Facultad de Ingeniería - UNCuyo

Control Semi-Automático de Grúa Portuaria de Muelle tipo Pórtico

Autómatas y Control Discreto

Maximiliano Garcia – Jonathan Obredor

1-1-2019

Índice

Resumen

En el presente trabajo se desarrolla un autómata híbrido de control para operación semiautomática coordinada de una grúa portacontenedores portuaria de muelle tipo pórtico. Se busca verificar el buen desempeño del sistema ante las distintas situaciones de funcionamiento.[1]

Para esto se plantea el diseño, implementación de modelo conceptual y análisis de desempeño mediante simulación “Model-in-the- Loop” usando Simulink/Stateflow. Luego se implementa en un entorno de programación de autómatas industriales bajo norma IEC 61131 usando CODESYS y se simula “Software-in-the-Loop”. Finalmente se presentan los resultados.

Introducción

El sistema físico a controlar considera dos movimientos principales continuos en un plano vertical (2D): traslación del carro (horizontal) e izaje de la carga (vertical), impulsados mediante motores eléctricos. Ambos movimientos están acoplados entre sí por la carga (contenedor y gancho) que se balancea, suspendida del carro.

El autómata híbrido de control debe considerar dos niveles de control (más un nivel 0 de seguridad). Se dice híbrido porque admite entradas y salidas que son digitales y/o analógicas.

0. Autómata de seguridad.
1. Control supervisor global.
2. Controladores de movimiento.

Los autómatas y la planta se modelaron en Matlab y posteriormente se programaron y simularon en Codesys utilizando diagramas de transición de estados (SFC) para representar los estados y transiciones de los distintos autómatas (nivel 0 y 1), y texto estructurado (ST) para todo lo demás.

Modelo matemático simplificado de la planta

Carga suspendida:

$$m_l \cdot \ddot{x}_l(t) = -F_{w(t)} \cdot \sin\theta\,(t)$$

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

Carro y accionamiento traslación:

$$M_t \cdot \ddot{x}_t(t) = F_t(t) - b_t \cdot \dot{x}_t(t) + F_w(t) \cdot \sin\theta(t)$$

Accionamiento izaje:

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

Cable elástico amortiguado:

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

Restricción geométrica:

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

$$y_{l(t)} = y_{t0} - l(t) \cdot \cos\theta(t)$$

Equivalente:

$$l(t) = \sqrt{\big(x_l(t) - x_t(t)\big)^2 + \big(y_{t0} - y_l(t)\big)^2}$$

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

Senos y cosenos:

$$\sin\theta(t) = \frac{x_l(t) - x_t(t)}{l(t)}$$

$$\cos\theta(t) = \frac{y_{t0} - y_{l(t)}}{l(t)}$$

Reemplazando para la carga suspendida obtenemos:

$$m_l \cdot \ddot{x}_l(t) = -F_{w(t)} \cdot \frac{x_l(t) - x_t(t)}{l(t)}$$

$$m_l \cdot \ddot{y}_l(t) = F_w(t) \cdot \frac{y_{t0} - y_{l(t)}}{l(t)} - m_l \cdot g$$

Reemplazando obtenemos para el carro:

$$M_t \cdot \ddot{x}_t(t) = F_t(t) - b_t \cdot \dot{x}_t(t) - m_l \cdot \ddot{x}_l(t)$$

Modelo matemático del carro

$$J_m \cdot \dot{\omega}_m = T_m - b_m \cdot \omega_m - T_l$$

$$\omega_w = \frac{\omega_m}{r} \; ; T_w = r \cdot T_l \quad \text{(b)}$$

Siendo r=15 el factor de reducción de la caja reductora del carro.

Se convierte al eje lento:

$$J_w \cdot \dot{\omega}_w = T_w - b_w \cdot \omega_w - T_c$$

$$v_t = \omega_w \cdot R_w \; ; \; F_c = \frac{T_c}{R_w} \quad \text{(a)}$$

Donde Fc es la fuerza que mueve al carro.

Reemplazando las relaciones de transmisión:

$$J_m \cdot \dot{\omega}_w \cdot r = T_m - b_m \cdot \omega_w \cdot r \; - \frac{T_w}{r}$$

Despejando Tw:

$$T_w = T_m \cdot r - b_m \cdot \omega_w \cdot r^2 \; - J_m \cdot \dot{\omega}_m \cdot r^2$$

Remplazando a y b:

$$J_w \cdot \dot{\omega}_w = T_m \cdot r - b_m \cdot \omega_w \cdot r^2 \; - J_m \cdot \dot{\omega}_m \cdot r^2 - b_w \cdot \omega_w - T_c$$

$$(J_w + J_m \cdot r^2) \cdot \dot{\omega}_w = -(b_m \cdot r^2 + b_w) \cdot \omega_w + T_m \cdot r - T_c$$

$$\frac{(J_w + J_m \cdot r^2)}{R_w} \cdot \dot{v}_t = -(b_m \cdot r^2 + b_w) \cdot \frac{v_t}{R_w} + T_m \cdot r - F_c \cdot R_w$$

Despejando Fc:

$$F_c = -\frac{(J_w + J_m \cdot r^2)}{R_w^2} \cdot \dot{v}_t - (b_m \cdot r^2 + b_w) \cdot \frac{v_t}{R_w^2} + \frac{T_m \cdot r}{R_w} \quad \text{(c)}$$

Teniendo en cuenta la segunda ley de Newton, la ecuación del carro es:

$$m_c \cdot \dot{v}_t = F_c + F_w \cdot \sin\theta$$

Reemplazamos lo anterior en (c):

$$m_c \cdot \dot{v}_t = -\frac{(J_w + J_m \cdot r^2)}{R_w^2} \cdot \dot{v}_t - (b_m \cdot r^2 + b_w) \cdot \frac{v_t}{R_w^2} + \frac{T_m \cdot r}{R_w} + F_w \cdot \sin\theta$$

Reordenando obtenemos:

$$\left(m_c + \frac{(J_w + J_m \cdot r^2)}{R_w^2}\right) \cdot \dot{v}_t = \frac{T_m \cdot r}{R_w} - \left(\frac{(b_m \cdot r^2 + b_w)}{R_w^2}\right) \cdot v_t + F_w \cdot \sin\theta$$

$$M_t = \left(m_c + \frac{(J_w + J_m \cdot r^2)}{R_w^2}\right)$$

$$F_t = \frac{T_m \cdot r}{R_w}$$

$$b_t = \left(\frac{(b_m \cdot r^2 + b_w)}{R_w^2}\right) = \left(\frac{b_{eq}}{R_w^2}\right)$$

Donde el b eq es el coeficiente de fricción visto desde el sistema traslacional, reemplazando volvemos a obtener la ecuación del carro.

$$M_t \cdot \dot{v}_t = F_t - b_t \cdot v_t + F_w \cdot \sin\theta$$

Modelo Matemático del izaje

$$J_m \cdot \dot{\omega}_m = T_m - b_m \cdot \omega_m - T_l$$

$$\omega_d = \frac{\omega_m}{r} \; ; T_d = r \cdot T_l$$

Siendo r=30 el factor de reducción de la caja reductora del izaje. Se convierte al eje lento.

$$J_d \cdot \dot{\omega}_d = T_d - b_d \cdot \omega_d - T_c$$

$$\ddot{l}_h = \omega_d \cdot R_d \; ; \; F_c = -\frac{T_c}{R_d} \; (b)$$

Donde Fc es la fuerza que mueve al izaje en este caso.

Reemplazando las relaciones de transmisión:

$$J_m \cdot \dot{\omega}_d \cdot r = T_m - b_m \cdot \omega_d \cdot r \; - \frac{T_w}{r}$$

Despejando Tw:

$$T_w = T_m \cdot r - b_m \cdot \omega_d \cdot r^2 \; - J_m \cdot \dot{\omega}_m \cdot r^2$$

Remplazando a y b:

$$J_d \cdot \dot{\omega}_d = T_m \cdot r - b_m \cdot \omega_d \cdot r^2 \; - J_m \cdot \dot{\omega}_m \cdot r^2 - b_d \cdot \omega_d - T_c$$

$$(J_d + J_m \cdot r^2) \cdot \dot{\omega}_d = -(b_m \cdot r^2 + b_d) \cdot \omega_d + T_m \cdot r - T_c$$

$$\frac{(J_d + J_m \cdot r^2)}{R_d} \cdot \ddot{l}_h = -(b_m \cdot r^2 + b_d) \cdot \frac{\dot{l}_h}{R_d} + T_m \cdot r + F_c \cdot R_d$$

Despejando Fc:

$$F_c = \frac{(J_d + J_m \cdot r^2)}{R_d^2} \cdot \ddot{l}_h + (b_m \cdot r^2 + b_d) \cdot \frac{\dot{l}_h}{R_d^2} - \frac{T_m \cdot r}{R_d} \ (c)$$

Teniendo en cuenta la segunda ley de Newton, la ecuación del izaje es:

$$m_c \cdot \ddot{l}_h = -F_c + F_w \text{(a)}$$

Reemplazamos lo anterior en (c):

$$m_c \cdot \ddot{l}_h = -\frac{(J_d + J_m \cdot r^2)}{R_d^2} \cdot \ddot{l}_h - (b_m \cdot r^2 + b_d) \cdot \frac{\dot{l}_h}{R_d^2} + \frac{T_m \cdot r}{R_d} + F_w$$

Reordenando obtenemos:

$$\left(m_c + \frac{(J_d + J_m \cdot r^2)}{R_d^2}\right) \cdot \ddot{l}_h = \frac{T_m \cdot r}{R_d} - \left(\frac{(b_m \cdot r^2 + b_d)}{R_d^2}\right) \cdot \dot{l}_h + F_w$$

Considerando m_c la masa del cable $= 0$ tenemos:

$$M_h = \frac{(J_d + J_m \cdot r^2)}{R_d^2}$$

$$F_h = -\frac{T_m \cdot r}{R_d}$$

$$b_h = \left(\frac{(b_m \cdot r^2 + b_d)}{R_d^2}\right) = \frac{b_{eq}}{R_d^2}$$

Donde el b eq es el coeficiente de fricción visto desde el sistema traslacional, reemplazando volvemos a obtener la ecuación del izaje.

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

Es de mayor interés considerar la velocidad del izaje Vt, reemplazando con $\dot{l}_h = V_h$ obtenemos:

$$M_h \cdot \dot{V}_h(t) = -F_h(t) - b_h \cdot V_h(t) + F_w(t)$$

Modelo matemático del Balanceo de carga

Partimos desde la hipótesis de que nuestro sistema es subactuado, es decir, tiene más grados de libertad que actuadores.

Se considera como un robot articulado de 3 grados de libertad, donde las articulaciones del carro e izaje son traslacionales, mientras que el balanceo de la carga es rotacional. De estas, se pueden controlar la traslación del carro y el ascenso del izaje, pero no se cuenta con un actuador para el

balanceo, por lo que los Controladores del carro y del balanceo se "disputaran" el mismo actuador. Para lograr este efecto se sumarán las salidas de los mismos previamente ponderadas, y así permitir un comportamiento coherente con las especificaciones de ángulos tolerables.

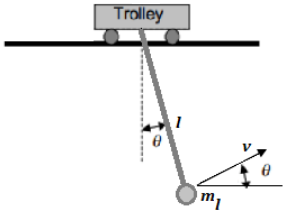


Figura 1: Diagrama del carro y carga suspendida

Se va a encontrar la función de transferencia entre la traslación del carro y el ángulo de balanceo del contenedor suspendido. Se consideran coordenadas generalizadas del sistema x, θ.

Para poder diseñar el controlador de ángulo debemos conocer la función de transferencia entre el ángulo de balanceo y la velocidad del carro [2], [3].

Consideramos la velocidad de la carga suspendida como:

$$\vec{v}_l = \vec{v}_\theta + \vec{v}_t$$

$$v_l = \left(\dot{\theta} * l * \cos(\theta)\,\check{\imath} + \dot{\theta} * l * \sin(\theta)\,\check{\jmath}\right) + (\dot{x}_t\check{\imath})$$

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

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

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

Consideramos la energía cinética total del sistema como la suma de la E. del carro y la E. de la carga suspendida:

$$K = K_t + K_l$$

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

Por otro lado, la energía potencial total del sistema es:

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

Finalmente, el lagrangiano del sistema es:

$$\mathcal{L} = K - U = \frac{1}{2} * m_c * \dot{x}_t^2 + \frac{1}{2} * m_l * \left(\dot{x}_t^2 + 2 * \dot{x}_t * \dot{\theta} * l * \cos(\theta) + \dot{\theta}^2 * l^2\right) - m_l * g * l * (1 - \cos(\theta))$$

Ahora, considerando la ecuación de **Lagrange**[4] para el carro y el izaje:

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

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

Carro

Siendo en la primera ecuación de Lagrange:

$$\frac{\partial \mathcal{L}}{\partial x_t} = 0$$

$$\frac{\partial \mathcal{L}}{\partial \dot{x}_t} = \dot{x}_t(m_c + m_l) + m_l * \dot{\theta} * l * \cos(\theta)$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{x}_t}\right) = \ddot{x}_t * (m_c + m_l) + m_l * l * \left(\ddot{\theta} * \cos(\theta) - \dot{\theta}^2 * \sin(\theta)\right)$$

Reemplazando entonces:

$$(m_c + m_l)\,\ddot{x}_t + m_l * l * \left(\ddot{\theta} * \cos(\theta) - \dot{\theta}^2 * \sin(\theta)\right) = F_t - b_t * \dot{x}_t$$

Izaje

Para la segunda ecuación de Lagrange:

$$\frac{\partial \mathcal{L}}{\partial \theta} = -m_l * \dot{x}_t * \dot{\theta} * l * \sin(\theta) - m_l * g * l * \sin(\theta) = -m_l * l * \sin(\theta) * \left(\dot{\theta} * \dot{x}_t + g\right)$$

$$\frac{\partial \mathcal{L}}{\partial \dot{\theta}} = m_l * \left(\dot{x}_t * l * \cos(\theta) + \dot{\theta} * l^2\right)$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}}\right) = m_l * l * \left(\ddot{x}_t * \cos(\theta) - \dot{x}_t * \sin(\theta) * \dot{\theta} + \ddot{\theta} * l\right)$$

Reemplazando entonces:

$$-m_l\, l\left(\sin(\theta) * \left[\dot{\theta}\,\dot{x}_t + g\right] + \ddot{x}_t * \cos(\theta) - \sin(\theta)\;\dot{\theta}\,\dot{x}_t + \ddot{\theta} * l\right) = 0$$

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

Linealización del péndulo

Para simplificar la notación y a la vez obtener la variable que usaremos en el controlador, utilizaremos $v_t = \dot{x}_t$. Por lo tanto, las ecuaciones (no lineales) del sistema carro-péndulo son:

$$\begin{cases} (m_c + m_l)\,\dot{v}_t + m_l * l * \left(\ddot{\theta} * \cos(\theta) - \dot{\theta}^2 * \sin(\theta)\right) = F_t - b_t * v_t \\ \sin(\theta) * g + \dot{v}_t * \cos(\theta) + \ddot{\theta} * l = 0 \end{cases}$$

Considerando ángulos pequeños podemos expresar: $sin(\theta) = \theta\ ;\ cos(\theta) = 1;\ \theta\dot{\theta}^2 = 0$

Por lo que el sistema lineal resulta:

$$\begin{cases} (m_c + m_l)\ + m_l\, l\, \ddot{\theta} = F_t - b_t\, v_t & (1) \\ g\,\theta + \dot{v}_t + l\,\ddot{\theta} = 0 & (2) \end{cases}$$

Recordando la ecuación de movimiento del carro (sin considerar la perturbación, y además el cable como rígido) tenemos:

$$M_t \cdot \dot{v}_t = F_t - b_t \cdot v_t + 0 \Longrightarrow \dot{v}_t = \frac{F_t - b_t * v_t}{M_t}$$

Reemplazando $\dot{v}_t$ en (2) obtenemos:

$$\theta * g + \frac{F_t - b_t * v_t}{M_t} + \ddot{\theta} * l = 0$$

Despejando:

$$F_t = -M_t * \left(\ddot{\theta} * l + \theta * g\right) + b_t * v_t$$

Reemplazando F_t en (1):

$$(m_c + m_l)\, \dot{v}_t + m_l * l * \ddot{\theta} = -M_t * \left(\ddot{\theta} * l + \theta * g\right)$$

Polos del sistema

Para el carro

Partiendo del modelo matemático del carro se tiene:

$$M_t \cdot \dot{v}_t = F_t - b_t \cdot v_t + F_w \cdot \sin\theta$$

Se considera a Fw como una perturbación por lo que para el análisis de polos no se tendrá en cuenta. Aplicando la transformada de Laplace y considerando condiciones iniciales nulas:

$$M_t\, s\, V_t(s) = F_t(s) - b_t\, V_t(s)$$

$$(M_t\, s + b_t)\; V_t(s) = F_t(s)$$

$$G_t(s) = \frac{V_t(s)}{F_t(s)} = \frac{1}{(M_t\, s + b_t)}$$

Por lo tanto, el polo de este sistema es: $\omega_t = -\frac{b_t}{M_t} = -0.4576$

Para el izaje

Partiendo del modelo matemático del izaje se tiene:

$$M_h\, \dot{V}_h(t) = -F_h(t) - b_h\, V_h(t) + F_w(t)$$

Se considera a Fw como una perturbación

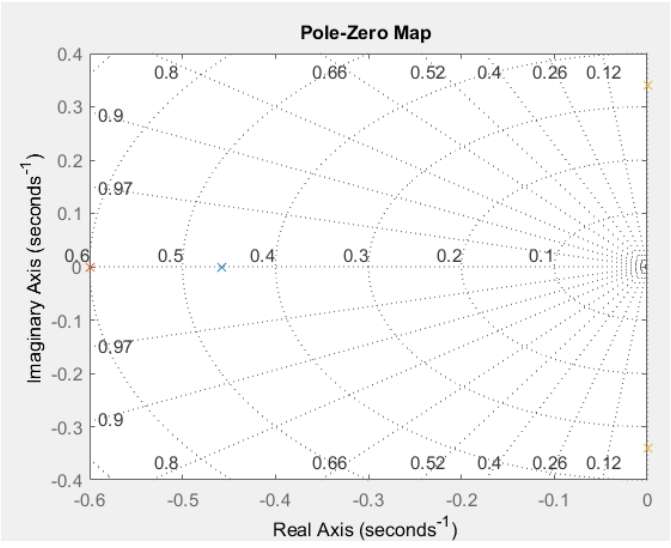


Figura 2 polos del carro, izaje y control de balanceo

por lo que para el análisis de polos no se tendrá en cuenta. Aplicando la transformada de Laplace y considerando condiciones iniciales nulas:

$$M_h\, s\, V_h(s) = F_h(s) - \frac{b_h}{M_h} V_h(s)$$

$$(M_h\, s + b_h)\, V_h(s) = F_h(s)$$

$$G_h(s) = \frac{V_h(s)}{F_h(s)} = \frac{1}{(M_h\, s + b_h)}$$

Por lo tanto, el polo de este sistema es: $\omega_h = -\frac{b_h}{M_h} = -0.5998$

Para el control de balanceo

Partiendo de la ecuación hallada en el modelo matemático:

$$(m_c + m_l)\, \dot{v}_t + m_l * l * \ddot{\theta} = -M_t * \left(\ddot{\theta} * l + \theta * g\right)$$

Se aplica transformada de Laplace (con condiciones iniciales nulas):

$$s * V(s) * (m_c + m_l) + m_l * l * s^2 * \theta(s) = -M_t * s^2 * \theta(s) * l - M_t * \theta(s) * g$$

$$s * V(s) * (m_c + m_l) = -M_t * s^2 * \theta(s) * l - M_t * \theta(s) * g - m_l * l * s^2 * \theta(s)$$

$$V(s) * s * (m_c + m_l) = (-M_t * s^2 * l - M_t * g - m_l * l * s^2) * \theta(s)$$

$$G(s) = \frac{\theta(s)}{V(s)} = \frac{-(m_c + m_l) * s}{(m_l + M_t) * l * s^2 + M_t * g}$$

Esta es la función de transferencia que tiene como entrada la velocidad del carro y como salida el ángulo de balanceo. Corresponden un par de polos:

$$\omega_a = \pm \frac{\sqrt{-4 * (m_l + M_t) * M_t * g * l}}{2 * (m_l + M_t) * l} = 0.0000 \pm 0.3416i$$

Determinación de torques máximos

Se calculan los torques máximos para la saturación de las consignas y dimensionamiento de los motores. Se supone que cada sistema está desacoplado, es decir, el carro es independiente del izaje y viceversa.

Para el carro

Considerando $\dot{v}_{maxt} = 1m/s^2$ y $v_{maxt} = 4m/s$. Se tiene como hipótesis la masa del contenedor sobre el carro sin la presencia del sistema de izaje ni del cable como péndulo:

$$F_{tmax=}\left(M_t + +m_{lfull}\right) \cdot \dot{v}_{maxt} + b_t \cdot v_{maxt}$$

$$T_{mmax} = F_{tmax} \cdot \frac{R_w}{r} \cong 7{,}73 \,.\, 10^3 \; N.m$$

Para el izaje

Considerando $\dot{v}_{maxh} = 1m/s^2$ y $v_{maxh} = 3m/s$. Se tiene como hipótesis el cable rígido y sin movimiento del carro:

$$F_{hmax=} - \left(M_h + m_{lfull}\right) \cdot \dot{v}_{maxh} - b_h \cdot v_{maxh}$$

$$T_{mmax} = -F_{hmax} \cdot \frac{R_d}{r} \cong 4{,}99 \,.\, 10^3 \; N.m$$

Este torque es el que produce la máxima aceleración, sin embargo, el torque real en el motor en el instante de ascenso debe incluir a la gravedad, por lo que queda:

$$T_{mmax} \cong 3{,}26 \,.\, 10^4 \; N.m$$

Condiciones iniciales de la planta

Se eligen los siguientes estados iniciales para que sean consistentes en la simulación y resulten realistas:

Carro y carga sobre el muelle, sin balanceo:

$$x_{l0} = -30m$$
$$\dot{x}_{l0} = 0m$$
$$\ddot{x}_{l0} = 0m$$
$$x_{t0} = -30m$$
$$\dot{x}_{t0} = 0m$$
$$\ddot{x}_{t0} = 0m$$

Carga completamente izada:

$$y_{t0} = 45m$$
$$\dot{y}_{l0} = 0m$$
$$\ddot{y}_{l0} = 0m$$
$$l_{h0} = 5m$$
$$\dot{l}_{h0} = 0m$$
$$\ddot{l}_{h0} = 0m$$

Estiramiento estático del cable

Se calcula la condición inicial $t = 0s$.

$$F_w(0) = K_w \cdot \left(l(0) - l_h(0)\right) + b_w \cdot \left(\dot{l}(0) - \dot{l}_h(0)\right)$$

$$l(t) = \sqrt{\left(x_l(t) - x_t(t)\right)^2 + \left(y_{t0} - y_l(t)\right)^2}$$

$$l_0 = y_{t0} - y_{l0}$$

$$F_{w0} = K_w \cdot (y_{t0} - y_{l0} - l_{h0}) = K_w \cdot (45m - y_{l0} - 5m)$$

$$F_{w0} = K_w \cdot (40m - y_{l0})$$

Considerando:

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

$$\frac{m_l \cdot \ddot{y}_l(t) + m_l \cdot g}{\cos\theta\,(t)} = F_w(t)$$

$$F_{w0} = m_l \cdot g = K_w \cdot (40m - y_{l0})$$

$$y_{l0} = 40m - \frac{m_l \cdot g}{K_w}$$

$$l_0 = 45m - y_{l0}$$

Controladores de movimiento

Para todos los controladores se usará el método de sintonía serie.

Controlador PID de traslación del carro

Como se vio anteriormente, el polo de este sistema es: $\omega_t = -\frac{b_t}{M_t}$

Se plantea un control PID de torque en el motor del carro de la siguiente forma:

$$T_t(s) = \left(b_a + \frac{k_{sa}}{s} + \frac{k_{sia}}{s^2}\right) E_{v_t}(s) = \left(b_a + \frac{k_{sa}}{s} + \frac{k_{sia}}{s^2}\right) \left[V_t^{deseado}(s) - V_t^{medido}(s)\right]$$

Se eligen los siguientes parámetros de diseño: $\begin{cases} \omega_{post} = 10 * \omega_t \\ n_t = 2\,, para\ \xi = 0{,}5 \end{cases}$

Por lo tanto, las ganancias son: $\begin{cases} b_{at} = M_t * n_t * \omega_{post} \\ K_{sat} = M_t * n_t * \omega_{post}^2 \\ K_{siat} = M_t * \omega_{post}^3 \end{cases}$

Controlador PID de izaje de la carga

Partiendo del modelo matemático del izaje se tiene:

$$M_h\, \dot{V}_h(t) = -F_h(t) - b_h\, V_h(t) + F_w(t)$$

Se considera a Fw como una perturbación por lo que para el análisis de polos no se tendrá en cuenta. Aplicando la transformada de Laplace y considerando condiciones iniciales nulas:

$$M_h\, s\, V_h(s) = F_h(s) - \frac{b_h}{M_h} V_h(s)$$

$$(M_h\, s + b_h)\, V_h(s) = F_h(s)$$

$$G_h(s) = \frac{V_h(s)}{F_h(s)} = \frac{1}{(M_h\, s + b_h)}$$

Por lo tanto, el polo de este sistema es: $\omega_h = -\frac{b_h}{M_h}$

Se eligen los siguientes parámetros de diseño: $\begin{cases} \omega_{posh} = 10 * \omega_h \\ n_h = 2 \end{cases}$

Siendo el controlador $T_h(s) = \left(b_a + \frac{k_{sa}}{s} + \frac{k_{sia}}{s^2}\right) E_{v_h}(s)$

Por lo tanto, las ganancias son: $\begin{cases} b_{ah} = M_h * n_h * \omega_{posh} \\ K_{sah} = M_h * n_h * \omega_{posh}^2 \\ K_{siah} = M_h * \omega_{posh}^3 \end{cases}$

Controlador PD de balanceo de carga

Se necesita amortiguar la oscilación de la carga durante la trayectoria y eliminar el error de posicionamiento final de la misma. Para esto se cuenta con un sensor de ángulo entre la carga y el carro, con respecto a la vertical. Dado que se conoce un modelo preciso de la planta se opta por el control PD con ganancias autoajustadas (gain scheduling) en función de la longitud del cable del izaje, y el peso del contenedor suspendido.

Se eligen los siguientes parámetros de diseño: $\begin{cases} \omega_{posa} = 10 * \omega_a \\ n_a = 3;\ para\ \xi = 1\ (amort.crítico) \end{cases}$

Se considera la masa equivalente como: $M_a = \frac{(m_l + M_t)}{(m_c + m_l)} * l$

Siendo el controlador $T_a(s) = \left(b_a + \frac{k_{saa}}{s}\right) E_a(s)$

Por lo tanto, las ganancias son: $\begin{cases} b_{aa} = M_a * n_a * \omega_{posa} \\ K_{saa} = b_{aa} * \omega_{posa} \end{cases}$

Problema del muestreo en tiempo discreto

Ya que nuestro sistema de control es digital y funciona en tiempo discreto será necesario estudiar la estabilidad. Debe observarse que el comportamiento dinámico del sistema de control en tiempo discreto depende del período de muestreo T [5], es decir, el periodo de muestreo modifica la ubicación de los polos y en consecuencia el comportamiento de respuesta.

Al realizar el muestreo por impulsos, las variables complejas *z* y *s* están relacionadas mediante

$$Z = e^{Ts}$$

Los polos discretos deben estar dentro del circulo unitario para asegurar la estabilidad del sistema.

Se simula la planta a T=0.001s y el controlador a T=0.01s, por lo tanto:

Carro: $\omega_{post} = 10 * (-0{,}4576) \Rightarrow z_t = e^{0{,}01 * -4{,}576} = e^{-0{,}04576} = 0{,}955$

Izaje: $\omega_{posh} = 10 * (-0{,}5998) \Rightarrow z_h = e^{0{,}01 * -5{,}998} = e^{-0{,}05998} = 0{,}941$

Estos polos discretos están cerca del circulo unitario, presentando una respuesta oscilatoria, por lo tanto, se corrige eligiendo:

$$\omega_{post} \leq 1.2 * \omega_t$$

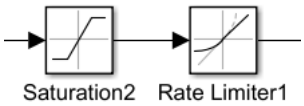
$$\omega_{posh} \leq 1.2 * \omega_h$$

$$\omega_{posa} \leq 1.6 * \omega_a$$

Generación de trayectorias

Los movimientos bruscos (jerk) tienden a causar un mayor desgaste del mecanismo y causan vibraciones al excitar las resonancias en el manipulador. Para garantizar caminos suaves, debemos poner algún tipo de restricciones en las cualidades espaciales y temporales del camino entre los puntos de paso.[6].

Se utilizan dos bloques en la entrada de los controladores de movimiento:



- se saturan las consignas de velocidad, configurando los valores de diseño del proyecto: 4 m/s para el carro, y 3m/s para el izaje vacío (1,5m/s con un contenedor).
- se limitan las consignas de aceleración usando el “rate limiter”, usando $1\frac{m}{s^2}$.

Triángulo de Velocidades

Si bien generar trayectorias usando splines puede ser tan preciso como se desee, se puede lograr un buen rendimiento con el método más sencillo que se presenta a continuación. Justifica esta elección la necesidad de buena performance y velocidad de trabajo, y no tanta precisión espacial, que puede ser corregida por la mano del operario. En casos como la robótica el uso de splines es justificado.

Se sabe que el carro y el izaje tienen velocidades diferentes, por lo que es deseable que ambos lleguen a destino juntos, es decir que el más rápido (carro) acompañe al otro. Para esto se diagrama un triángulo donde el cateto horizontal es la velocidad máxima del carro, y el cateto vertical es la velocidad máxima del izaje.

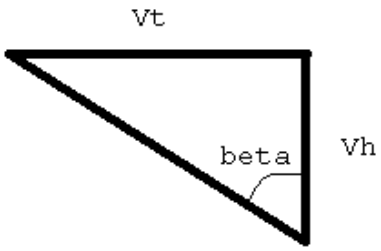


Figura 3: Triángulo de velocidades

Se desea determinar en qué momento del izaje es posible comenzar a desplazar el carro evitando colisiones con los contenedores apilados cerca.

Graficando el triángulo sobre el lugar de partida del contenedor que se desea mover se obtendrá un punto en el izaje donde comenzar a mover el carro de forma segura.

En la gráfica se pueden observar dos triángulos, de aceleración y desaceleración. Los catetos se modifican dependiendo de la carga suspendida.

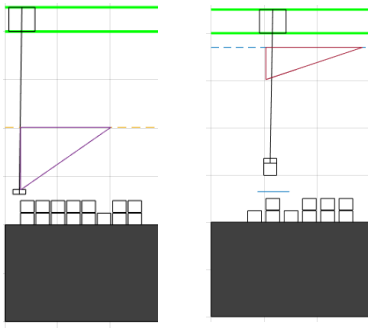


Figura 4: Trayectorias con el izaje vacío y cargado.

Reducción de velocidad lineal hacia los límites

Cuando el carro o el izaje están llegando a sus límites, es necesario reducir su velocidad para llegar a los mismos lentamente y evitar grandes aceleraciones que puedan sacarlos de su operación normal.

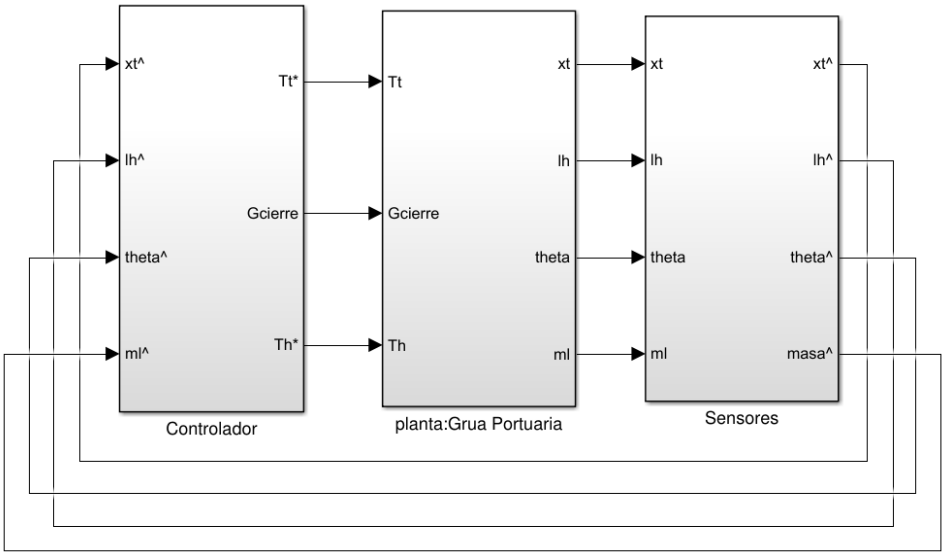
Para esto se plantea una consigna de velocidad que va disminuyendo en forma de línea recta hasta volverse cero en los límites de operación. Esta funcionalidad se empaqueta en un bloque de función.

```
function yRecta = rectaDosPuntos(y2,y1,x2,x1,x)
m=(y2-y1)/(x2-x1);
yRecta=m*(x-x1)+y1;
end
```

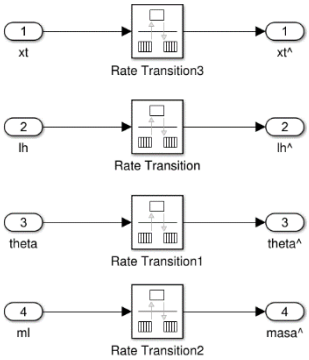
Simulación en Matlab

Se utiliza Matlab 2017a y 2015a.

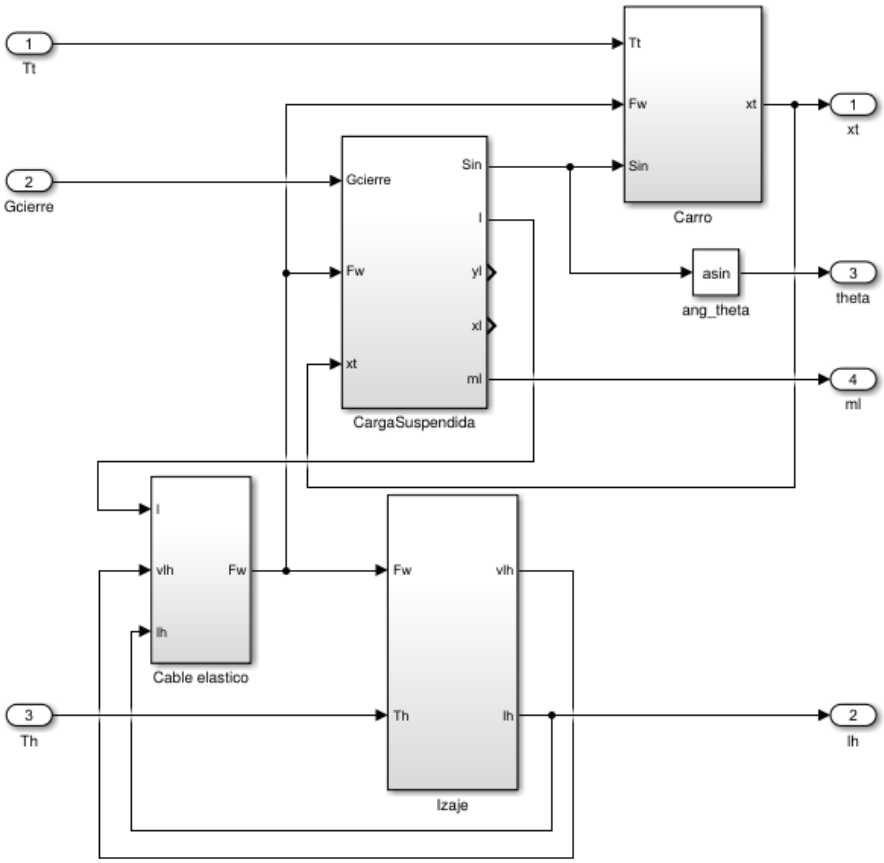
Es importante configurar la condición inicial de los integradores y derivadores. Para simular nuestro sistema en Simulink se plantea el siguiente diagrama de bloques general:



Bloque de sensores

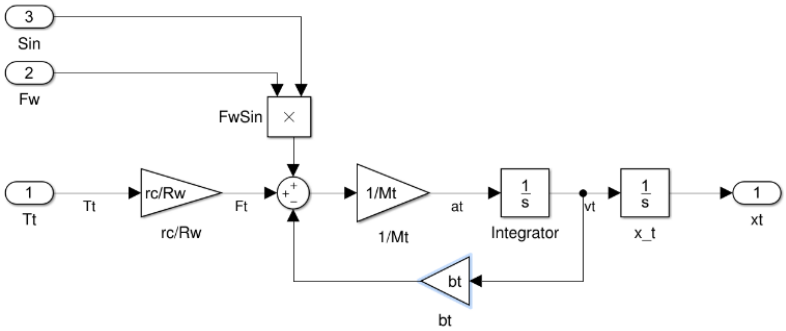


Bloque de la Planta



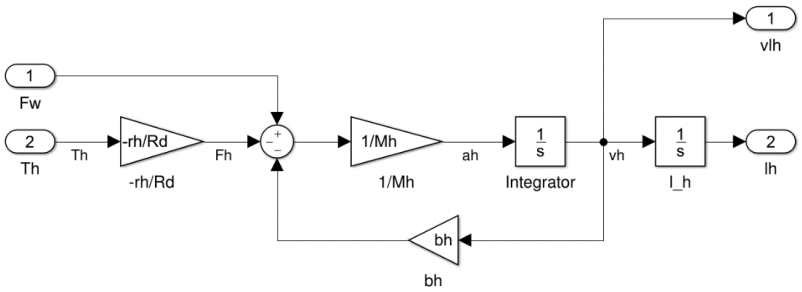
Traslación del carro

Ecuación de movimiento horizontal de traslación del carro.

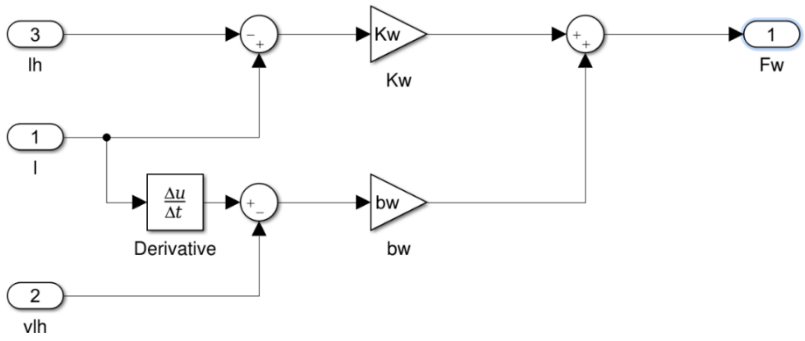


Izaje de la carga

Movimiento vertical de izaje de la carga.

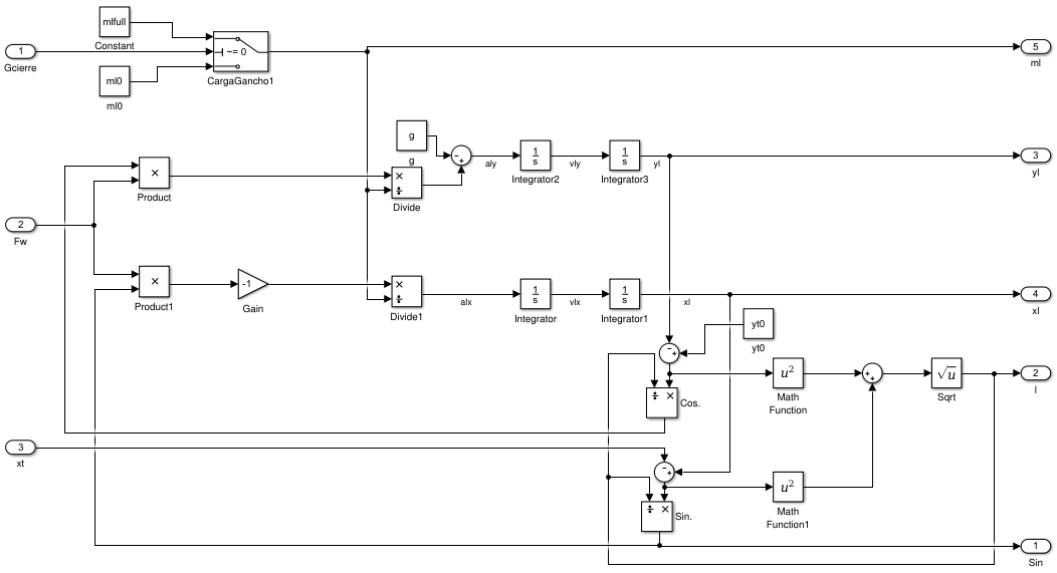


Cable elástico



Carga suspendida

Este bloque tiene una entrada binaria “Gcierre” que indica si el gancho está cerrado, y con esto se simula un peso que representa al contenedor suspendido.



Bloque del Controlador

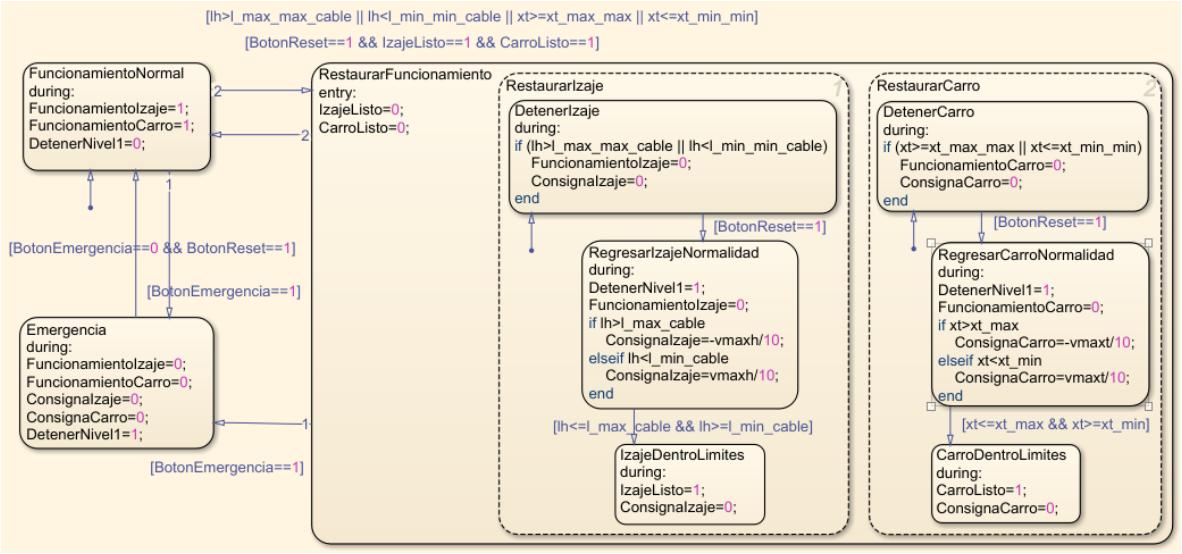
Nivel 0: Autómata de Seguridad

Si el carro o el izaje se salen de los límites, el autómata de seguridad toma el control, deteniendo el funcionamiento del accionamiento comprometido y permitiendo el uso del otro accionamiento. Detiene instantáneamente todas las consignas e inhibe al Nivel 1.

Al presionar el botón de *reset* el autómata toma el control completo, y lleva la planta a una condición dentro de los límites de operación segura.

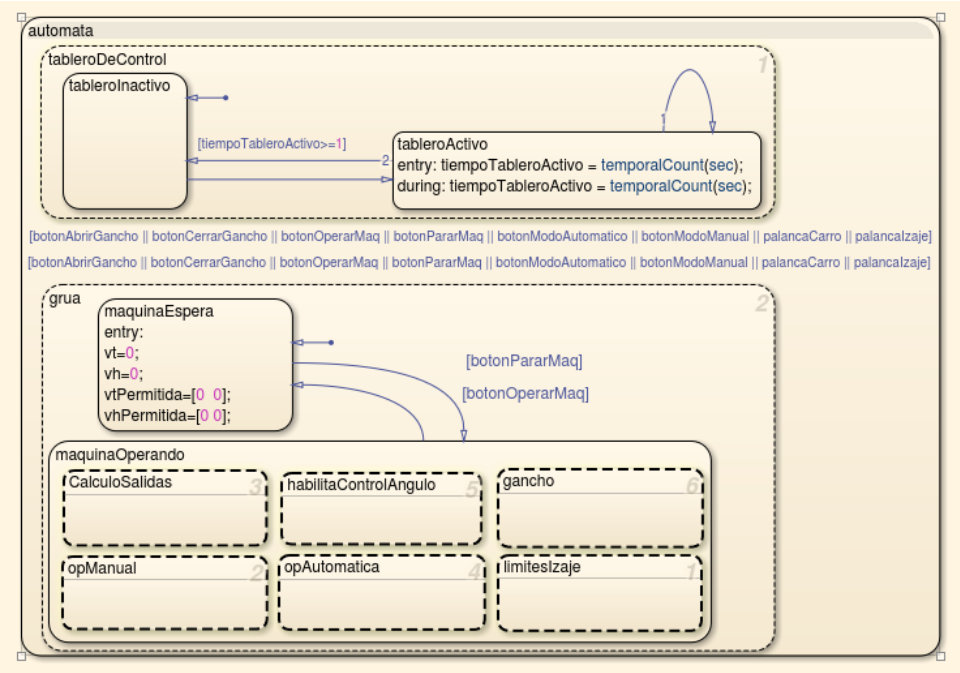
Al volver a presionar *reset*, si lo anterior se ha cumplido, se restaura el funcionamiento normal, y el autómata de nivel 1 pasa a estar en Espera.

Cuando se aprieta el botón de emergencia se detiene el funcionamiento de la maquina y del autómata de nivel 1.



Nivel 1: Control supervisor global

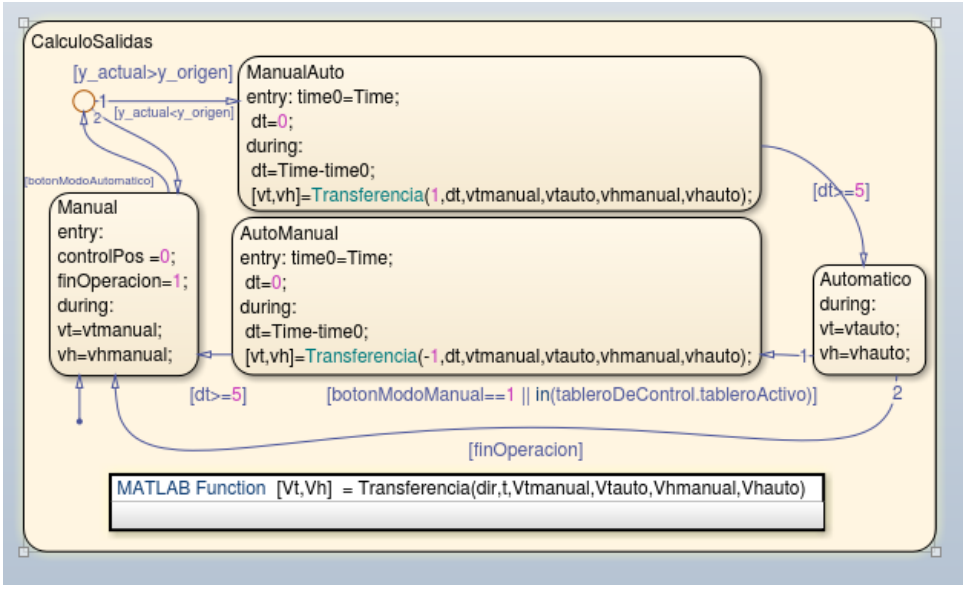
Este nivel propone un autómata de estados discretos activados por eventos, encargado de la operación de la máquina y el diagnostico de alarmas y fallas.



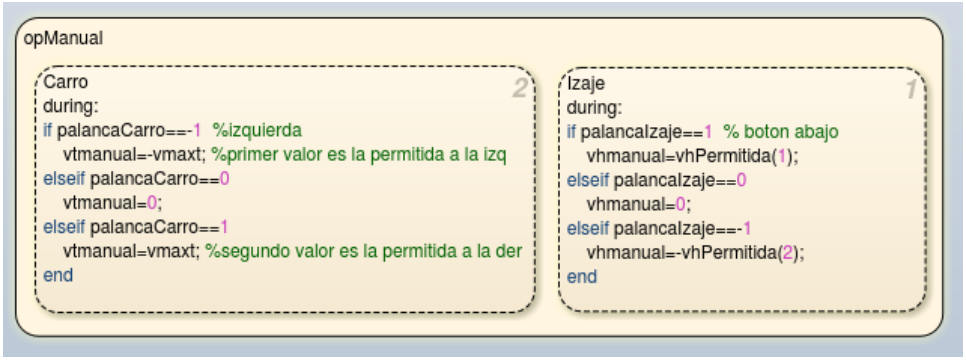
Inicialmente se ven dos estados paralelos:

1. tableroDeControl, que observa si el operario está manipulando el joypad o algún botón de la GUI;
2. grua, que implementa el resto de controles del sistema. La máquina inicia en Espera, y luego el usuario la puede llevar a operar para comenzar sus tareas.

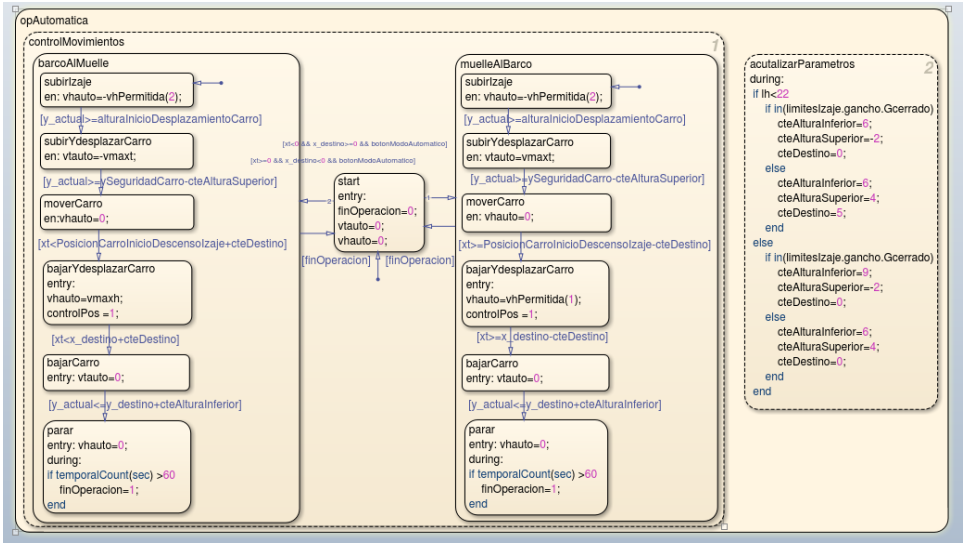
Este último bloque tiene a su vez 6 subestados paralelos que se describen a continuación.



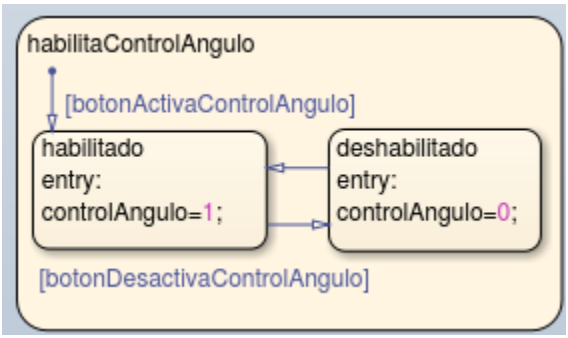
CalculoSalidas manipula los valores de salida del autómata, y gestiona las transiciones de consignas en el paso del modo manual al automático, generando una transición suave o *bumpless transfer*.



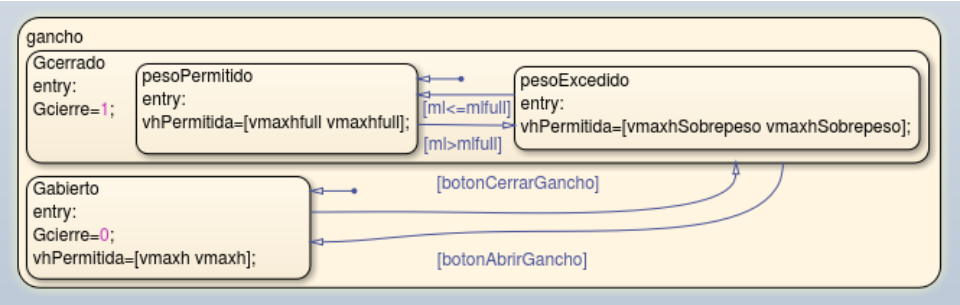
opManual permite la manipulación de la grúa generando consignas de velocidad en función de las entradas del operario con el joypad.



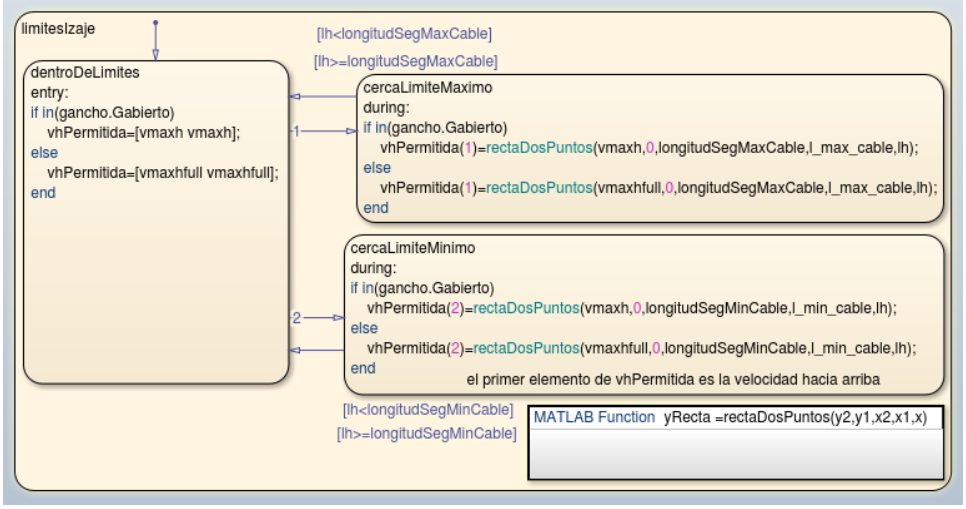
opAutomatica tiene dos bloques principales: por un lado controla la grúa en cuanto el operador presiona el botón de operación automática; y por otro lado, calcula una serie de constantes que sirven de ajuste fino para optimizar el funcionamiento de la operación.



HabilitaControlAngulo gestiona el control de balanceo según las indicaciones del operario.



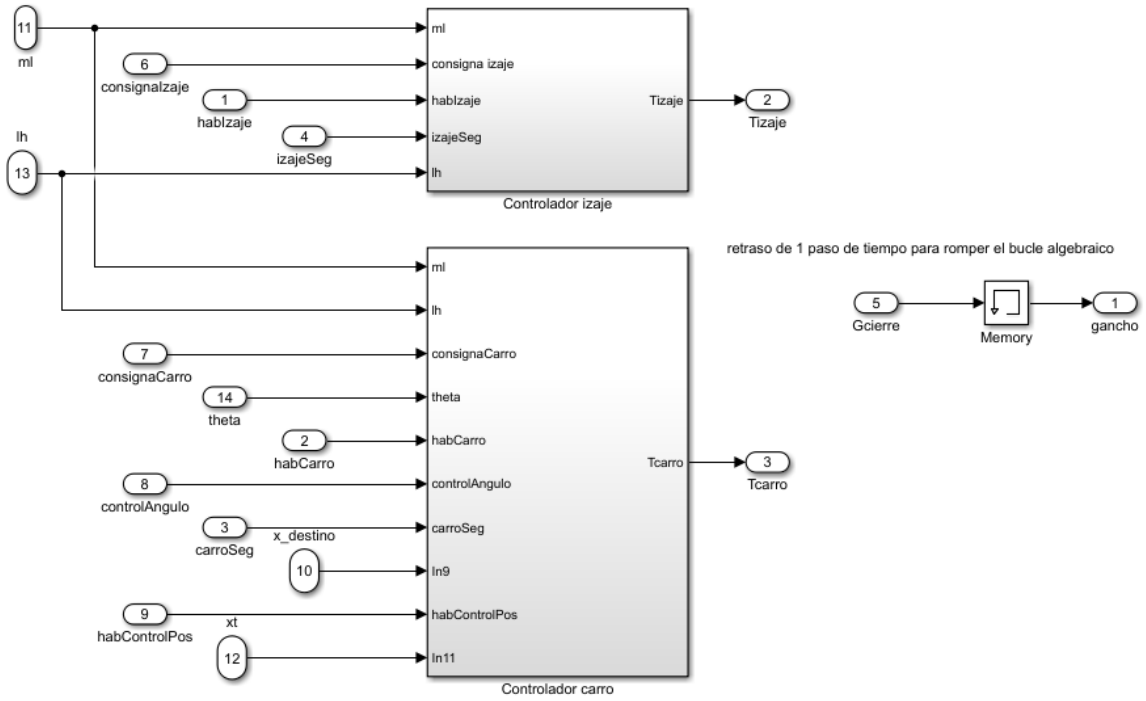
Gancho gestiona la apertura del sistema de gancho extensible o *spreader*. Además, activa una salida que simulara la carga de un contenedor.



limitesIzaje verifica las consignas de velocidad cuando el izaje está próximo a los límites de operación.

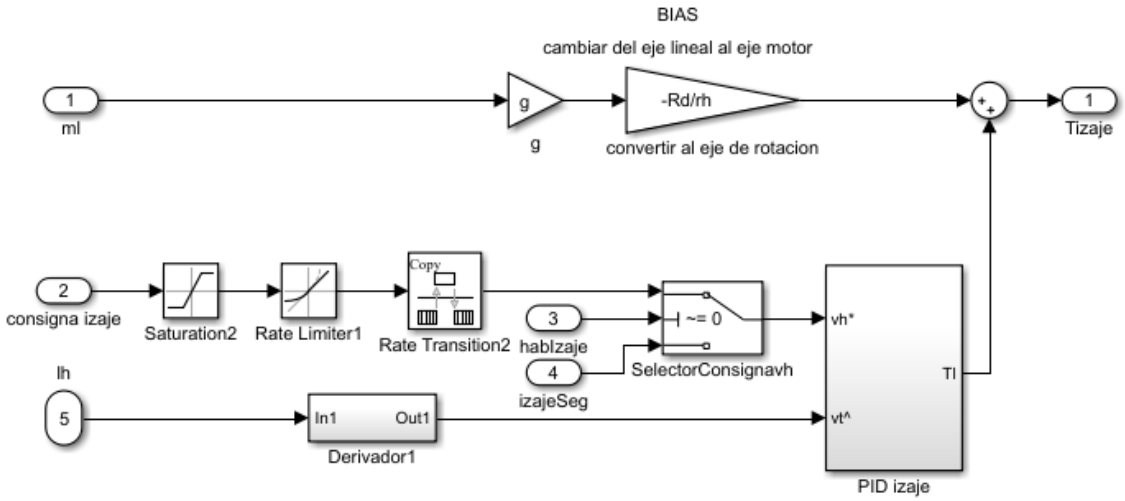
Nivel 2: Controladores de movimiento

Para todos los controladores se usará el método de sintonía serie. Se han implementado los PID discreto con bloques integradores y derivadores propios.

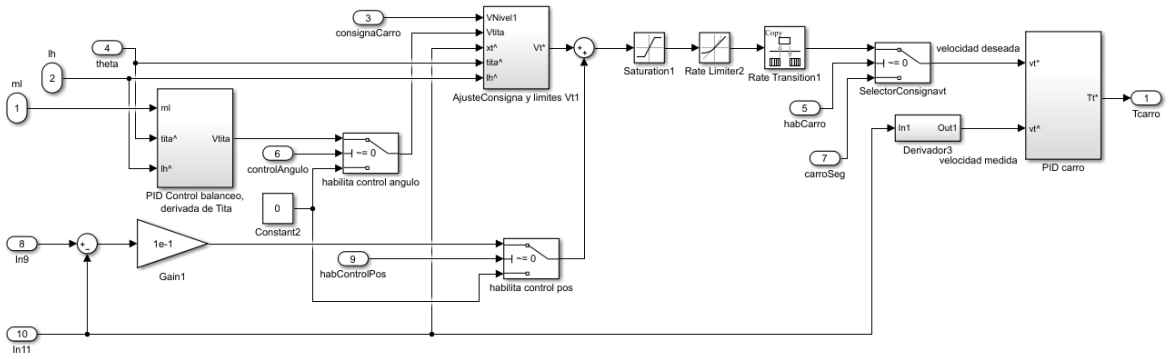


Izaje

Aquí se implementa un BIAS que contrarresta los efectos de la gravedad. También se observa el PID discreto y los limitadores de consignas.



Carro



En este bloque esta implementado el control de ángulo, el PID discreto del carro y un ajuste de consigna que no permite que se salga de los límites de operación. También están los limitadores de consigna.

Simulación en Codesys

El desarrollo e implementación del autómata híbrido de control se realizó en un entorno de programación de autómatas industriales bajo la norma IEC 61131 (CODESYS). Los lenguajes definidos por la norma que se utilizan son SFC (Bloques de función secuenciales) y ST (Texto Estructurado).

Inicialmente apareció la posibilidad de utilizar el modelo matemático de la planta hecho en Matlab y programar el controlador en Codesys. Para esto es necesaria una intercomunicación entre los dos, que se podría resolver mediante OPC. Dado que éste se usa en general para operaciones de visualización de datos y existe demora en la transmisión de la información, no se pudo lograr una sincronización en tiempo real de las dos aplicaciones. Por lo tanto, se decide reescribir tanto la planta como el controlador en Codesys, dejando solamente OPC para mostrar datos en la GUI desarrollada en Simulink.

En el anexo se describe como establecer dicha comunicación entre las dos aplicaciones.

Planta en ST

Se muestran las partes principales del código con las que se programó la planta en Codesys. El mismo se ejecuta cada 1 milisegundo.

```
1  Planta_CableElastico(Lh:= Planta_Izaje.Lh , l:= Planta_CargaSuspendida.l , vl:= Planta_Izaje.vl);
2  Planta_CargaSuspendida(Gancho:= Gcierre , Fw:= Planta_CableElastico.Fw , x:= Planta_Carro.x);
3  Planta_Carro(SenoCarga:= Planta_CargaSuspendida.Sn , Fw:= Planta_CableElastico.Fw , T:= Tt);
4  Planta_Izaje(Fw:= Planta_CableElastico.Fw , T:= Th);
5
6  xt:= Planta_Carro.x;
7  ml:= Planta_CargaSuspendida.m;
8  theta:= ASIN(Planta_CargaSuspendida.Sn);
9  lh:= Planta_Izaje.Lh;
```

Traslación del carro

```
IF (init=FALSE)
    THEN
    Integrador2(ValorActual:=-30/0.001);
    init:=TRUE;
END_IF
Integrador1(ValorActual:= (Fw*SenoCarga+T*rc_Rw-bt*Integrador1.Integral)*Mt_1);
Integrador2(ValorActual:= Integrador1.Integral);
x:= Integrador2.Integral;
```

Izaje de la carga

```
IF (init=FALSE)
    THEN
    Integrador2(ValorActual:=5/0.001);
    init:=TRUE;
END_IF
Integrador1(ValorActual:= (Fw-T*rh_Rd-bh*Integrador1.Integral)*Mh_1);
Integrador2(ValorActual:= Integrador1.Integral);
Lh:= Integrador2.Integral;
vl:= Integrador1.Integral;
```

Cable elástico

```
Derivador1(ValorActual:= l);
Fw:= (l-Lh)*Kw+(Derivador1.Derivada-vl)*bw;
```

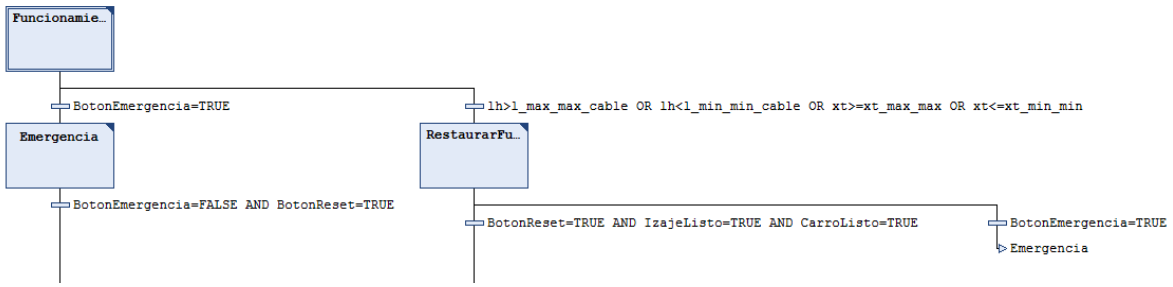
Carga suspendida

```
IF (init=FALSE)
    THEN
    Integradorx2(ValorActual:= -30/0.001);
    Integradory2(ValorActual:= 39.918277916666670/0.001);
    init:=TRUE;
END_IF
IF (Gancho=0)
    THEN
    m:=15000;
    ELSE
    m:=65000;
END_IF
l:= SQRT(EXPT(Integradorx2.Integral-x,2)+EXPT(-Integradory2.Integral+yt0,2));
Cs:= (-Integradory2.Integral+yt0)/l;
Sn:= (Integradorx2.Integral-x)/l;
Integradory1(ValorActual:= Cs*Fw/m-g);
Integradory2(ValorActual:= Integradory1.Integral);
Integradorx1(ValorActual:= -Sn*Fw/m);
Integradorx2(ValorActual:= Integradorx1.Integral);
```

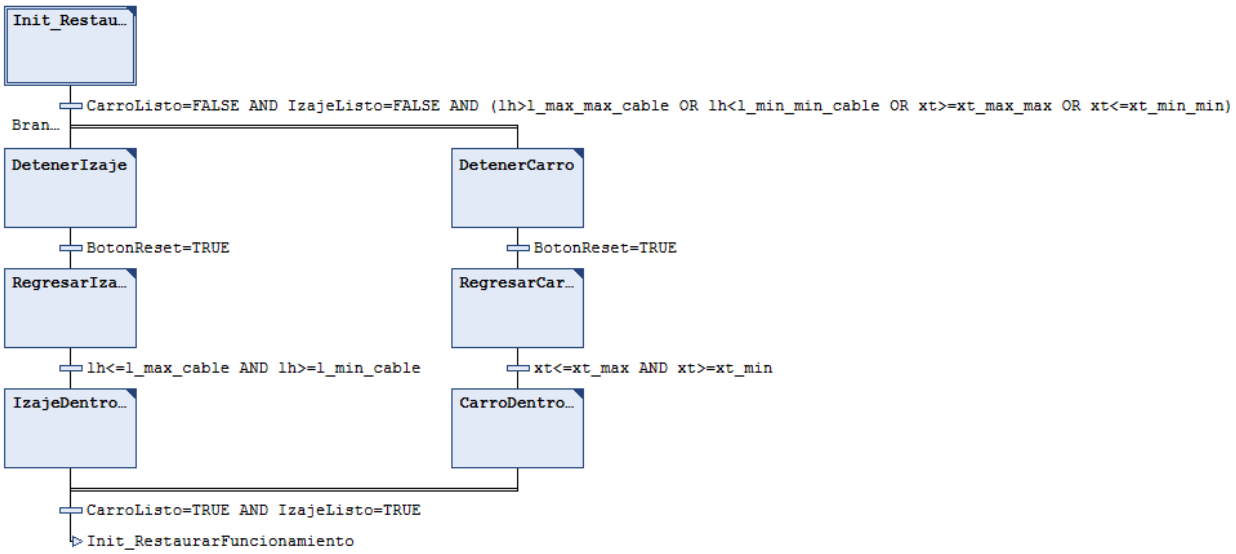
Controlador

Nivel 0: Autómata de Seguridad

El programa del autómata de seguridad se programó en SFC (“sequential function chart”), dicho diagrama de estados se consulta cada 100 milisegundos.

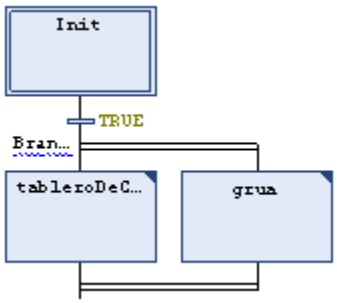


Restaurar Funcionamiento

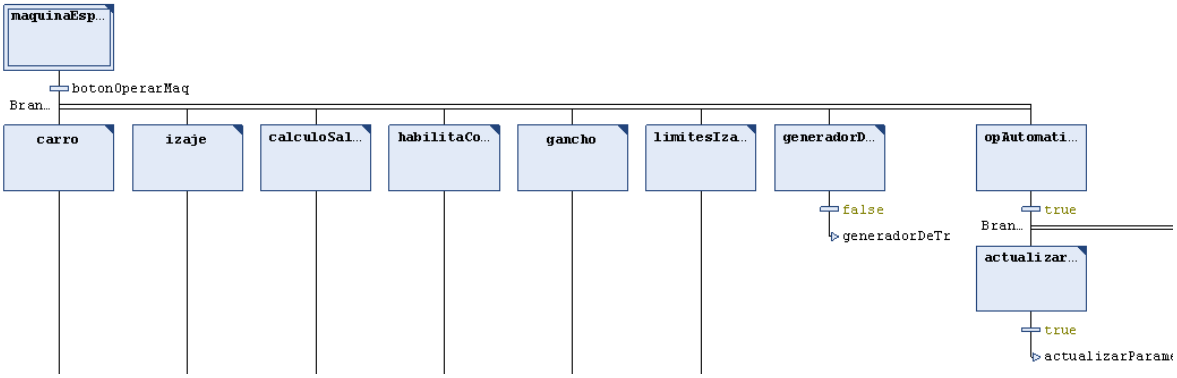


Nivel 1: Control Supervisor Global

El programa del autómata de control supervisor se programó en SFC, dicho diagrama de estados se consulta cada 50 milisegundos.



Dentro de grúa se despliega el siguiente esquema (no se muestra el ciclo de operación automática).



Se busca en todo momento mantener la igualdad con el modelo Simulink, salvando algunos detalles de la implementación, inherentes al nuevo entorno de desarrollo.

Nivel 2: Controladores de movimiento

El controlador de movimientos se programó en ST, y se ejecuta cada 22 milisegundos.

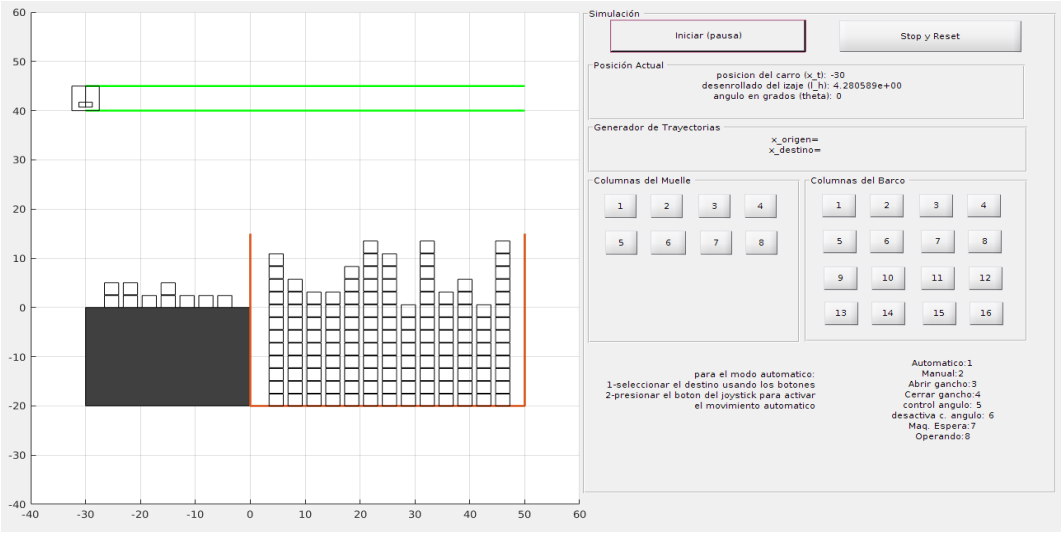
```
//Izaje
Saturacionl(Limite:=vmaxh , Entrada:=vh);
RateLimiterl(Limite:=amaxh , Entrada:=Saturacionl.Salida);
Derivadorl(ValorActual:=lh , Paso:=0.01);
IF (FuncionamientoIzaje=TRUE)
    THEN
    PIDl(vhConsigna:=RateLimiterl.Salida , vhMedido:=Derivadorl.Derivada);
    ELSE
    PIDl(vhConsigna:=ConsignaIzaje , vhMedido:=Derivadorl.Derivada);
END_IF
Th:=PIDl.th+Bias*ml;
//CARRO
ControlTita(lh:=lh , ml:=ml , TitaMedido:=theta);
IF (ControlPos=TRUE)
    THEN
    ConsignaPos:= (x_destino-xt)*0.1;
    ELSE
    ConsignaPos:= 0;
END_IF
IF (ControlAngulo=TRUE)
    THEN
    Saturaciont(Limite:=vmaxt , Entrada:=AjusteBordesCarro(xt,AjusteGananciasCarro(vt,ControlTita.Vtita,theta,lh))+ConsignaPos);
ELSE
    Saturaciont(Limite:=vmaxt , Entrada:=AjusteBordesCarro(xt,AjusteGananciasCarro(vt,0,theta,lh))+ConsignaPos);
END_IF
RateLimitert(Limite:=amaxt , Entrada:=Saturaciont.Salida);
Derivadort(ValorActual:=xt , Paso:=0.01);
IF (FuncionamientoCarro=TRUE)
    THEN
    PIDt(vtConsigna:=RateLimitert.Salida , vtMedido:=Derivadort.Derivada);
    ELSE
    PIDt(vtConsigna:=ConsignaCarro , vtMedido:=Derivadort.Derivada);
END_IF
Tt:=PIDt.tt;
```

Interfaz hombre máquina

Interfaz Gráfica de Usuario

Para actualizar los valores en la interfaz con las posiciones del carro y del izaje, y poder visualizarlos en forma de animación, se usa la instrucción *get_param* sobre los bloques que corresponden a la posición del carro y del izaje. De esta manera, se obtiene un objeto que contiene la posición y otros datos del bloque en cuestión y permite acceder a los datos del bloque durante la simulación. Esto es posible a partir de Simulink 2011.[7][8]

Para actualizar los valores mostrados en la GUI se utiliza un *toggle button* que junto con un bucle *while*, permite ir modificando los elementos mostrados en la pantalla, con la frecuencia que se desee.



Entrada de datos por parte del operador

Para la entrada de datos a la simulación se usa un *joypad* de 8 botones, donde las flechas de mando manipulan las consignas de velocidad del carro e izaje, y los demás botones activan estados dentro del *chart.* dentro de la GUI se detallan más usos del *joypad.*

Resultados

Se realiza una simulación partiendo desde el muelle en la primera posición, hasta el barco en la décima columna. Para esto se utiliza el modo de operación automática, y a continuación se muestran los resultados, a la izquierda con el gancho vacío, y a la derecha con el gancho cerrado, cargando un contenedor.

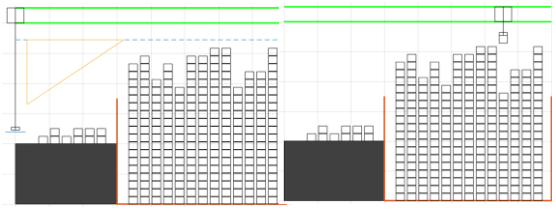


Figura 5: Vista de la simulación

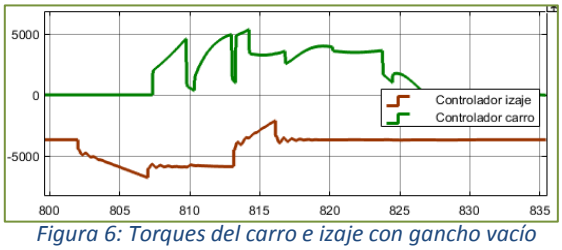


Figura 6: Torques del carro e izaje con gancho vacío

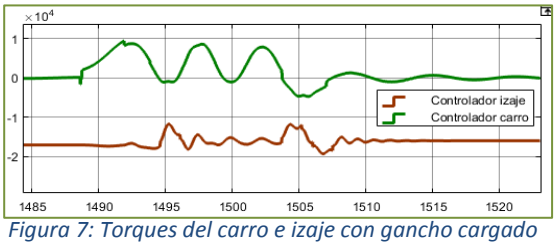


Figura 7: Torques del carro e izaje con gancho cargado

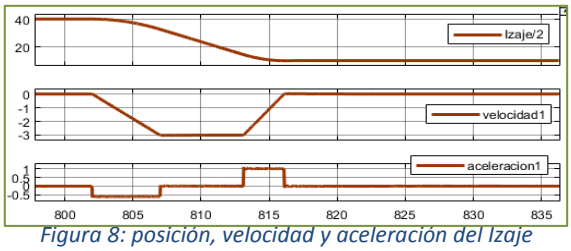


Figura 8: posición, velocidad y aceleración del Izaje

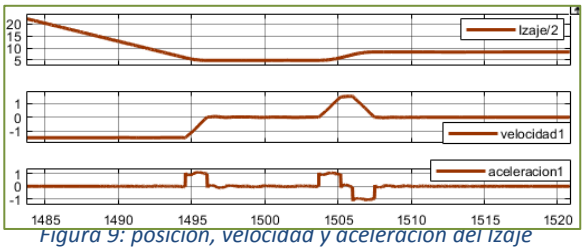


Figura 9: posición, velocidad y aceleración del Izaje

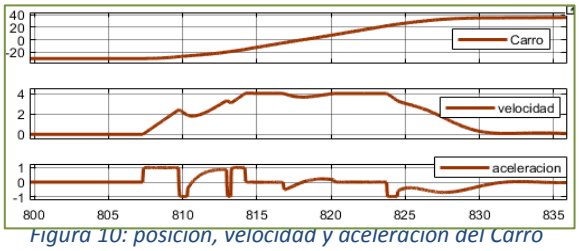


Figura 10: posición, velocidad y aceleración del Carro

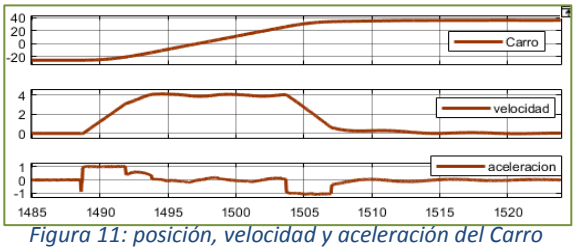


Figura 11: posición, velocidad y aceleración del Carro

Conclusiones

En el presente trabajo se desarrolló exitosamente un autómata capaz de controlar el movimiento semiautomático de una aplicación industrial simplificada. Se obtuvo un sistema dinámico híbrido que engloba una interacción de aspectos continuos y discretos.

Se logró una generación de trayectorias que permite minimizar el tiempo de trabajo, manteniendo el ángulo de balanceo pequeño.

A futuro se puede evaluar el desarrollo de un control de balanceo mediante lógica difusa, que después de plantear la borrosificación y deborrosificación, implica una matemática sencilla para el controlador y más rápida hablando de tiempo de cálculo. Una consideración futura es la comparación entre los esfuerzos de control de ambos métodos ya que se afirma que en los controladores difusos [9] son menores para el caso del péndulo, lo que podría significar un ahorro de energía conveniente para el proyecto.

Luego de la verificación del modelo en Matlab se programó en Codesys, adecuándose al estándar IEC 61131-3 y usando las técnicas de programación propuestas por la norma.

En ambos casos se acompañó la simulación con una interfaz de usuario fácil de interpretar y que ayuda al operario a conocer el estado de la máquina.

Bibliografía

[1] G. Julian, “Autómatas y Control Discreto, Guía de Trabajo Final Integrador, Universidad Nacional de Cuyo.” Mendoza, 2018.

[2] K. Ogata, *Ingeniería de control moderna*, 5th ed. Madrid: Pearson Education, 2010.

[3] L. RAMÍREZ-GONZÁLEZ, J. GARCÍA-MARTÍNEZ, S.-V. y Xóchitl, and R. GARCIA-RAMOS, “Método de Euler-Lagrange en el modelado y control de un péndulo invertido sobre un carro,” *Revista de Ingeniería Eléctrica*, vol. 1, pp. 1–8, 2017.

[4] R. W. Clough and J. Penzien, *Dynamics of structures*, 3rd ed. 1995.

[5] K. Ogata, *Sistemas de Control en Tiempo Discreto*, 2nd ed. Mexico: Prentice Hall, 1996.

[6] J. J. Craig, *Introduction to Robotics*, 3rd ed. New Jersey: Pearson Prentice Hall, 2005.

[7] la.mathworks.com, “reading-a-wire-value-from-simulink-into-the-command-window-or-guide,” 2012. [Online]. Available: https://la.mathworks.com/matlabcentral/answers/46363-reading-a-wire-value-from-simulink-into-the-command-window-or-guide.

[8] la.mathworks.com, “accessing-block-data-during-simulation,” 2018. [Online]. Available: https://la.mathworks.com/help/simulink/ug/accessing-block-data-during-simulation.html.

[9] F. A. Raheem, B. F. Midhat, and H. S. Mohammed, “PID and Fuzzy Logic Controller Design for Balancing Robot Stabilization,” vol. 18, no. 1, pp. 1–10, 2018.

[10] R. Cobo, “OPC:El estándar para comunicaciones entre dispositivos y sistemas de control de procesos.” [Online]. Available: http://www.emb.cl/electroindustria/articulo.mvc?xid=764&srch=fabelec&act=3.

[11] www.matrikonopc.es, “OPC UA (Arquitectura Unificada).” [Online]. Available: https://www.matrikonopc.es/opc-ua/index.aspx.

Anexo: Comunicación y estándar OPC

OPC (OLE for Process Control, o Object Linking and Embedding for Process Control) es un estándar para comunicaciones entre dispositivos y sistemas de control de procesos, permitiendo el intercambio de datos entre distintas fuentes tales como controladores programables, unidades remotas, bases de dato y HMI.

De acuerdo a la aplicación que se desee dar OPC puede tener diferentes especificaciones tales como OPC-DA que está pensado para la adquisición de datos en tiempo real, trabajando solo con el último dato sin utilizar estampa de tiempo, OPC-HDA es similar al anterior pero operando con datos históricos, OPC-A&E se utiliza para compartir información de eventos, OPC-DX permite intercambiar datos entre distintos servidores OPC, OPC-XML permite intercambiar datos entre sistemas con distintos sistemas operativos. [10]

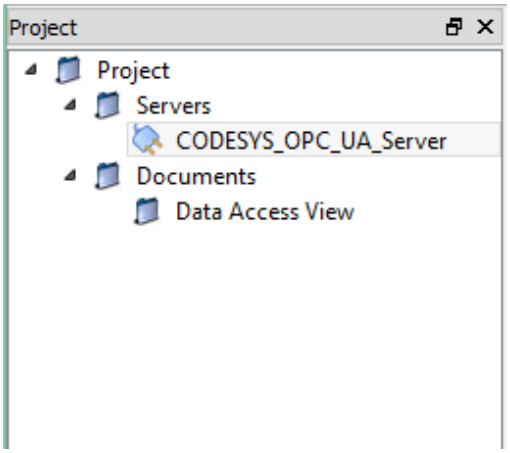
El estándar de OPC UA surge con la necesidad de portabilizar el estándar OPC (basado en Microsoft DCOM) a sistemas de distintos fabricantes, unificando los protocolos OPC DA, HDA, A&E y Seguridad.[11]

Implementación de servidor y cliente OPC con UAExpert y Codesys

Para realizar la implementación de un servidor OPC necesita en primer lugar tener instalados los softwares de Codesys y UAExpert, ambos disponibles de forma gratuita en sus sitios oficiales.

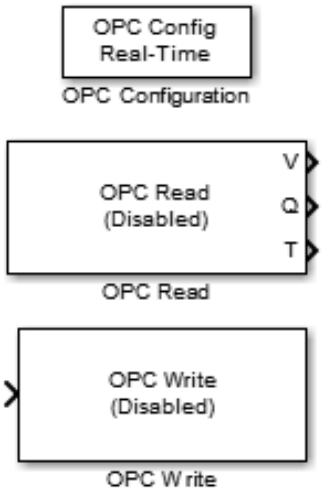
En segundo lugar, para iniciar las variables que se utilizan en el servidor debe crearse un proyecto en Codesys y haciendo clic derecho en *application* seleccionamos agregar objeto-> Lista de variables globales, esto generara una lista con variables globales donde deberán estar colocadas las variables que se compartirán con el servidor OPC. Posteriormente de igual manera, haciendo clic derecho en *application* seleccionamos agregar objeto-> Symbol *configuration*, en el cuadro de diálogo seleccionamos "Support OPC UA Features", lo que permite configurar que variables se compartirán con el servidor, para esto el programa nos pedirá que ejecutemos una *build* para poder configurar los símbolos, luego de realizada seleccionamos las variables que se compartirán con el servidor, tanto entradas como salidas. Debería quedar de esta forma:

Posteriormente, con el PLC activado y el programa cargado abrimos UAExpert, en proyect ->Servers hacemos clic derecho y seleccionamos Add, en el cuadro de diálogo que se abre seleccionamos CODESYS_OPC_UA_Server, luego hacemos clic en Connect server en la barra de herramientas y ya el servidor queda armado y funcionando.



Implementación de un cliente OPC con Matlab y Simulink

Para implementar un cliente OPC en Simulink es necesario en primer lugar instalar y activar el toolbox de OPC de Matlab, para ello se ejecuta el comando *opcregister*, posteriormente en Simulink colocamos el bloque OPC configuration, el cual permite configurar la conexión con un servidor OPC. Dentro del bloque seleccionamos Configure OPC Clients, seleccionamos Add y luego Select para seleccionar el servidor.



Luego para realizar la lectura se coloca el bloque OPC Read, el cual contiene tres puertos de salida, como solo nos interesa el dato leído en el servidor, dentro del cuadro desmarcamos las opciones de Quality Port y Time Stamp para quedarnos solo con el puerto que necesitamos. Posteriormente para agregar Items para su lectura seleccionamos Add Items y buscamos en la lista de variables globales las variables que necesitemos, las cuales se encontrarán multiplexadas en el orden dado por el cuadro.

Para realizar la escritura utilizamos el bloque OPC Write, el cual se configura de la misma forma que el bloque de lectura. En la siguiente imagen se muestran los tres bloques necesarios para realizar la comunicación con el servidor.