha trayectoria en la simulación luego de ser afectada por los controladores.

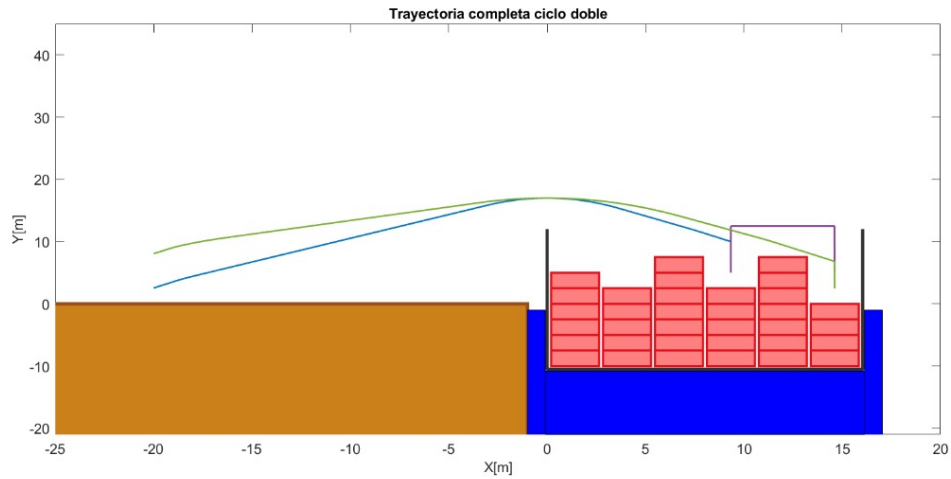


Figura 34: Trayectoria calculada de ciclo doble

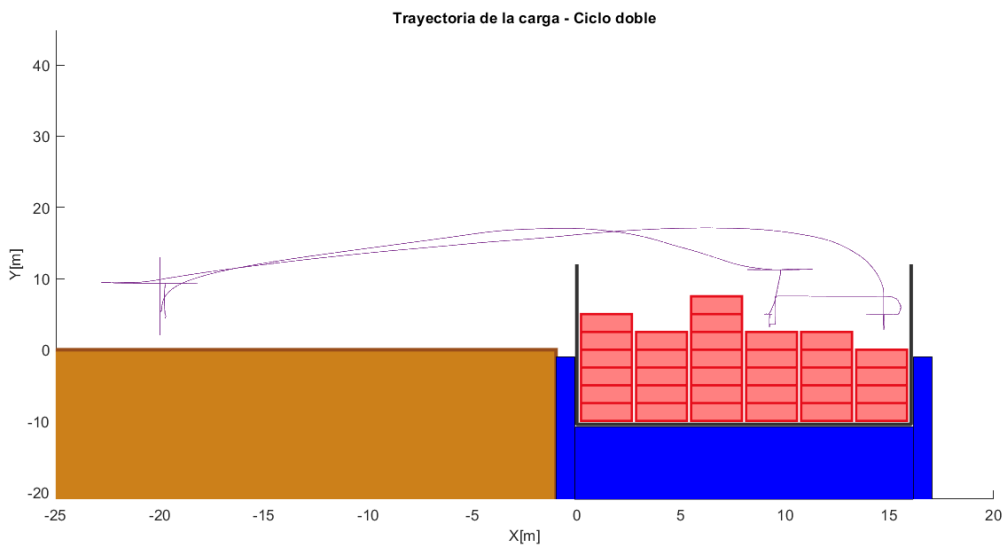


Figura 35: Trayectoria de ciclo doble

3.3. Nivel 0: Seguridad y Protección

En la realidad, la función de seguridad o protección consiste en un autómata más reducido y confiable, separado del autómata del Nivel 1, que debe tomar el control ante falla críticas de niveles inferiores y/o riesgos de seguridad. Por limitaciones de Codesys, este nivel se implementó

como una POU paralela al Nivel 1, que lo inhabilita ante cualquier falla o riesgo, pero siguen estando ambos niveles dentro del mismo dispositivo. Este nivel se puede observar en la Fig. 36

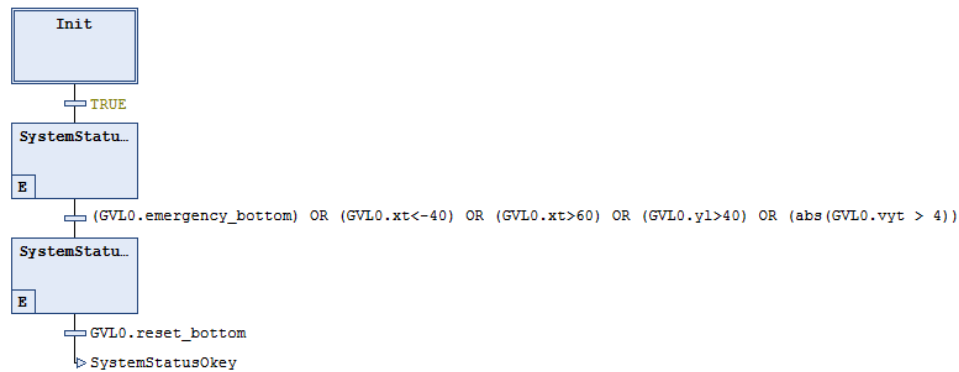


Figura 36: POU del autómata de Nivel 0

3.4. Co-simulación y servidor OPC UA

La migración de los sistemas de nivel 1 y nivel 0 a Codesys, trajo consigo el planteo inicial de una co-simulación. La misma fue lograda con éxito, haciendo uso del protocolo de comunicación OPC (OLE for Process Control) UA (Unified Architecture). OPC es un estándar que define la comunicación de datos entre dispositivos de diferentes fabricantes. En dicho sistema de comunicación, existe un servidor OPC UA al cual se conectan uno o mas clientes OPC UA estableciendo cada uno un socket con TCP/IP. Tal servidor es capaz de almacenar todas las variables del sistema en un caché al cual pueden acceder todos los clientes y escribir o tomar lectura sobre las mismas según los derechos de acceso. En nuestro sistema en particular se cuenta con un PLC simulado por Codesys (cliente), un servidor OPC UA provisto y ejecutado por Codesys y la planta junto con el nivel 2 de control simulado en Simulink (cliente). Este esquema permite la co-simulación, ya que es posible supervisar la planta simulada en otro proceso a través del PLC intercambiando datos por medio de un servidor en común. En Fig. 37 puede observarse el diagrama que ilustra el sistema de comunicación descripto.

El cliente PLC y el servidor OPC UA, son ejecutados juntos con el proceso ejecutable *CODESYS Control Win V3 - x64.exe*. Donde el servidor se encuentra escuchando por defecto en el puerto 4840 y dispondrá en su cache todas las variables seleccionadas en el objeto "Symbol Configuration" del entorno de desarrollo de Codesys.

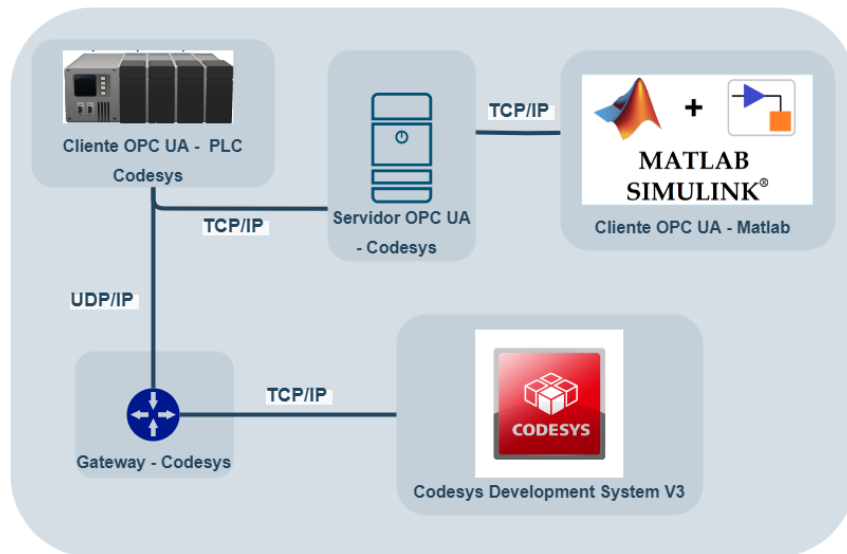


Figura 37: Esquema de comunicaciones de Co-simulación

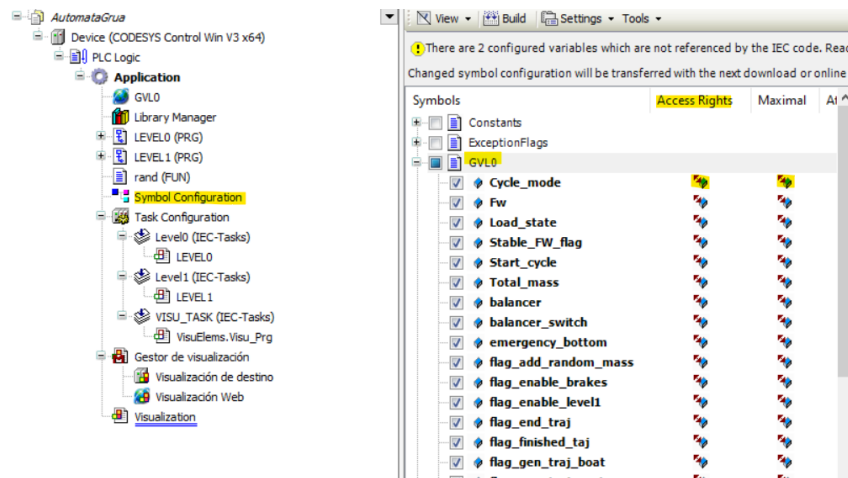


Figura 38: Selección de variables del PLC compartidas por el servidor OPC UA

Por otro lado, desde Matlab, como cliente se desarrollo una función encargada de establecer la conexión con el servidor y leer y escribir las variables del cache según correspondiera. Haciendo analogía con StateFlow, todas las variables que eran entradas de la maquina de estados pasaron a ser variables del servidor escritas desde el cliente con información resultante de Simulink y viceversa con las de salida. Tal función es llamada desde un subsistema en cada paso de tiempo de la simulación. En Fig. 39 se observa este bloque, siendo las señales de la izquierda del bloque de escritura y las de su derecha de lectura.

Una vez establecida la conexión con éxito, es posible supervisar el sistema a través de Codesys e interactuar con la planta a través de un HMI. En Fig. 40 y Fig. 41 se ilustra el HMI con la visualización de dos variables de la planta (cliente Simulink).

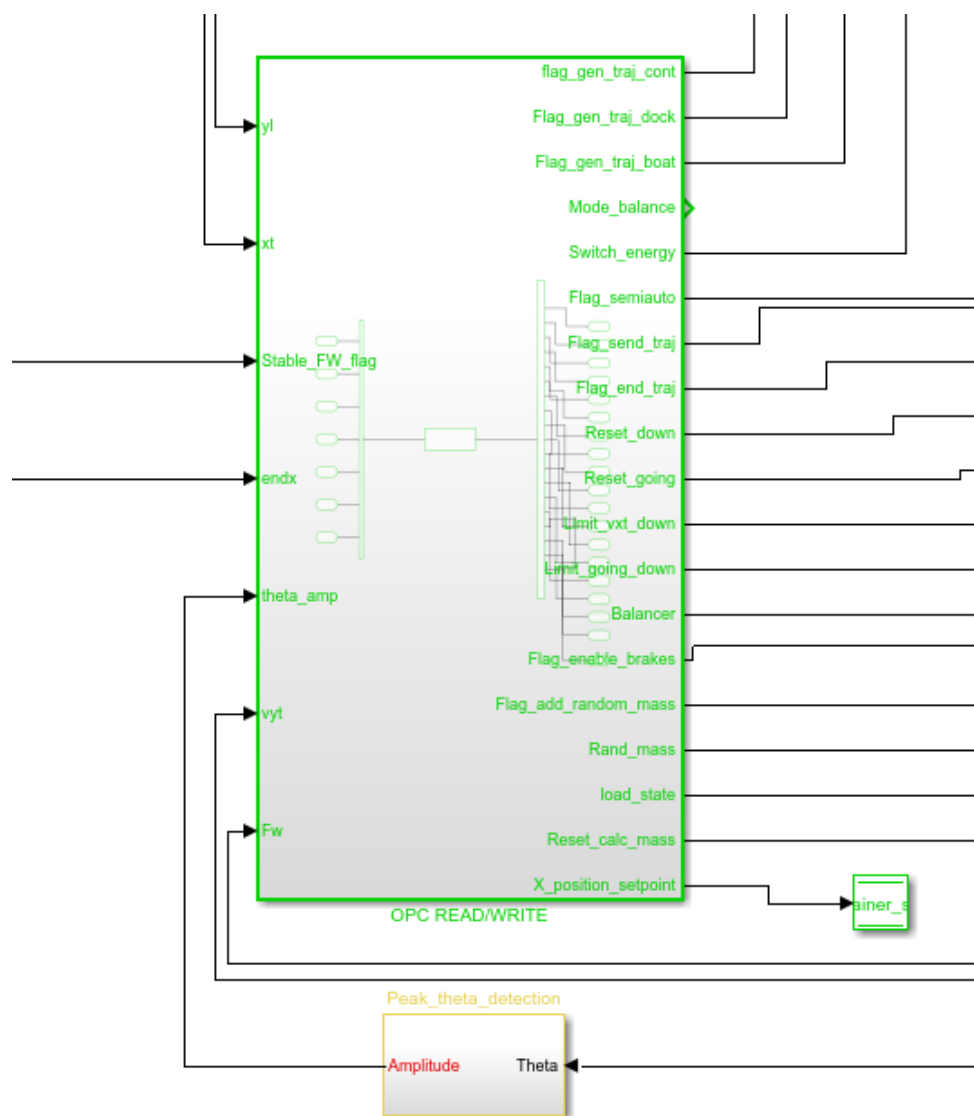


Figura 39: Bloque de escritura y lectura OPC UA

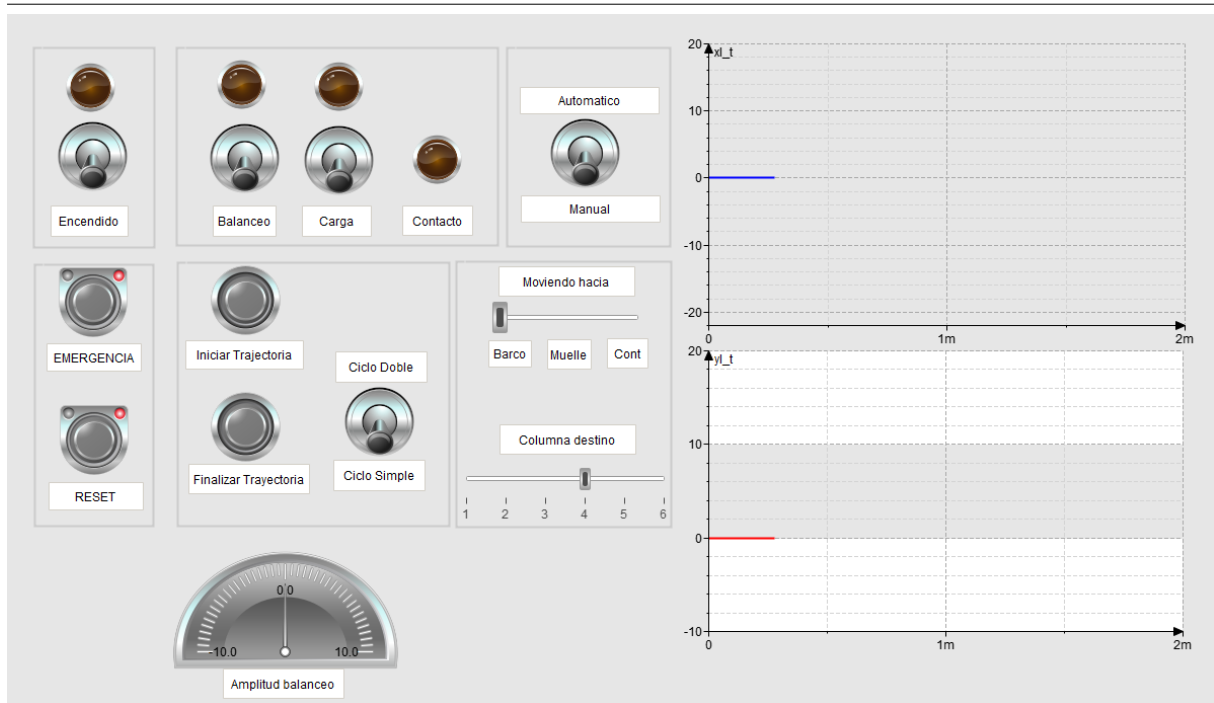


Figura 40: HMI con el sistema apagado

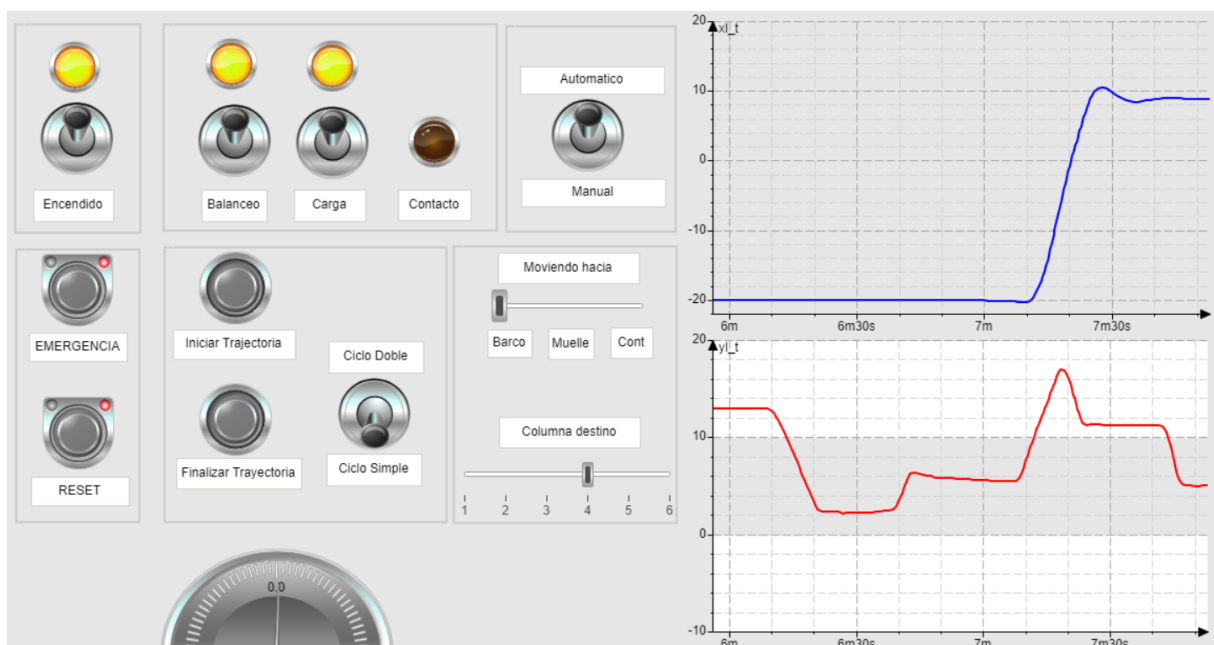


Figura 41: HMI con el sistema encendido

4. Resultados

En la presente sección, se muestran los resultados de dos ensayos; el primero es el correspondiente a la ejecución de un ciclo simple, y el otro a un ciclo doble. Resulta necesario aclarar que la posición del carril del muelle se encuentra en la posición $x = -20[m]$.

4.1. Trayectoria Ciclo Simple

A continuación se muestran los resultados de una carga de contenedor en la columna 4 del barco y retorno sin carga al muelle. El sistema parte de una condición inicial descargado y como se explicó en la Sección 2.4, el operario debe descender, tomar la carga y ascender lentamente para completar el proceso de detección de la masa. Posteriormente se cambia a modo automático donde se obtienen los perfiles de velocidad del muelle a la columna destino. Luego, el sistema pasa a modo manual donde el operario debe realizar la maniobra de descarga. Por último, se vuelve al modo automático y se regresa al muelle finalizando la trayectoria mostrada en la Fig.42. Se pueden ver gráficas de parámetros representativos de este recorrido en las Fig. 42, 43, 44, 45, 46, 47 y 48

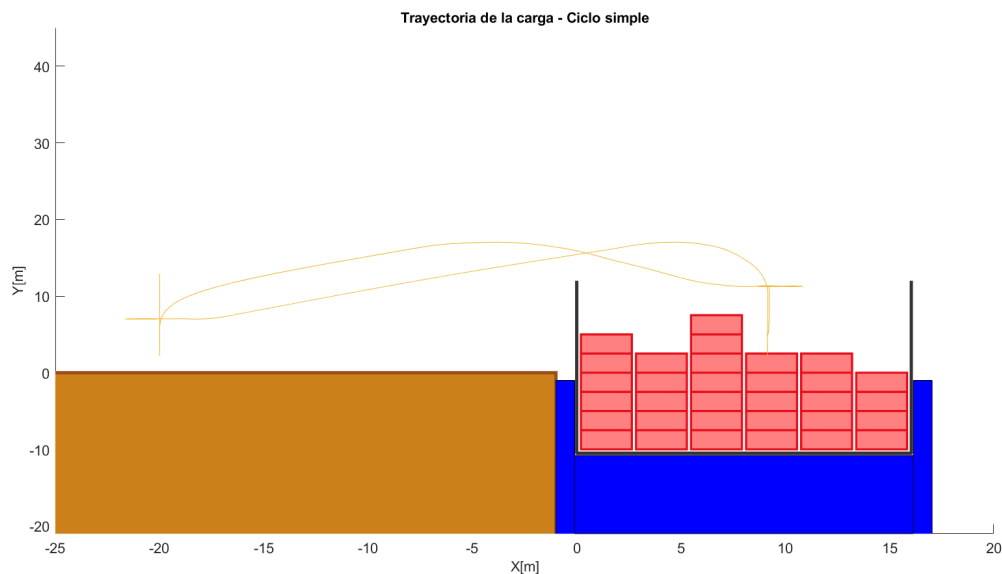


Figura 42

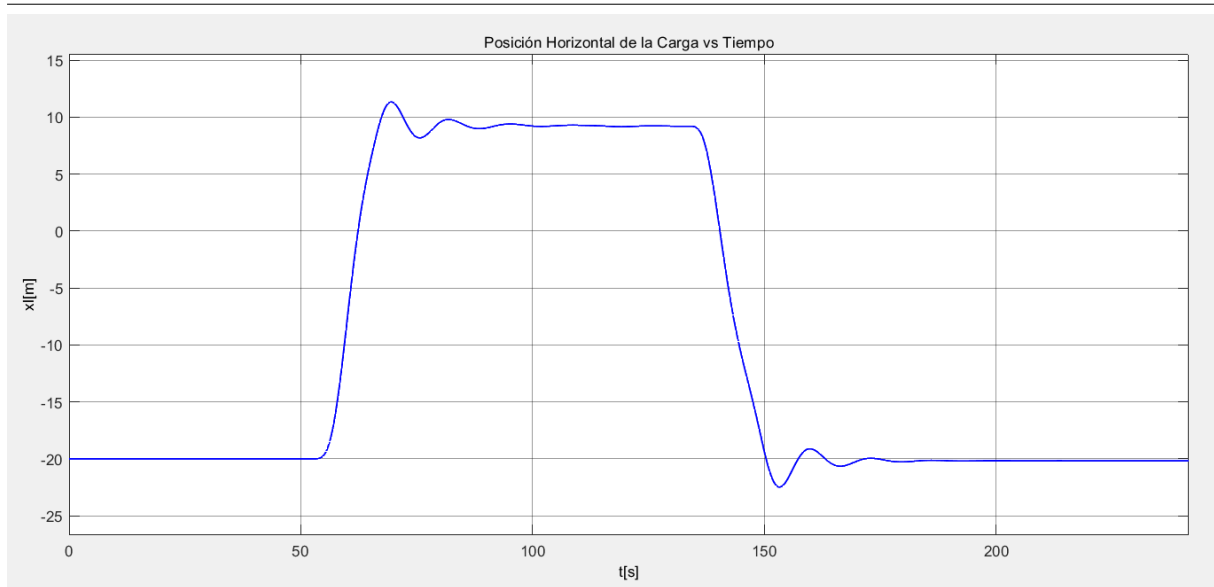


Figura 43

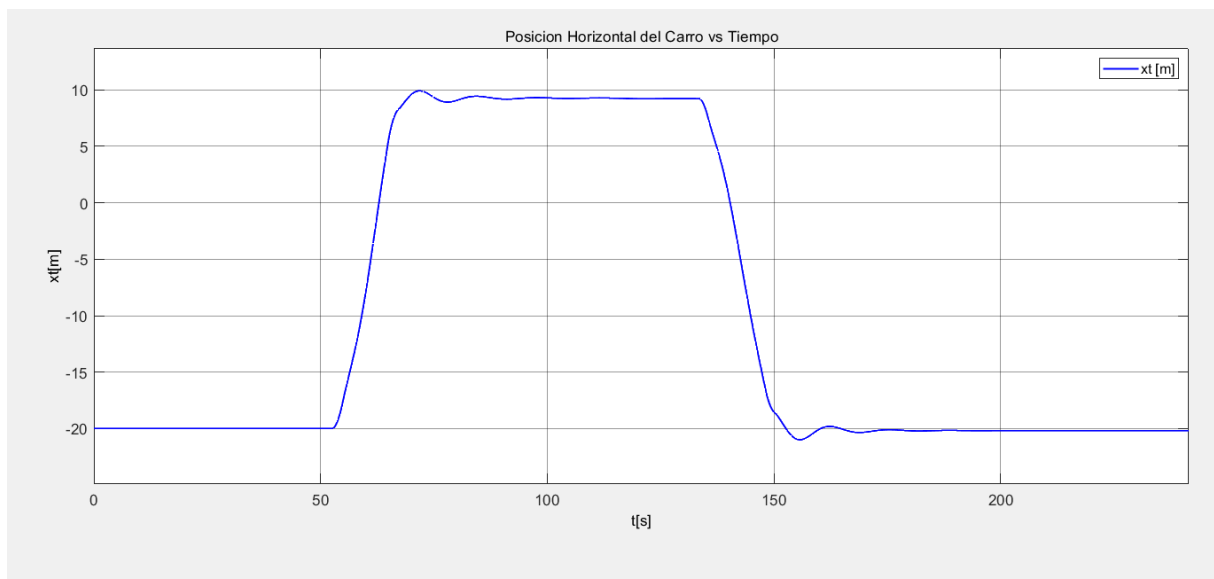


Figura 44

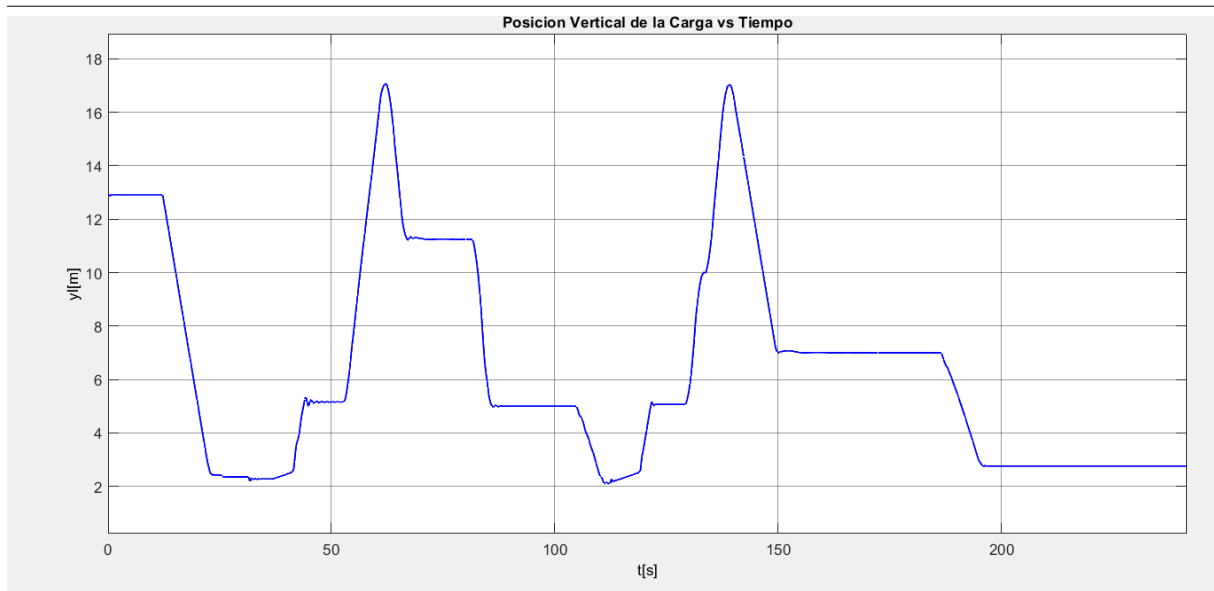


Figura 45

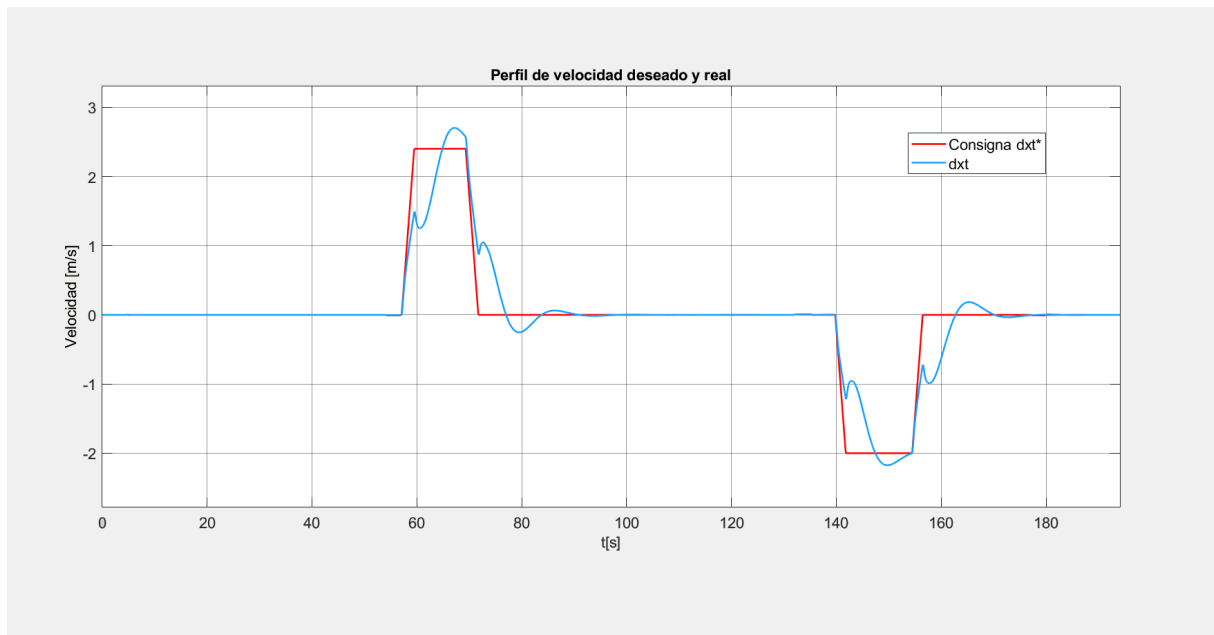


Figura 46

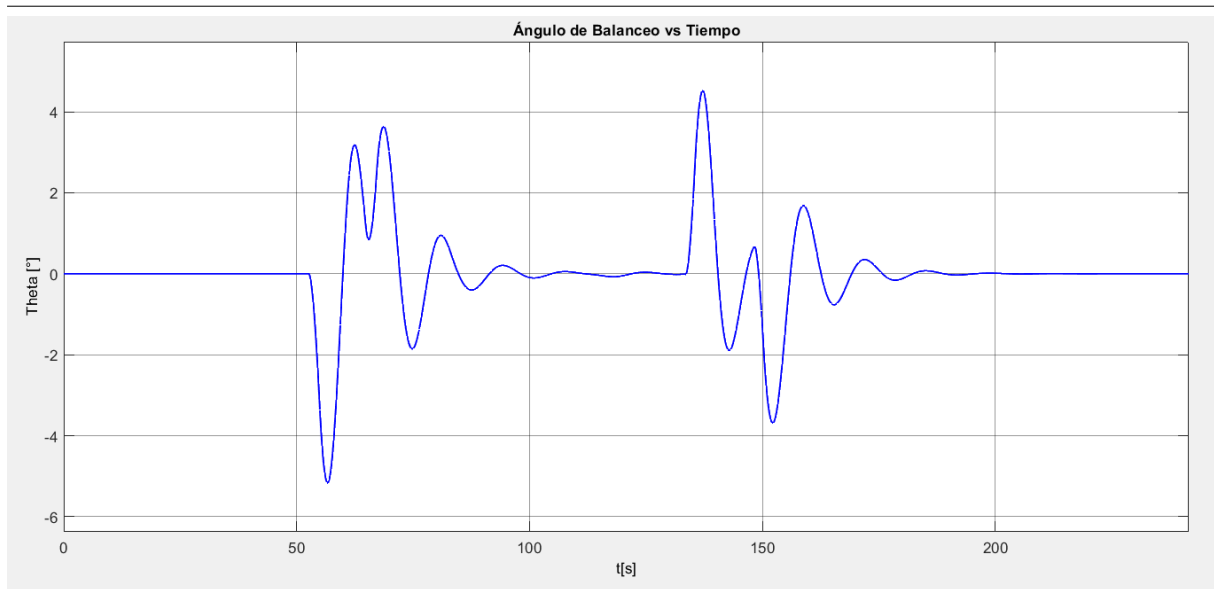


Figura 47

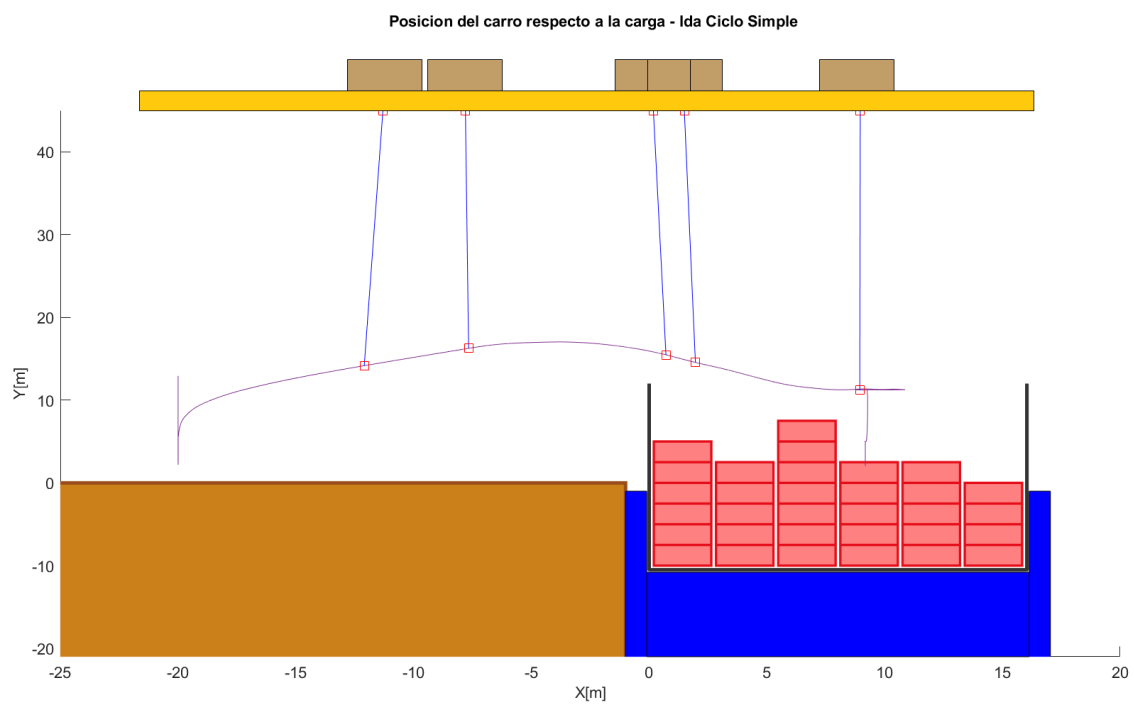


Figura 48

4.2. Trayectoria Ciclo Doble

A continuación se muestran los resultados de un ciclo doble en el que se carga un contenedor en la columna 4 del barco, luego un movimiento sin carga hacia columna 6, se toma un nuevo contenedor y finalmente se retorna al muelle, obteniéndose las Fig. 49, 50, 51, 52, 53 y 54

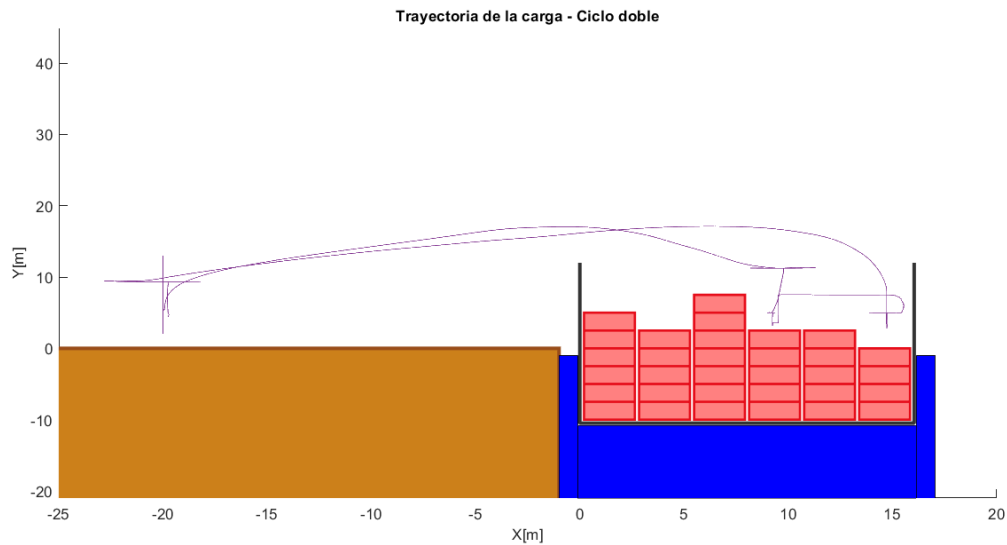


Figura 49

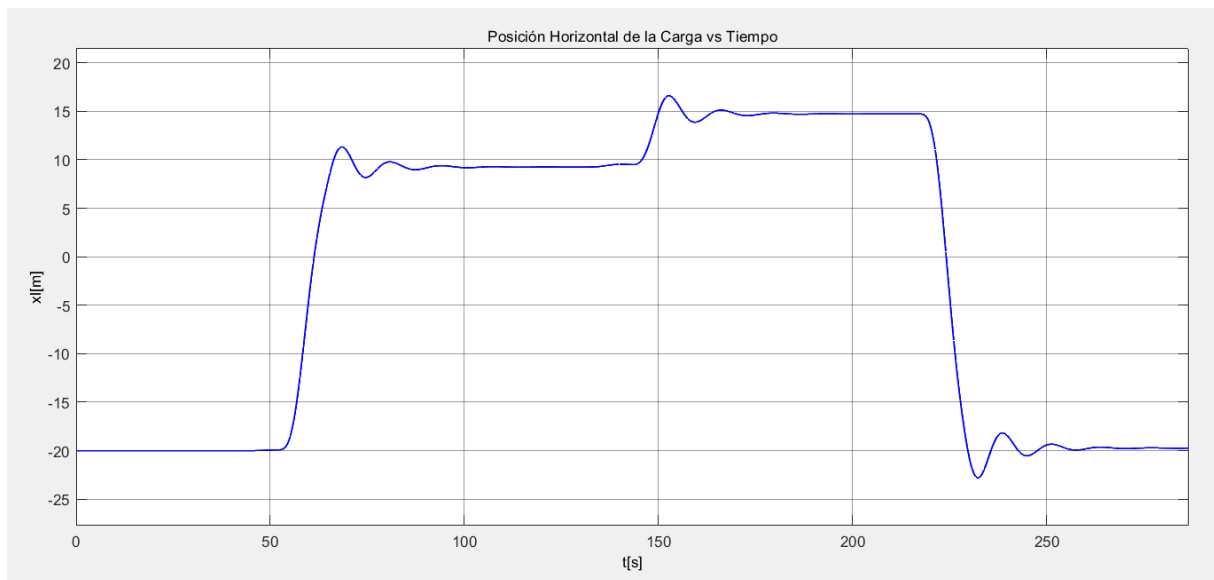


Figura 50

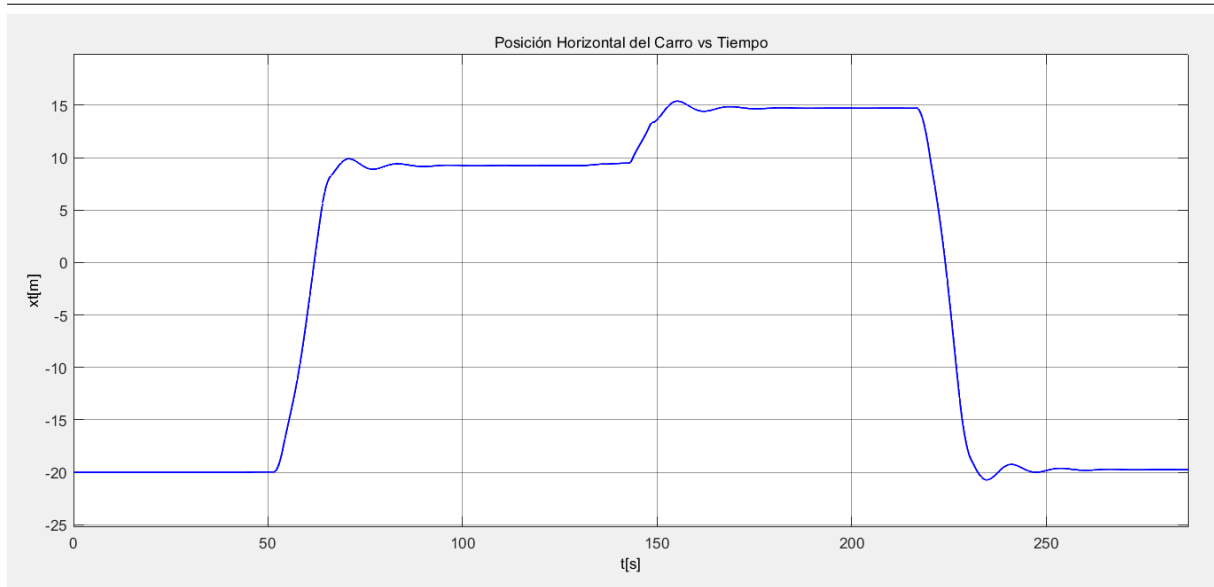


Figura 51

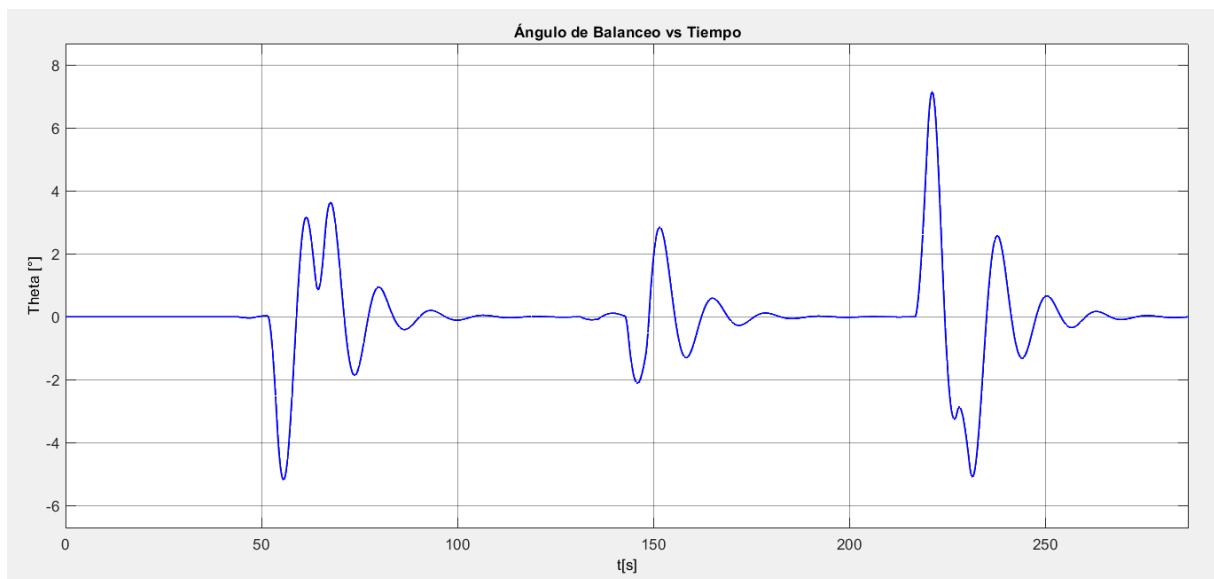


Figura 52

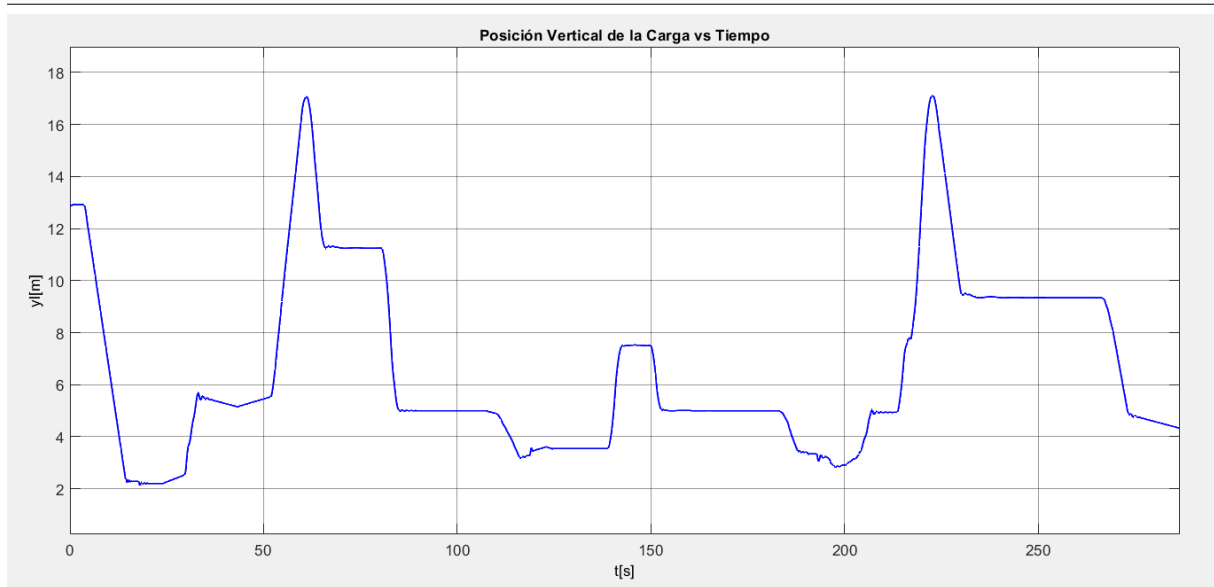


Figura 53

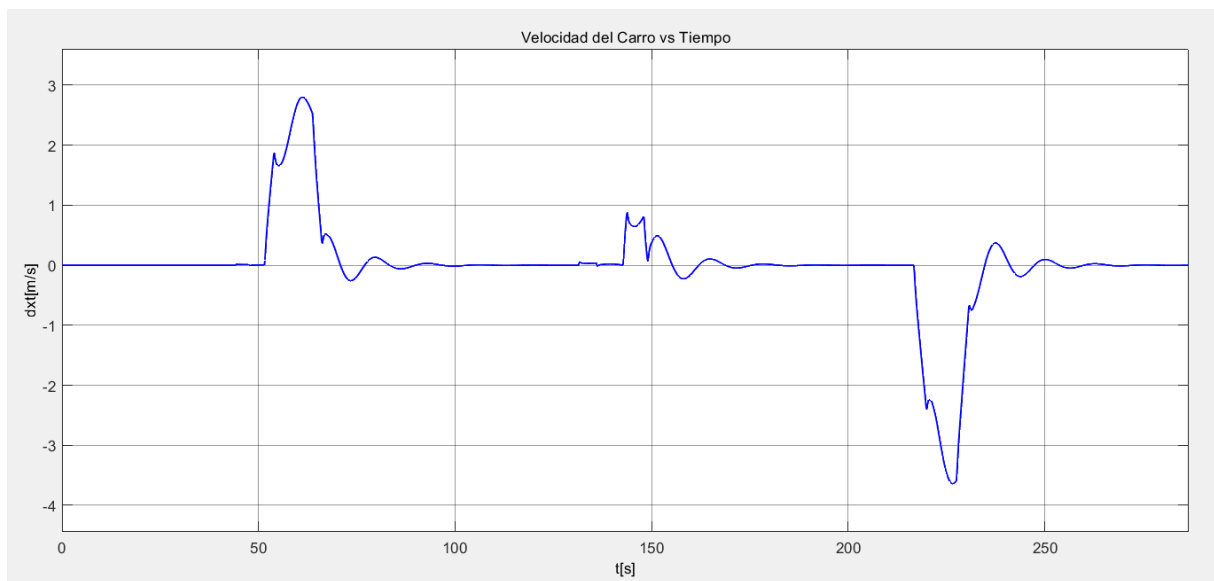


Figura 54

5. Conclusiones

A partir del planteo de una aplicación real simplificada con fines académicos, en el presente proyecto se utilizaron los conocimientos adquiridos en las cátedras de **Automática y Máquinas Eléctricas** y de **Autómatas y Control Discreto** para diseñar e implementar un modelado 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, como se puede ver en las figuras de la Sección 4, donde se reflejan cada uno de los conceptos aplicados en secciones anteriores, logrando trayectorias optimizadas y seguras ante fallas, con un control efectivo de movimientos y posicionamiento final correcto.

Como puntos particulares a destacar se encuentran en control de balanceo, que luego de varios planteos diferentes se alcanzó un resultado muy eficiente; y la comunicación OPC UA que es un protocolo industrial emergente que pueden utilizar los PLCs reales, lo que lo vuelve un logro muy interesante. Esto último, junto con la implementación en Codesys, el cual cumple con el estándar internacional IEC 61131, permiten un primer acercamiento a la programación de autómatas en la industria.

En conclusión, resultó ser un proyecto arduo pero muy interesante, con la aplicación de una gran cantidad de conceptos técnicos, tanto teóricos como prácticos, aprendidos no solo en las cátedras mencionadas, sino también en varios años de estudios de la carrera, lo que incrementa aún más la satisfacción lograda con el éxito de los resultados.

6. Referencias

- [1] Guía de Trabajo - Proyecto Global Integrador AyCD - 2020 - Ing. Gabriel L. Julián
- [2] Material de clase de las cátedras “Autómatas y Control Discreto” y “Automática y Máquinas Eléctricas” - Ing. Gabriel L. Julián
- [3] Ingeniería de Control Moderna - 5^{ta} Edición - Katsuhiko Ogata
- [4] Documentación de MATLAB
- [5] Documentación de CODESYS
- [6] Tutorial Toolbox OPC Matlab (<https://www.halvorsen.blog/documents/technology/opc/>)

7. Anexos

7.1. Función de lectura y escritura OPC UA - Matlab

```
1 function x = Read_OPC_Func(y)
2
3 %Variables
4 persistent init_Server;
5 persistent init_Nodes;
6 persistent ua_Client2;
7
8 %Nodes
9 %Writeable Variables
10 persistent xt;
11 persistent yl;
12 persistent vyt;
13 persistent Stable_FW_flag;
14 persistent Fw;
15 persistent theta_amp;
16 persistent xend;
17
18 %Readable Variables
19 persistent balancer;
20 persistent flag_add_random_mass;
21 persistent rand_mass;
22 persistent flag_semiauto;
23 persistent limit_going_down;
24 persistent limit_vxt_down;
25 persistent mode_balance;
26 persistent reset_calc_mass;
27 persistent flag_end_traj;
28 persistent flag_send_traj;
29 persistent reset_going;
30 persistent reset_down;
31 persistent Load_state;
32 persistent flag_gen_traj_boat;
33 persistent flag_gen_traj_dock;
34 persistent switch_energy;
35 persistent flag_enable_brakes;
36 persistent x_position_setpoint;
37 persistent flag_gen_traj_cont;
38
39
40
41 %initialize variables
42 if isempty(init_Server)
43     init_Server = 0;
44     init_Nodes = 0;
45 end
46
47 if init_Server == 0
48     init_Server = 1;
49     server = opcuaserverinfo('localhost');
```

```

50 ua_Client2 = opcua(server);
51 connect(ua_Client2,'tom','tom');
52 %connect(ua_Client2, 'gabrielq', 'incorrecta1');
53 end
54
55 if ua_Client2.isConnected && init_Nodes == 0
56
57     init_Nodes =1;
58     DB_Node = findNodeByName(ua_Client2.Namespace,'GVL0','-once');
59
60     %READ
61     balancer = findNodeByName(DB_Node,'balancer','-once');
62     flag_add_random_mass = findNodeByName(DB_Node,'flag_add_random_mass','-once'
);
63     rand_mass = findNodeByName(DB_Node,'rand_mass','-once');
64     flag_semiauto = findNodeByName(DB_Node,'flag_semiauto','-once');
65     limit_going_down = findNodeByName(DB_Node,'limit_going_down','-once');
66     limit_vxt_down = findNodeByName(DB_Node,'limit_vxt_down','-once');
67     mode_balance = findNodeByName(DB_Node,'mode_balance','-once');
68     reset_calc_mass = findNodeByName(DB_Node,'reset_calc_mass','-once');
69     flag_end_traj = findNodeByName(DB_Node,'flag_end_traj','-once');
70     flag_send_traj = findNodeByName(DB_Node,'flag_send_traj','-once');
71     reset_going = findNodeByName(DB_Node,'reset_going','-once');
72     reset_down = findNodeByName(DB_Node,'reset_down','-once');
73     Load_state = findNodeByName(DB_Node,'Load_state','-once');
74     flag_gen_traj_boat = findNodeByName(DB_Node,'flag_gen_traj_boat','-once');
75     flag_gen_traj_dock = findNodeByName(DB_Node,'flag_gen_traj_dock','-once');
76     switch_energy = findNodeByName(DB_Node,'switch_energy','-once');
77     flag_enable_brakes = findNodeByName(DB_Node,'flag_enable_brakes','-once');
78     x_position_setpoint = findNodeByName(DB_Node,'x_position_setpoint','-once');
79     flag_gen_traj_cont = findNodeByName(DB_Node,'flag_gen_traj_cont','-once');
80
81
82     %WRITE
83     xt = findNodeByName(DB_Node,'xt','-once');
84     yl = findNodeByName(DB_Node,'yl','-once');
85     vyt = findNodeByName(DB_Node,'vyt','-once');
86     Fw = findNodeByName(DB_Node,'Fw','-once');
87     theta_amp = findNodeByName(DB_Node,'theta_amp','-once');
88     Stable_FW_flag = findNodeByName(DB_Node,'Stable_FW_flag','-once');
89     xend = findNodeByName(DB_Node,'xend','-once');
90 end
91
92 if ua_Client2.isConnected == 1 && init_Nodes==1
93
94     [Balancer,~,~] = readValue(ua_Client2,balancer);
95     [Flag_add_random_mass,~,~] = readValue(ua_Client2,flag_add_random_mass);
96     [Rand_mass,~,~] = readValue(ua_Client2,rand_mass);
97     [Flag_semiauto,~,~] = readValue(ua_Client2,flag_semiauto);
98     [Limit_going_down,~,~] = readValue(ua_Client2,limit_going_down);
99     [Limit_vxt_down,~,~] = readValue(ua_Client2,limit_vxt_down);
100     [Mode_balance,~,~] = readValue(ua_Client2,mode_balance);
101     [Reset_calc_mass,~,~] = readValue(ua_Client2,reset_calc_mass);
102     [Flag_end_traj,~,~] = readValue(ua_Client2,flag_end_traj);
103     [Flag_send_traj,~,~] = readValue(ua_Client2,flag_send_traj);

```

```
104 [Reset_going, ~,~] = readValue(ua_Client2,reset_going);
105 [Reset_down, ~,~] = readValue(ua_Client2,reset_down);
106 [load_state, ~,~] = readValue(ua_Client2,Load_state);
107 [Flag_gen_traj_boat, ~,~] = readValue(ua_Client2,flag_gen_traj_boat);
108 [Flag_gen_traj_dock, ~,~] = readValue(ua_Client2,flag_gen_traj_dock);
109 [Switch_energy, ~,~] = readValue(ua_Client2,switch_energy);
110 [Flag_enable_brakes, ~,~] = readValue(ua_Client2,flag_enable_brakes);
111 [X_position_setpoint, ~,~] = readValue(ua_Client2,x_position_setpoint);
112 [Flag_gen_traj_cont, ~,~] = readValue(ua_Client2,flag_gen_traj_cont);
113
114
115 writeValue(ua_Client2,xt,y(1));
116 writeValue(ua_Client2,yl,y(2));
117 writeValue(ua_Client2,vyt,y(3));
118 writeValue(ua_Client2,Fw,y(4));
119 writeValue(ua_Client2,theta_amp,y(5));
120 writeValue(ua_Client2,Stable_FW_flag,y(6));
121 writeValue(ua_Client2,xend,y(7));
122 end
123 x = double([Balancer Flag_add_random_mass Rand_mass Flag_semiauto
Limit_going_down Limit_vxt_down Mode_balance Reset_calc_mass Flag_end_traj
Flag_send_traj Reset_going Reset_down load_state Flag_gen_traj_boat
Flag_gen_traj_dock Switch_energy Flag_enable_brakes X_position_setpoint
Flag_gen_traj_cont]);
124
125 end
```