

# PROYECTO INTEGRADOR DE AUTÓMATAS Y CONTROL DISCRETO

Juan Pablo Sibecas  
juan.sibecas@gmail.com

Matias Gaviño  
matias.linares.g@gmail.com

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

Junio de 2024

**Resumen**

# 1. Introducción

El presente informe aborda el control de una grúa portuaria destinada a la carga, descarga y reubicación de contenedores entre el barco y las bahías de carga. El objetivo es mejorar la eficiencia del sistema mediante trayectorias de movimiento que sean tanto suaves como rápidas.

Para desarrollar este control, se comienza con el modelado del sistema físico, que consta de un carro para el movimiento horizontal y un sistema de izaje para el movimiento vertical. Ambos movimientos están acoplados por la carga, la cual consiste en un contenedor y un "spreader" que lo sostiene. La traslación y el izaje son accionados por motores eléctricos, lo que permite un control preciso de las trayectorias.

El desarrollo del informe incluye el modelado del sistema físico, obteniendo las ecuaciones de movimiento y utilizando Simulink para simular su comportamiento. A continuación, se diseña un control híbrido de tres niveles: el nivel 2 se encarga del control en tiempo discretizado para los motores que accionan el izaje y el carro; el nivel 1 consiste en un controlador discreto basado en eventos, que genera trayectorias suaves y eficientes; y el nivel 0 actúa como sistema de seguridad, asegurando que el sistema entre en un estado seguro en caso de fallas. Para la simulación del controlador, se emplea el software Matlab/Simulink, y luego se implementa en Codesys para simular su ejecución en un PLC.

## 2. Desarrollo

### 2.1. Modelo del Sistema Físico

El modelo del sistema se simplifica a un control de posición de la carga en el plano. No se tendrán en cuenta grados de libertad que no están contemplados en el plano perpendicular al eje longitudinal del muelle. Se considera que la estructura de la grúa es rígida y que no se producen vibraciones, no así los cables que se consideran flexibles. Para el desarrollo de las ecuaciones se parte desde lo desarrollado en el enunciado del trabajo práctico.

#### 2.1.1. Subsistema de Izaje

Segunda ley de Newton del lado tambor:

$$J_{hd+heb} \frac{d\omega_{hd}}{dt} = T_{hd}(t) + T_{heb}(t) - b_{hd}\omega_{hd}(t) - T_{hdl}(t) \quad (1)$$

Segunda ley de Newton del lado motor:

$$J_{hm+hb} \frac{d\omega_{hm}}{dt} = T_{hm}(t) + T_{hb}(t) - b_{hm}\omega_{hm}(t) - T_{hml}(t) \quad (2)$$

relación de transmisión

$$i_h = \frac{\omega_{hm}(t)}{\omega_{hd}(t)} = \frac{T_{hd}(t)}{T_{hml}(t)} \quad (3)$$

si reemplazo 3 en 2 y despejo  $T_{hd}(t)$

$$T_{hd}(t) = J_{hm+hb} \frac{d\omega_{hd}}{dt} i_h^2 - b_{hm}\omega_{hd}(t) i_h^2 + i_h(T_{hm}(t) + T_{hb}(t)) \quad (4)$$

reemplazando en 1 y operando se obtiene

$$(J_{hd+heb} + J_{hm+hb} i_h^2) \frac{d\omega_{hd}}{dt} = -(b_{hd} + b_{hm} i_h^2) \omega_{hd}(t) + i_h(T_{hm}(t) + T_{hb}(t)) + T_{heb}(t) - T_{hdl}(t) \quad (5)$$

como  $T_{hdl}(t) = F_{hw}(t) * r_{hd}$ ,  $2V_h = r_{hd} * \omega_{hd}(t)$  y  $V_h = -\frac{dl_h(t)}{dt}$  y dividiendo por  $r_{hd}$ :

$$2 \frac{(J_{hd+heb} + J_{hm+hb} i_h^2)}{r_{hd}^2} \frac{d^2 l_h(t)}{dt^2} = -2 \frac{(b_{hd} + b_{hm} i_h^2)}{r_{hd}^2} \frac{dl_h(t)}{dt} - \frac{i_h}{r_{hd}} (T_{hm}(t) + T_{hb}(t)) - \frac{T_{heb}(t)}{r_{hd}} + F_{hw}(t) \quad (6)$$

Reemplazando por parametros equivalentes:

$$M_{Eh}\ddot{l}_h(t) = -b_{Eh}\dot{l}_h(t) - \frac{i_h}{r_{hd}}(T_{hm}(t) + T_{hb}(t)) - \frac{T_{hEb}(t)}{r_{hd}} + F_{hw}(t) \quad (7)$$

Donde

$$M_{Eh} = 2 \frac{(J_{hd+hEb} + J_{hm+hbi_h^2})}{r_{hd}^2} \quad (8)$$

$$b_{Eh} = 2 \frac{(b_{hd} + b_{hm}i_h^2)}{r_{hd}^2} \quad (9)$$

$$(10)$$

En la figura 1 se muestra el modelo de Simulink del subsistema de izaje. Se observa que se implementa la ecuación 7 con sus respectivos parámetros equivalentes. Además, se incluye el modelo del sistema de freno y freno de emergencia.

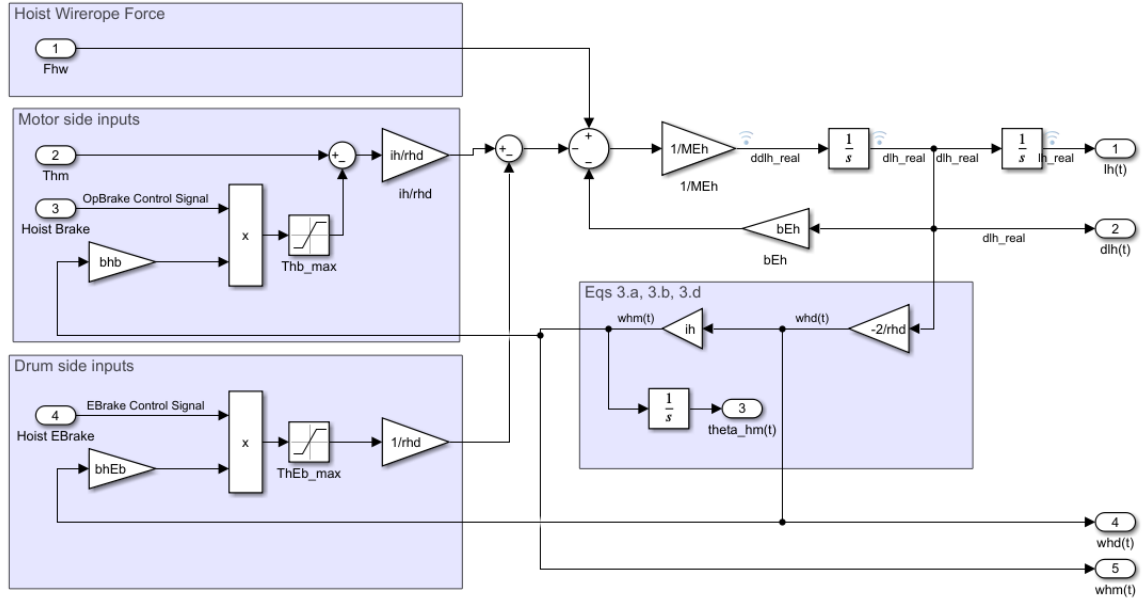


Figura 1: Modelo de Simulink del subsistema de izaje

### Modelo del cable de izaje

El modelo del cable de izaje se obtiene a partir de la ecuación 2/2.a. Además, se considera que el cable solo soporta cargas de tracción por lo que si el resultado de la tensión obtenido por dicha ecuación es negativo se toma como cero. En la figura 2 se muestra el modelo implementado en Simulink del cable de izaje.

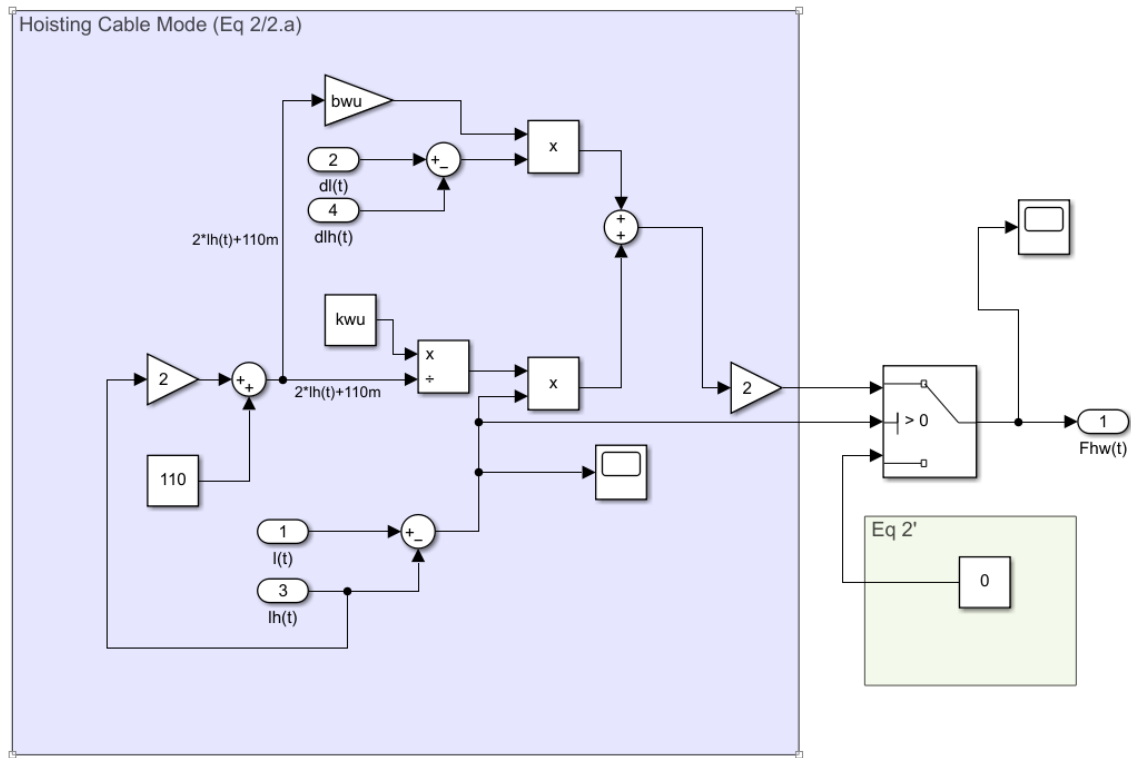


Figura 2: Modelo de Simulink del cable de izaje

### 2.1.2. Subsistema Carro

La conducción de carro se modela como tres subsistemas, el de accionamiento del motor, el del tambor y el del cable. En la figura 3 se muestra el modelo de Simulink de estos subsistemas.

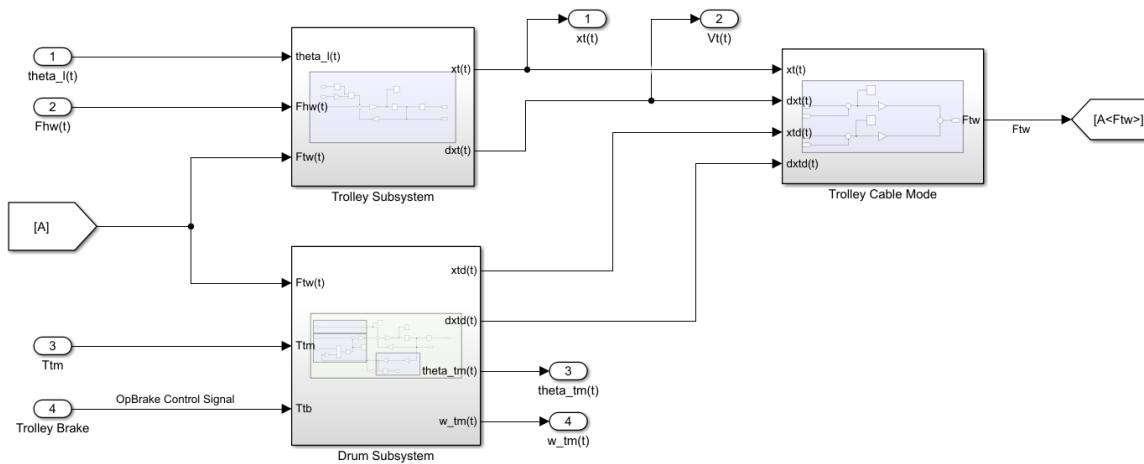


Figura 3: Modelo de Simulink del subsistema carro

### Motor del carro

El modelo se obtiene a partir de la ecuación 4 del enunciado. En la figura 3 se muestra el modelo implementado en Simulink.

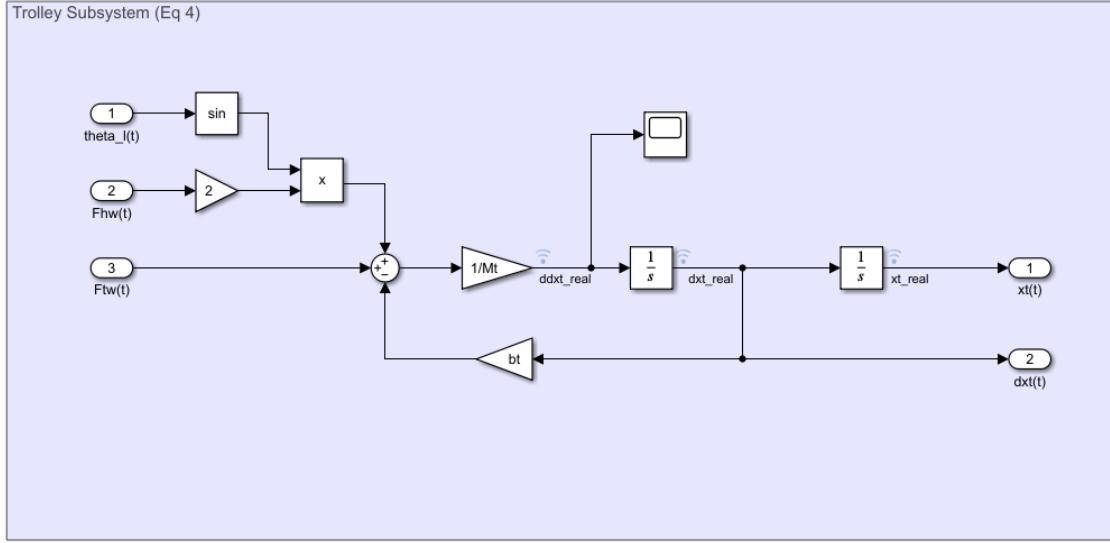


Figura 4: Modelo de Simulink del subsistema carro

### Tambor del carro

Segunda ley de Newton del lado tambor:

$$J_{td} \frac{d\omega_{td}(t)}{dt} = T_{td}(t) - b_{td}\omega_{td}(t) - T_{tdl}(t) \quad (11)$$

Segunda ley de Newton del lado motor:

$$J_{tm+tb} \frac{d\omega_{tm}(t)}{dt} = T_{tm}(t) + T_{tb}(t) - b_{tm}\omega_{tm}(t) - T_{tml}(t) \quad (12)$$

relacion de transmision

$$i_t = \frac{\omega_{tm}(t)}{\omega_{td}(t)} = \frac{T_{td}(t)}{T_{tml}(t)} \quad (13)$$

si reemplazo 13 en 12 y despejo  $T_{td}(t)$

$$T_{td}(t) = J_{tm+tb} \frac{d\omega_{td}(t)}{dt} i_t^2 - b_{tm}\omega_{td}(t) i_t^2 + i_t(T_{tm}(t) + T_{tb}(t)) \quad (14)$$

Reemplazo 14 en 11 y reordeno:

$$(J_{td} + J_{tm+tb} * i_t^2) \frac{d\omega_{td}(t)}{dt} = i_t(T_{tm}(t) + T_{tb}(t)) - (b_{td} + b_{tm}i_t^2)\omega_{td}(t) - T_{tdl}(t) \quad (15)$$

Como  $\omega_{td}(t)r_{td} = V_{td}(t)$ ,  $F_{tw}(t)r_{td} = T_{tdl}(t)$  y  $V_{td}(t) = \frac{dx_{td}}{dt}$  y dividiendo por  $r_{td}$ :

$$\frac{(J_{td} + J_{tm+tb} * i_t^2)}{r_{td}^2} \frac{d^2x_{td}(t)}{dt^2} = -\frac{(b_{td} + b_{tm}i_t^2)}{r_{td}^2} \frac{dx_{td}(t)}{dt} + \frac{i_t}{r_{td}}(T_{tm}(t) + T_{tb}(t)) - F_{tw}(t) \quad (16)$$

Reemplazando por parametros equivalentes se obtiene la ecuacion del tambor del subsistema carro:

$$M_{Etd}\ddot{x}_{td}(t) = -b_{Etd}\dot{x}_{td}(t) + \frac{i_t}{r_{td}}(T_{tm}(t) + T_{tb}(t)) - F_{tw}(t) \quad (17)$$

La ecuacion de movimiento del carro es:

$$M_t\ddot{x}_t(t) = -b_t\dot{x}_t(t) + F_{tw}(t) + 2F_{hw}(t) \sin \theta_l(t) \quad (18)$$

Y la fuerza transmitida por el cable del subsistema carro es:

$$F_{tw}(t) = K_{tw}(x_{td}(t) - x_t(t)) + b_{tw}(\dot{x}_{td}(t) - \dot{x}_t(t)) \quad (19)$$

En la figura 5 se muestra el modelo de Simulink que responde a la ecuación 18 con sus respectivos parametros equivalentes. Además, se introduce el modelo del freno del tambor.

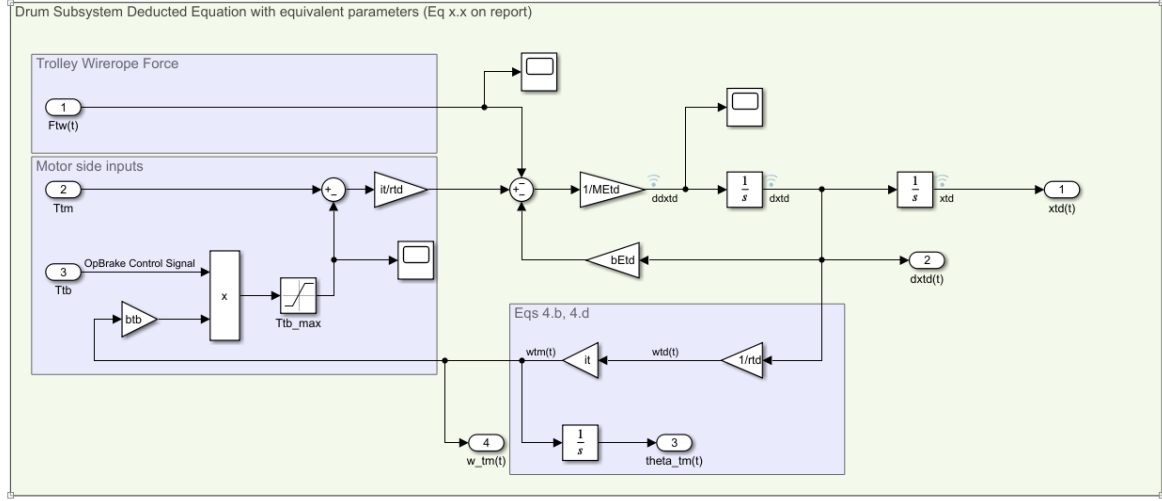


Figura 5: Modelo de Simulink del tambor del subsistema carro

### Modelo del cable del carro

La figura 6 muestra el modelo de Simulink del cable del subsistema carro. Se observa que se implementa la ecuación 4.a del enunciado para su modelado.

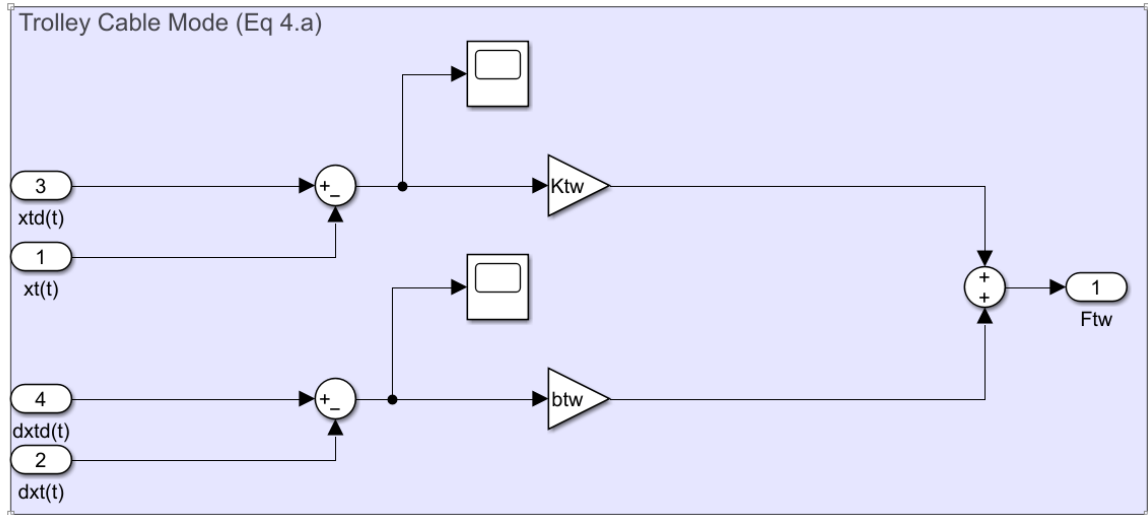


Figura 6: Modelo de Simulink del cable del subsistema carro

### 2.1.3. Perfil de obstáculos

El perfil de obstaculos se modela como una matriz de Nx2 donde la primera columna representa la posicion en x y la segunda la altura del obstáculos perfil del obstáculo. El perfil esta discretizado en x segun un dx determinado. AL iniciar la simulacion se inicializa esta matriz con el perfil de los obstáculos inmoviles. Luego se actualiza la matiz teneniendo en cuenta la cantidad de containes en cada comunma del barco o en la bahía de carga inicialmente. Cuando durante la simulación se recoje o se deja un contenedor se actualiza la matriz de obstáculos mediante el modelo en stateflow que se muestra en la figura 7.

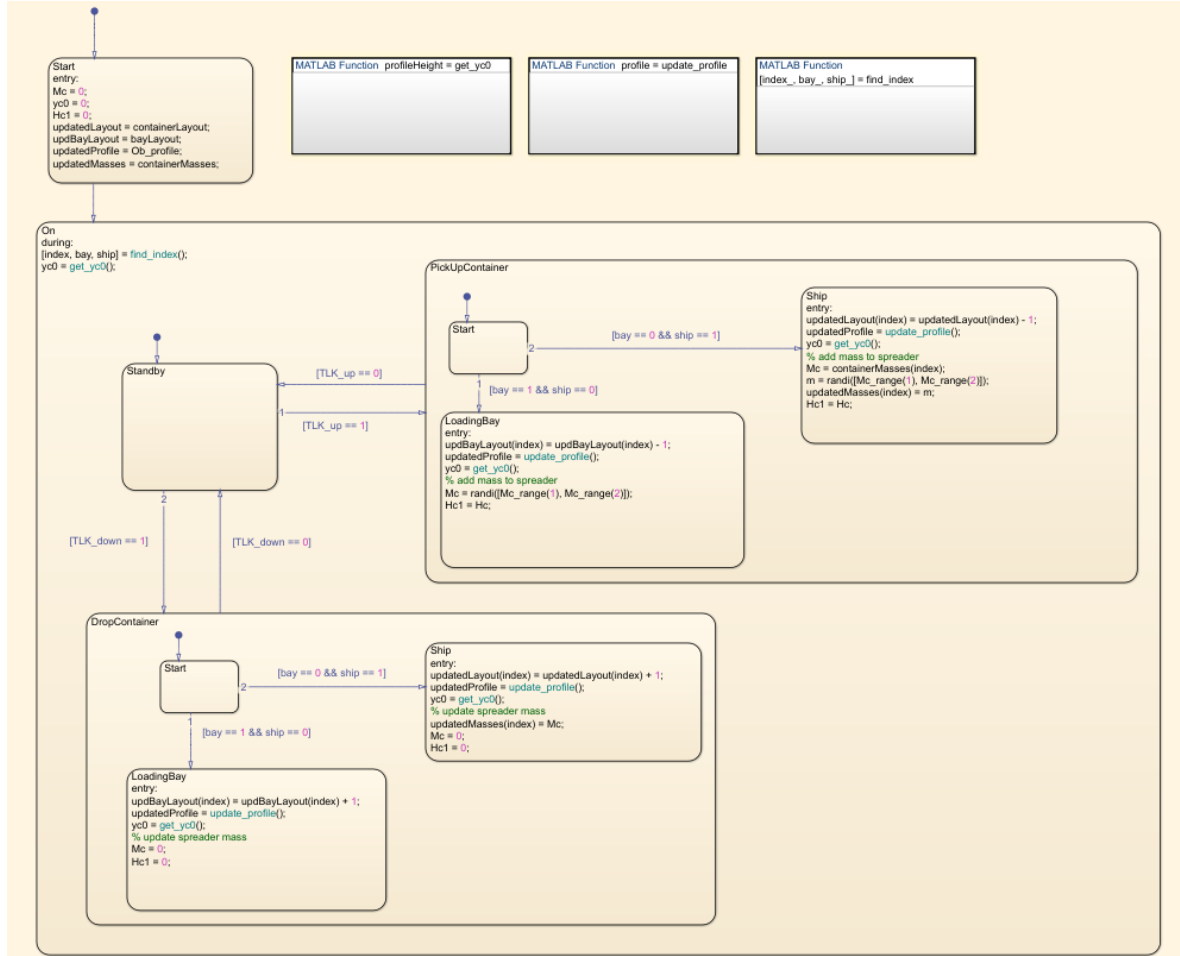


Figura 7: Modelo en stateflow de la actualización del perfil de obstáculos

#### 2.1.4. Modelo de la carga

La carga se modela teniendo en cuenta la tensión del cable de izaje, así como muestran las ecuaciones 1.a y 1.b del enunciado. En la figura 8 se muestra el modelo de Simulink de la carga.

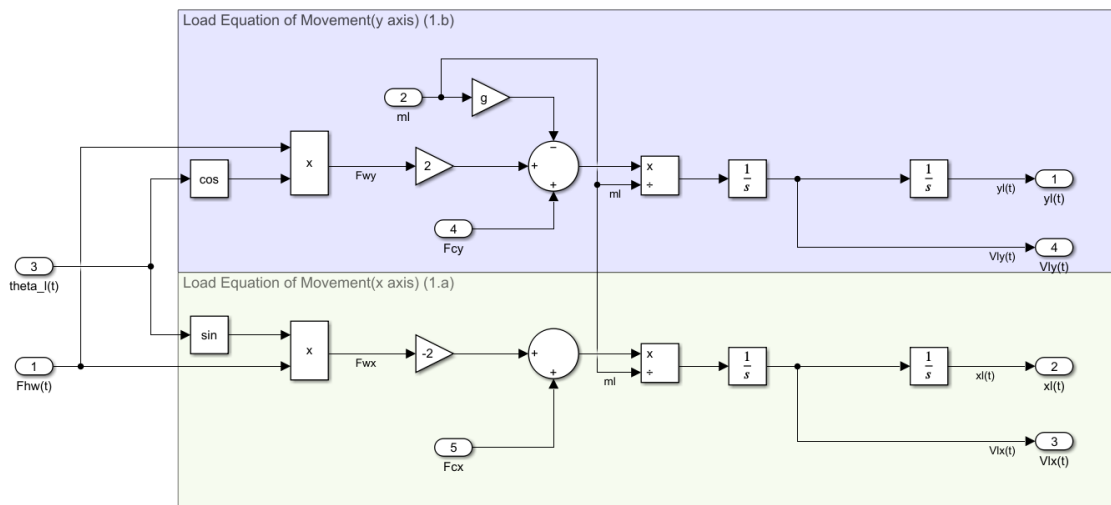


Figura 8: Modelo de Simulink de la carga

El ángulo del cable de izaje respecto de la vertical se obtiene a partir de la posición de la carga y el carro, así como indica la ecuación 0.d del enunciado. En la figura 9 se muestra el modelo de Simulink del balanceo de la carga.

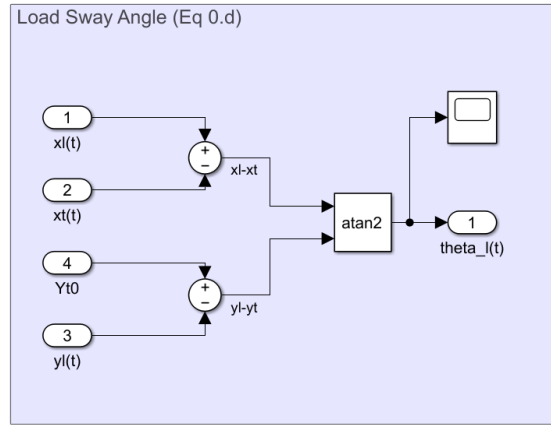


Figura 9: Modelo de Simulink del balanceo de la carga

Al igual que el ángulo del cable de izaje, el largo del mismo se obtiene a partir de la posición de la carga y el carro, así como indica la ecuación 0.c del enunciado. La velocidad de cambio de esta longitud se calcula de forma analítica utilizando la ecuación 1.c'. En la figura 10 se muestra el modelo de Simulink lo anteriormente mencionado.

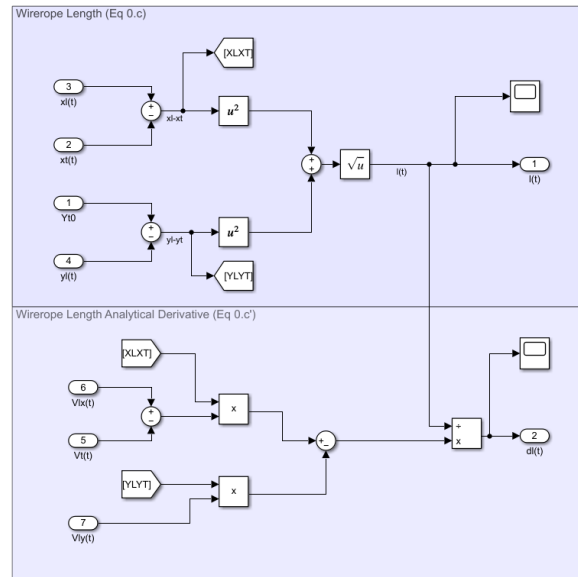


Figura 10: Modelo de Simulink del largo del cable de izaje

### Modelo de la fuerza de contacto

La fuerza de contacto de la carga con el perfil de obstáculos se modela a partir de las ecuaciones 1.a y 1.b del enunciado. En la figura 11 se muestra el modelo de Simulink de la fuerza de contacto.

## 2.2. Diseño del controlador

$$T'_m(t) = b_a e_\omega(t) + K_{sa} e_\theta(t) + K_{sia} \int e_\theta(t) dt \quad (20)$$



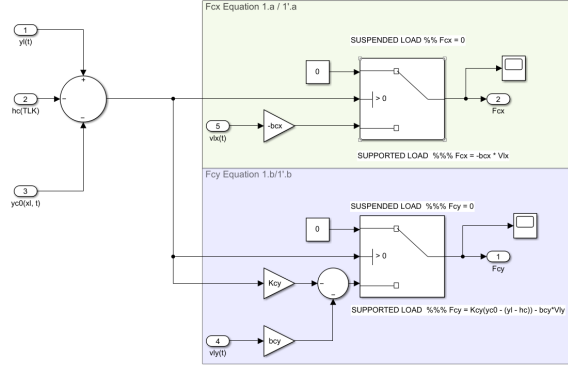


Figura 11: Modelo de Simulink de la fuerza de contacto

Por lo tanto, por Laplace:

$$T_m(s) = G(s)[b_a E_\omega(s) + K_{sa} \frac{1}{s} + K_{sia} \frac{1}{s^2}] E_\theta(s) \quad (21)$$

Donde  $G_T(s)$  es la función de transferencia del modulador de torque que, como se supone ideal, es igual a 1.

Para obtener la expresión que nos permita obtener las constante que definen al controlador se reemplaza la ecuacion 20 en la ecuacion de movimiento del izaje y del carro, se obtiene: Para el izaje, reemplazando 20 en 7 y transformandola con Laplace, se obtiene:

$$M_{Eh} \ddot{L}_h(s) = -b_{Eh} s L_h(s) - \frac{i_h}{r_{hd}} [G(s)[b_a E_\omega(s) + K_{sa} \frac{1}{s} + K_{sia} \frac{1}{s^2}] E_\theta(s)] + F_{hw}(s) \quad (22)$$

despejando

### 2.2.1. Control del Carro

Segun el modelo del sistema, la ecuacion de movimiento del carro es:

$$M_t \ddot{x}_t(t) = -b_E \dot{x}_t(t) + \frac{i_t}{r_{td}} T_{tm}(t) - F_L(t) \quad (23)$$

Se desacopla el primer termino sustituyendo:

$$T_m(t) = T_{tm}^*(t) + \frac{r_{td} b_E}{i_t} \dot{x}_t(t) \quad (24)$$

Además, si no se tiene en cuenta  $F_L(t)$  se obtiene:

$$M_E \ddot{x}_t(t) = \frac{i_t}{r_{td}} T_{tm}^*(t) \quad (25)$$

Aplicando la transformada de Laplace y definiendo como ley de control un controlador PID, se obtiene:

$$M_E s^2 X_t(s) = \frac{i_t}{r_{td}} T_{tm}^*(s) \quad (26)$$

$$T_m^*(s) = G_T(s) \left( b_{at} + \frac{1}{s} k_{sat} + \frac{1}{s^2} k_{siat} \right) (v_t^*(s) - v_t(s)) \quad (27)$$

$$M_E s^2 X_t(s) = \frac{i_t}{r_{td}} G_T(s) \left( b_{at} + \frac{1}{s} k_{sat} + \frac{1}{s^2} k_{siat} \right) (v_t^*(s) - v_t(s)) \quad (28)$$

Despejando  $\frac{v_t(s)}{v_t^*(s)}$  se obtiene:

$$\frac{v_t(s)}{v_t^*(s)} = \frac{b_{at}s^2 + k_{sat}s + k_{siat}}{s^3 \frac{r_{td}}{i_t} M_E + s^2 b_{at} + s k_{sat} + k_{siat}} \quad (29)$$

Igualando el denominador con el polinomio deseado se obtiene:

$$p(s) = s^3 + \omega_n \eta s^2 + \omega_n^2 \eta s + \omega_n^3 \quad (30)$$

$$\begin{cases} \omega_n \eta = \frac{b_{at} i_t}{r_{td} M_E} \\ \omega_n^2 \eta = \frac{k_{sat} i_t}{r_{td} M_E} \\ \omega_n^3 = \frac{k_{siat} i_t}{r_{td} M_E} \end{cases} \quad (31)$$

Entonces, las constantes del controlador PID son:

$$\begin{cases} b_{at} = \frac{r_{td} M_E \omega_n \eta}{i_t} \\ k_{sat} = \frac{r_{td} M_E \omega_n^2 \eta}{i_t} \\ k_{siat} = \frac{r_{td} M_E \omega_n^3}{i_t} \end{cases} \quad (32)$$

Los parámetros  $\omega_n$  y  $\eta$  se definen de tal forma que el sistema sea estable con el control de balanceo.

Finalmente, se observa en la figura 12 la implementación del controlador en Simulink. Se realizó una discretización de tiempo aplicando la integral por el método de trapecios.

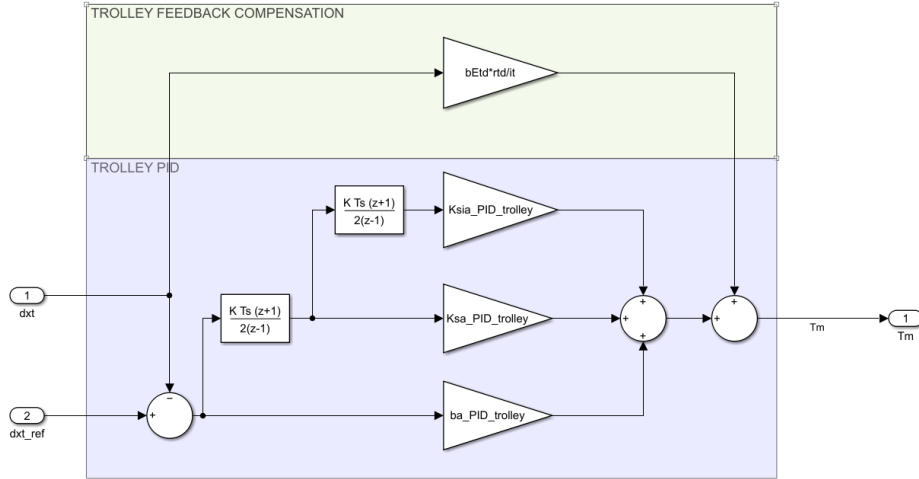


Figura 12: Controlador PID para el carro de la grúa

### 2.2.2. Control del Izaje

De forma similar al control del carro, se obtiene la ecuación dinámica para el control del izaje:

$$M_{Eh} \ddot{l}_h(t) = -b_{Eh} \dot{l}_h(t) - \frac{i_h}{r_{hd}} T_{hm}(t) \quad (33)$$

La ley de control PID en el dominio de Laplace es:

$$T_{hm}(s) = G_T(s) \left( b_{ah} + \frac{k_{sah}}{s} + \frac{k_{siah}}{s^2} \right) (v_h^*(s) - v_h(s)) \quad (34)$$

Aplicando la transformada de Laplace y despejando  $\frac{v_h(s)}{v_h^*(s)}$ , se obtiene:

$$\frac{v_h(s)}{v_h^*(s)} = \frac{b_{ah}s^2 + k_{sah}s + k_{siah}}{M_{Eh} \frac{r_{hd}}{i_h} s^3 + b_{ah}s^2 + k_{sah}s + k_{siah}} \quad (35)$$

Igualando el denominador con el polinomio característico deseado:

$$s^3 + 2\zeta\omega_n s^2 + \omega_n^2 s + \omega_n^3 = 0 \quad (36)$$

Se obtienen las siguientes ecuaciones de comparación:

$$\begin{cases} 2\zeta\omega_n = \frac{b_{ah}i_h}{r_{hd}M_{Eh}} \\ \omega_n^2 = \frac{k_{sah}i_h}{r_{hd}M_{Eh}} \\ \omega_n^3 = \frac{k_{siah}i_h}{r_{hd}M_{Eh}} \end{cases} \quad (37)$$

Finalmente, las constantes del controlador PID son:

$$\begin{cases} b_{ah} = \frac{2\zeta\omega_n r_{hd}M_{Eh}}{i_h} \\ k_{sah} = \frac{\omega_n^2 r_{hd}M_{Eh}}{i_h} \\ k_{siah} = \frac{\omega_n^3 r_{hd}M_{Eh}}{i_h} \end{cases} \quad (38)$$

Finalmente, se observa en la figura 13 la implementación del controlador en Simulink. Al igual que en el caso del carro, se realizó una discretización de tiempo aplicando la integral por el método de trapecios.

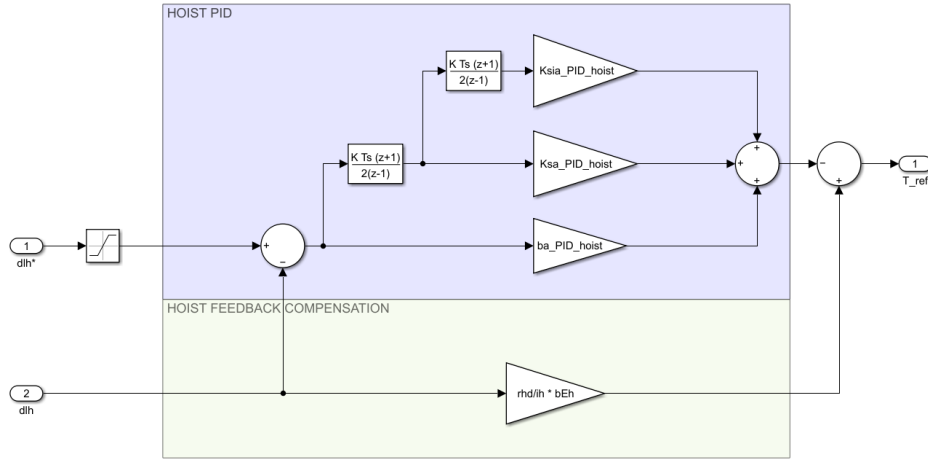


Figura 13: Controlador PID para el izaje de la grúa

### 2.2.3. Control de balanceo

A continuación se derivan las ecuaciones que modelan el sistema carro-pendulo. Se utilizará el metodo de Lagrange definiendo las cordenadas generalizadas  $x_t$  y  $\theta$ . Donde  $x_t$  es la posición del carro y  $\theta$  es el angulo del pendulo respecto a la vertical. A modo de simplificaion se toma  $l$  como un parametro y no como una funcion del tiempo. El sistema se modela siguiendo el modelo físico de la figura 3 del enunciado.

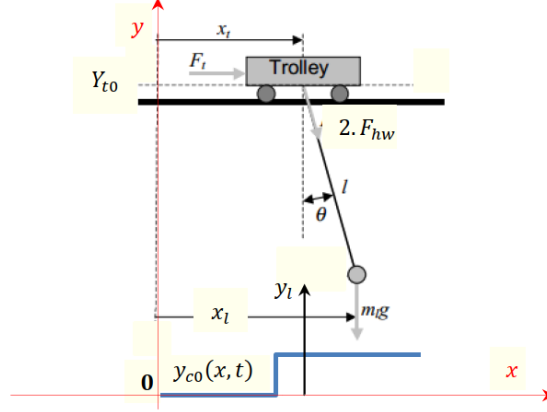


Figura 14: Modelo físico simplificado del subsistema Carro – Cable – Carga y Perfil de Obstáculos

$$K = K_t + K_{lx} + K_{ly} \quad (39)$$

$$x_l = x_t + l \sin \theta \quad (40)$$

$$\dot{x}_l = \dot{x}_t + l \cos \theta \dot{\theta} \quad (41)$$

$$y_l = Y_{t0} - l \cos \theta \quad (42)$$

$$\dot{y}_l = -l \sin \theta \dot{\theta} \quad (43)$$

$$K = \frac{1}{2} m_t \dot{x}_t^2 + \frac{1}{2} m_l \dot{x}_l^2 + \frac{1}{2} m_l \dot{y}_l^2 \quad (44)$$

$$K = \frac{1}{2} m_t \dot{x}_t^2 + \frac{1}{2} m_l (\dot{x}_t + l \cos \theta \dot{\theta})^2 + \frac{1}{2} m_l (-l \sin \theta \dot{\theta})^2 \quad (45)$$

$$K = \frac{1}{2} m_t \dot{x}_t^2 + \frac{1}{2} m_l (\dot{x}_t^2 + l^2 \cos^2 \theta \dot{\theta}^2 + 2l \dot{x}_t \cos \theta \dot{\theta}) + \frac{1}{2} m_l l^2 \sin^2 \theta \dot{\theta}^2 \quad (46)$$

$$U = -m_l g l \cos \theta \quad (47)$$

$$L = K - U \quad (48)$$

$$L = \frac{1}{2} m_t \dot{x}_t^2 + \frac{1}{2} m_l (\dot{x}_t^2 + l^2 \cos^2 \theta \dot{\theta}^2 + 2l \dot{x}_t \cos \theta \dot{\theta}) + \frac{1}{2} m_l l^2 \sin^2 \theta \dot{\theta}^2 + m_l g l \cos \theta \quad (49)$$

$$L = \frac{1}{2} m_t \dot{x}_t^2 + \frac{1}{2} m_l \dot{x}_t^2 + \frac{1}{2} m_l l^2 \dot{\theta}^2 + m_l \dot{x}_t l \cos \theta \dot{\theta} + m_l g l \cos \theta \quad (50)$$

Se define el sistema de ecuaciones de Euler-Lagrange:

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = Q_i \quad (51)$$

Para  $q_i = x_t$ :

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{x}_t} \right) - \frac{\partial L}{\partial x_t} = Q_t \quad (52)$$

$$\frac{\partial L}{\partial \dot{x}_t} = (m_t + m_l) \dot{x}_t + m_l l \cos \theta \dot{\theta} \quad (53)$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{x}_t} \right) = (m_t + m_l) \ddot{x}_t + m_l l \cos \theta \ddot{\theta} - m_l l \sin \theta \dot{\theta}^2 \quad (54)$$

$$\frac{\partial L}{\partial x_t} = 0 \quad (55)$$

Entonces:

$$(m_t + m_l)\ddot{x}_t + m_l l \cos \theta \ddot{\theta} - m_l l \sin \theta \dot{\theta}^2 = F_t(t) - b_{eqt} \dot{x}_t \quad (56)$$

Para  $q_i = \theta$ :

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = Q_\theta \quad (57)$$

$$\frac{\partial L}{\partial \dot{\theta}} = m_l \dot{x}_t l \cos \theta + m_l l^2 \dot{\theta} \quad (58)$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}} \right) = m_l \ddot{x}_t l \cos \theta - m_l \dot{x}_t l \sin \theta \dot{\theta} + m_l l^2 \ddot{\theta} \quad (59)$$

$$\frac{\partial L}{\partial \theta} = -m_l \dot{x}_t l \sin \theta \dot{\theta} - m_l g l \sin \theta \quad (60)$$

Entonces:

$$m_l \ddot{x}_t l \cos \theta - m_l \dot{x}_t l \sin \theta \dot{\theta} + m_l l^2 \ddot{\theta} + m_l \dot{x}_t l \sin \theta \dot{\theta} + m_l g l \sin \theta = 0 \quad (61)$$

Finalmente, el sistema de ecuaciones que define el modelo del sistema carro-pendulo es:

$$\begin{cases} (m_t + m_l)\ddot{x}_t + m_l l \cos \theta \ddot{\theta} - m_l l \sin \theta \dot{\theta}^2 = F_t(t) - b_{eqt} \dot{x}_t \\ m_l \ddot{x}_t l \cos \theta - m_l \dot{x}_t l \sin \theta \dot{\theta} + m_l l^2 \ddot{\theta} + m_l \dot{x}_t l \sin \theta \dot{\theta} + m_l g l \sin \theta = 0 \end{cases} \quad (62)$$

$$\boxed{\begin{cases} (m_t + m_l)\ddot{x}_t + m_l l \cos \theta \ddot{\theta} - m_l l \sin \theta \dot{\theta}^2 = 0 \\ \ddot{x}_t \cos \theta + l \ddot{\theta} + g \sin \theta = 0 \end{cases}} \quad (63)$$

Para representar lo en el espacio de estados se definen las siguientes variables de estado  $x$  y entradas  $u$ :

$$x = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} \quad (64)$$

$$u = \ddot{x}_l \quad (65)$$

Entonces, se obtiene el siguiente modelo en el espacio de estados no lineal:

$$\begin{cases} \dot{x} = f(x, u, t); x(0) = x_0 \\ y = h(x, u, t) \end{cases} \quad (66)$$

donde:

$$f(x, u, t) = \begin{bmatrix} \dot{\theta} \\ -\frac{(m_t + m_l)\ddot{x}_t}{m_l l \cos \theta} + \tan \theta \dot{\theta}^2 \end{bmatrix} \quad (67)$$

$$h(x, u, t) = -\frac{l \ddot{\theta}}{\cos \theta} - g \tan \theta \quad (68)$$

Realizando la linealización del sistema en el punto de equilibrio  $\theta = \theta^*$ ,  $\dot{\theta} = \dot{\theta}^*$  y  $\ddot{x}_t = \ddot{x}_t^*$  se obtiene el siguiente modelo linealizado:

$$\begin{cases} \dot{z} = Az + Bu \\ y = Cz + Dv \end{cases} \quad (69)$$

Donde  $z = x - x^*$ ,  $u = \ddot{x}_t - \ddot{x}_t^*$ ,  $y = \theta - \theta^*$  y  $v = \dot{\theta} - \dot{\theta}^*$ .

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{(m_t + m_l)\ddot{x}_t \tan \theta}{m_l l \cos \theta} + \frac{\dot{\theta}^2}{\cos^2 \theta} & 2 \tan \theta \dot{\theta} \end{bmatrix} \quad (70)$$

$$B = \begin{bmatrix} 0 \\ -\frac{1}{m_l l \cos \theta} \end{bmatrix} \quad (71)$$

#### 2.2.4. Control de balanceo

$$U = \ddot{X}_t = (k_d s + k_p) \frac{1}{s} (\dot{\Theta}^* - \dot{\Theta}) \quad (72)$$

$$U = \ddot{X}_t = \left(k_d + \frac{k_p}{s}\right) (\dot{\Theta}^* - \dot{\Theta}) \quad (73)$$

$$s^2 \Theta = A_{21} \Theta + B_2 \left(k_d + \frac{k_p}{s}\right) (\dot{\Theta}^* - \dot{\Theta}) \quad (74)$$

$$s \dot{\Theta} = A_{21} \frac{\dot{\Theta}}{s} + B_2 k_d \dot{\Theta}^* - B_2 k_d \dot{\Theta} + B_2 k_p \frac{\dot{\Theta}^*}{s} - B_2 k_p \frac{\dot{\Theta}}{s} \quad (75)$$

$$s^2 \dot{\Theta} = A_{21} \dot{\Theta} + s B_2 k_d \dot{\Theta}^* - s B_2 k_d \dot{\Theta} + B_2 k_p \dot{\Theta}^* - B_2 k_p \dot{\Theta} \quad (76)$$

$$\dot{\Theta}(s^2 - A_{21} + s B_2 k_d + B_2 k_p) = \dot{\Theta}^*(s B_2 k_d + B_2 k_p) \quad (77)$$

$$G(s) = \frac{\dot{\Theta}}{\dot{\Theta}^*} = \frac{s B_2 k_d + B_2 k_p}{s^2 + s B_2 k_d + B_2 k_p - A_{21}} \quad (78)$$

$$p_r(s) = (s^2 + 2\zeta\omega_n s + \omega_n^2) \quad (79)$$

$$\begin{cases} \omega_n^2 = B_2 k_p - A_{21} \\ 2\zeta\omega_n = B_2 k_d \end{cases} \quad (80)$$

$$\begin{cases} k_p = \frac{\omega_n^2 + A_{21}}{B_2} \\ k_d = \frac{2\zeta\omega_n}{B_2} \end{cases} \quad (81)$$

#### 2.2.5. Nivel 1 - Control Supervisor

Como se mencionó antes el Nivel 1 se basa en un autómata de estados discretos activados por eventos (DEDS - Discrete Event Dynamic System). Este nivel coordina las operaciones de la grúa portuaria gestionando estados lógicos y comandos, garantizando transiciones seguras y eficientes entre modos operativos. Este control opera mediante un autómata secuencial, que gestiona las transiciones entre diferentes modos de funcionamiento. Además, se estructura de forma jerárquica o concurrente para coordinar varios componentes de manera eficiente.

Entre sus funciones de supervisión, monitorea los comandos emitidos por el operador y las señales provenientes de sensores. También se encarga de gestionar los límites operativos normales, tales como posiciones máximas de movimiento y velocidades permitidas, asegurando que la operación se mantenga dentro de parámetros seguros. El Nivel 1 optimiza las trayectorias del movimiento para garantizar una operación eficiente y sin interrupciones. Gestiona la coordinación de frenos y el uso de *twistlocks* para evitar errores operativos. Además, permite realizar transiciones suaves entre los modos manual y automático sin detener la operación, maximizando así la eficiencia.

A continuación, se presenta la lógica de funcionamiento de este nivel utilizando *Stateflow* para modelar el autómata de estados. El modelo se organiza en varios estados que se ejecutan en **paralelo**. A continuación, se describen los estados más relevantes.

**Principal:** El estado se encarga de gestionar las consignas de referencia de cada eje según el modo en el que se encuentre. En la figura 15 se observan cuatro posibles estados: *Manual*, *Automatic*, *Homing* y *MassEstimation*.

En el **modo manual**, el operador controla directamente la velocidad de los motores, y la referencia de velocidad se calcula en función del control analógico del operario.

En el **modo automatic**, el sistema sigue la trayectoria generada por el módulo de planificación de trayectorias. Estas trayectorias se calculan cuando el sistema entra en modo automático, utilizando la función:

```
traj_gen(obstacle_profile, [x0, y0], [xf, yf], vy0, ay0, vy_max, vyf, ayf);
```

Esta función genera una matriz que contiene la posición, velocidad y aceleración de cada eje en función del tiempo. Se dará más detalle sobre esta función en la sección de generación de trayectorias 2.3. El seguimiento de la trayectoria generada se realiza de forma tal que no se vea afectada por diferencias entre la planificación y el solver o PLC en el paso de tiempo.

En el **modo homing**, el sistema lleva los motores a una posición inicial conocida, siguiendo la secuencia que se observa en la figura 15.

Por último, en el **modo MassEstimation**, el sistema calcula la masa de la carga. El calculo se realiza en otro estado.

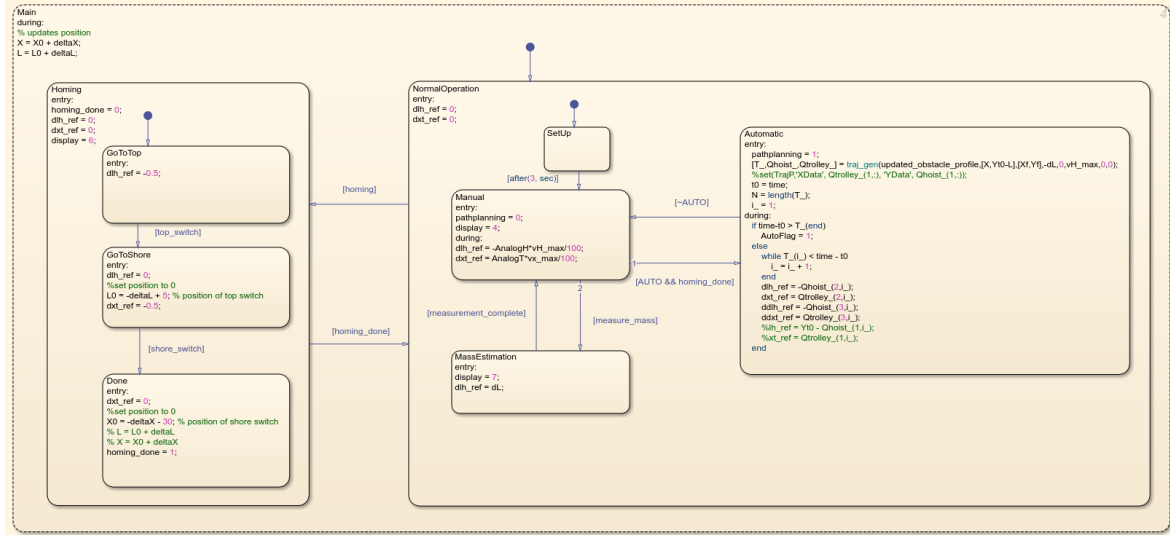


Figura 15: Estado principal del autómata de estados.

**Ejes de operación:** Estos estados gestionan de manera individual los movimientos del carro y el izaje, estableciendo la consigna de velocidad correspondiente según el estado operativo del sistema. En modo automático, el sistema sigue la trayectoria generada por el módulo de planificación de trayectorias. En modo manual, la consigna de velocidad es definida directamente por el operador.

Para garantizar transiciones suaves entre consignas, se utiliza la función `vel_prof_gen(v0, vf, a0, v_max, a_max, j)`, que genera un perfil de aceleración trapezoidal. Este perfil se recalcula automáticamente cada vez que la variación en la consigna de velocidad supera el 1%, asegurando así una operación suave. En la figura 16 se observa la implementación de lo mencionado en *Stateflow*.

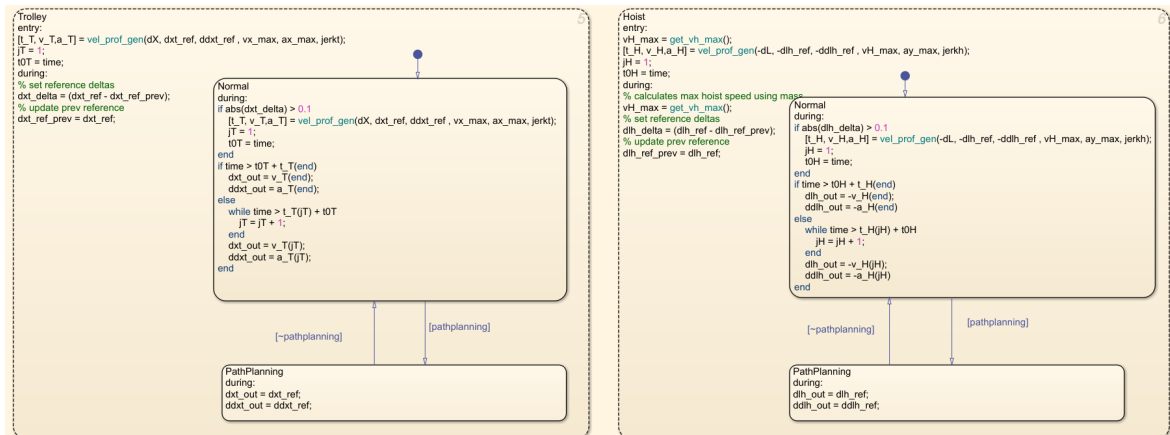


Figura 16: Estado de los ejes de operación del autómata de estados.

**Secuencial automáticas:** Este estado se encarga de gestionar las secuencias de movimientos de la grúa definidas por el operador en la HMI. En la figura 17 se muestra la implementación de este

estado en *Stateflow*, donde se definen cuatro estados: *DropBay*, *PickUpBay*, *DropShip* y *PickUpShip*. Cada uno se encarga de gestionar la secuencia correspondiente a un movimiento específico.

En cada estado se sigue una secuencia definida. Primero, se espera que el sistema salga de la zona manual para iniciar la trayectoria automática. A continuación, si se ha tomado un contenedor, se realiza la estimación de masa. Luego, se define la posición final deseada y se activa el modo automático para ejecutar el movimiento. Finalmente, el sistema vuelve a la zona manual para dejar o recoger un contenedor.



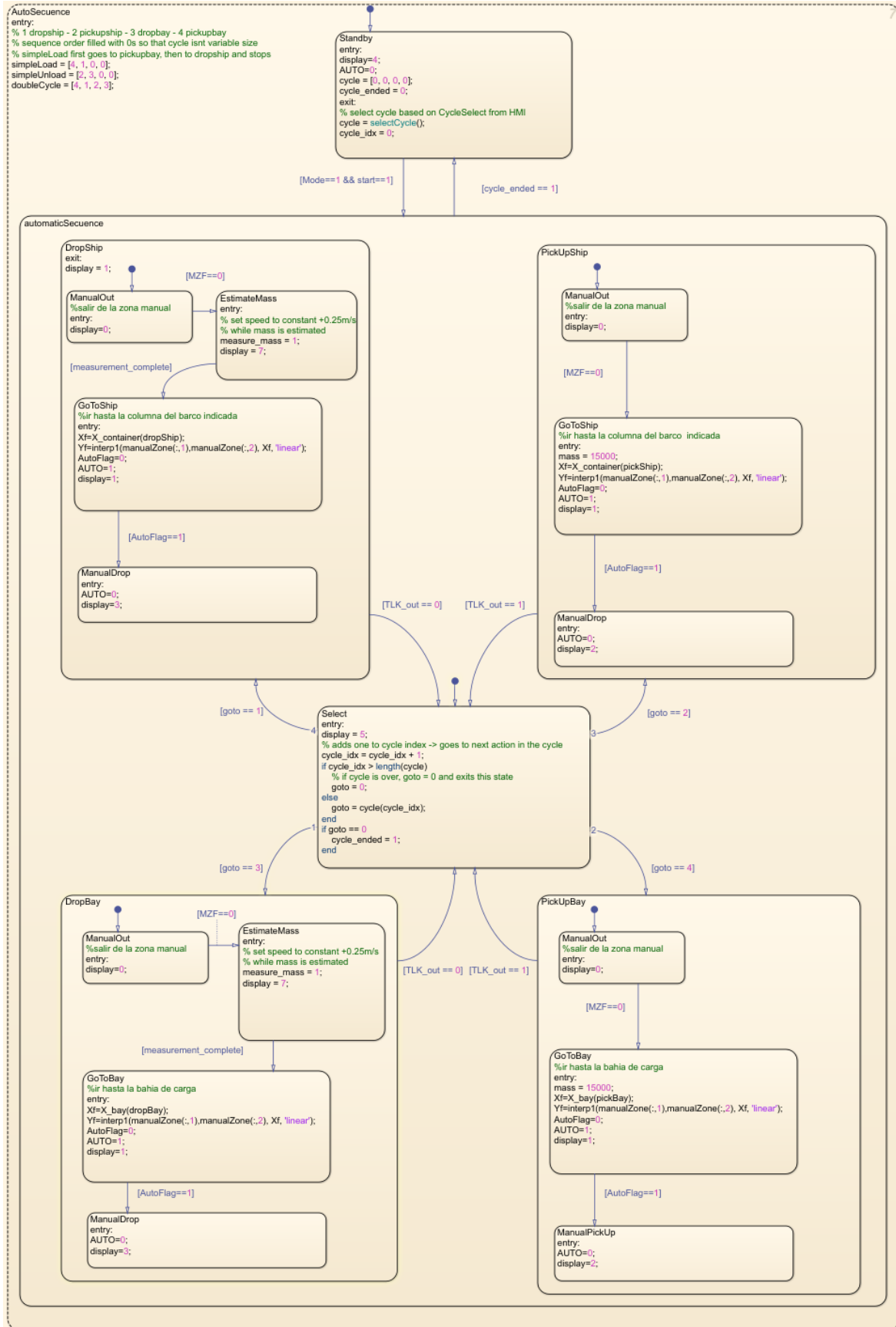


Figura 17: Estado Secuencia automática del autómata de estados.

**Estimación de masa:** El estado se encarga de calcular la masa del *spreader* + carga. Para ello, utiliza la señal de una celda de carga que mide la tensión del cable de izaje. La masa se calcula realizando una estimación en función de la tensión medida. En la figura 18 se muestra la implementación de este estado en *Stateflow*. El algoritmo se ejecuta de forma recursiva hasta que la masa estimada converge a un valor estable. Para calcular la masa en función de la tensión del cable, se utiliza la ecuación 82:

$$M_{est} = \frac{T}{g - a_y} \quad (82)$$

Donde  $T$  es la tensión medida en la celda de carga,  $g$  es la aceleración de la gravedad y  $a_y$  es la aceleración del sistema en la dirección vertical. Se tiene en cuenta la aceleración vertical porque la estimación se realiza mientras la carga está en movimiento acelerado.

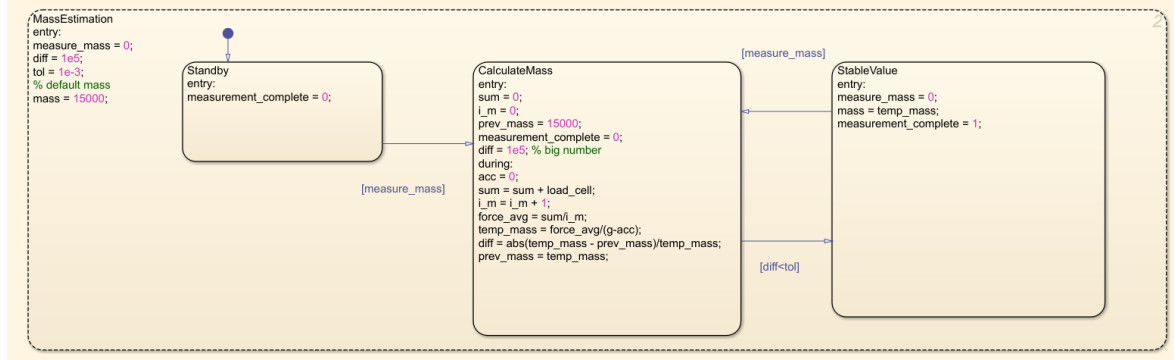


Figura 18: Estado de estimación de masa del autómata de estados.

**Perfil de obstáculos:** El estado *ReadObstacleProfile* se encarga de actualizar el perfil de obstáculos utilizando datos obtenidos del sensor LiDAR y de generar el perfil de obstáculos en función de esta información.

Al iniciar, el perfil de obstáculos y la zona manual se inicializan con la misma información. Durante la ejecución, si la etapa de *homing* ha finalizado y no hay una señal activa de bloqueo (*TLK\_out*), se busca el punto más cercano en el perfil de obstáculos a la posición actual en el eje  $X$ . Luego, se actualiza el valor correspondiente en el eje  $Y$  utilizando la información del sensor LiDAR. Finalmente, se genera la nueva zona manual mediante la función *ObsToManualZone()*, basada en el perfil de obstáculos actualizado.

En la figura 19 se observa la implementación de este estado en *Stateflow*.

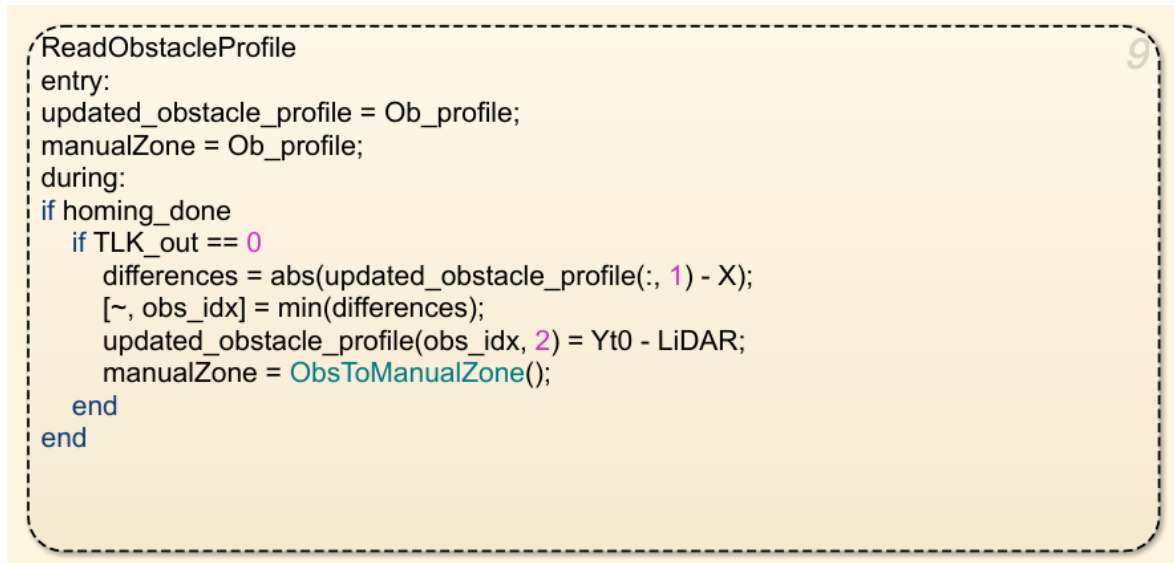


Figura 19: Estado de perfil de obstáculos del autómata de estados.

**Detección de zona manual:** El estado *ManualZoneDetection* se encarga de detectar si la carga se encuentra dentro o fuera de la zona manual, utilizando una bandera llamada MZF. Esta bandera se inicializa en 1, indicando que la carga está dentro de la zona manual, y se ajusta dinámicamente según la posición de la carga.

El estado emplea un mecanismo de histéresis para evitar cambios rápidos entre estados. A través de una interpolación lineal, compara la posición actual de la carga con los límites de la zona manual definidos en `manualZone`. Si la carga cruza estos límites, se actualiza la bandera MZF, permitiendo al sistema identificar cuándo pasar del control manual al automático, o viceversa. En la figura 20 se observa la implementación de este estado en *Stateflow*.

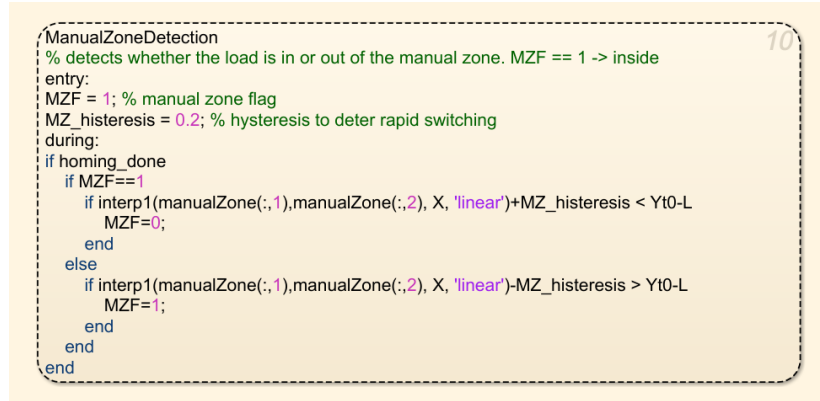


Figura 20: Estado de detección de zona manual del autómeta de estados.

### 2.3. Generación de trayectorias

La generación de trayectorias se implementa cuando la grúa opera en modo automático. Su función es definir un perfil que posicione a la grúa en una ubicación  $xy$  determinada de manera suave y eficiente. Para ello, se genera una matriz que describe la posición, velocidad y aceleración de cada motor en función del tiempo, utilizando un intervalo de tiempo  $dt$ . La trayectoria considera la posición actual de la grúa y la columna de contenedores o bahía de carga de destino. Además, se tiene en cuenta el perfil de obstáculos y se establece una distancia de seguridad en las direcciones  $x$  e  $y$ , definiendo así una zona de seguridad en la que el modo automático no puede operar.

En primer lugar, se definen los puntos significativos de la trayectoria: la posición inicial, la posición final y la altura máxima. Estos puntos se calculan considerando el perfil de obstáculos previamente mencionado.

Posteriormente, se calculan los perfiles de subida, bajada y traslación. Estos perfiles se ajustan teniendo en cuenta los estados inicial y final deseados, asegurando así que el movimiento sea suave. Los perfiles se determinan utilizando un perfil de aceleración trapezoidal, que incluye una sobreaceleración para lograr una trayectoria más fluida y eficiente. La figura 21 ilustra un ejemplo de un perfil de aceleración trapezoidal similar al que se utiliza. Se indican siete tiempos que definen el perfil necesario para alcanzar un estado deseado. En este perfil, se observa que tanto la velocidad como la aceleración inicial son distintas de cero, y el perfil se ajusta a estos valores de manera que la transición entre el estado inicial y el perfil sea suave.

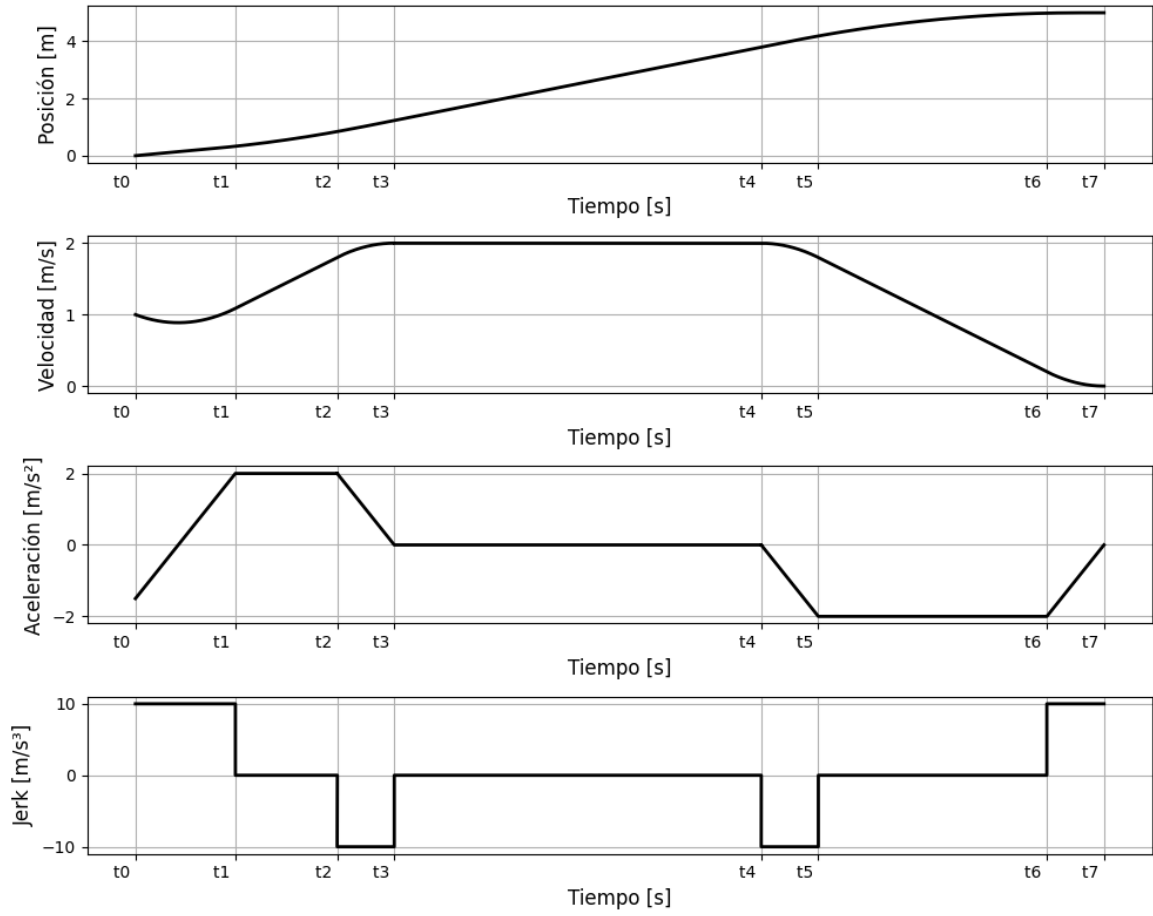


Figura 21: Perfil de aceleración trapezoidal con condiciones iniciales  $x_0 = 0$  m,  $v_0 = 1$  m/s y  $a_0 = -1,5$  m/s<sup>2</sup>.

Es importante destacar que la generación de la trayectoria admite casos degenerados, lo que contribuye a la robustez del sistema. Se aceptan situaciones en las que la distancia recorrida es menor que la requerida para alcanzar la aceleración o velocidad máxima. En estos casos, se ajustan los perfiles utilizando un algoritmo recursivo para adaptar el perfil trapezoidal a uno triangular, como se muestra en la figura 22.

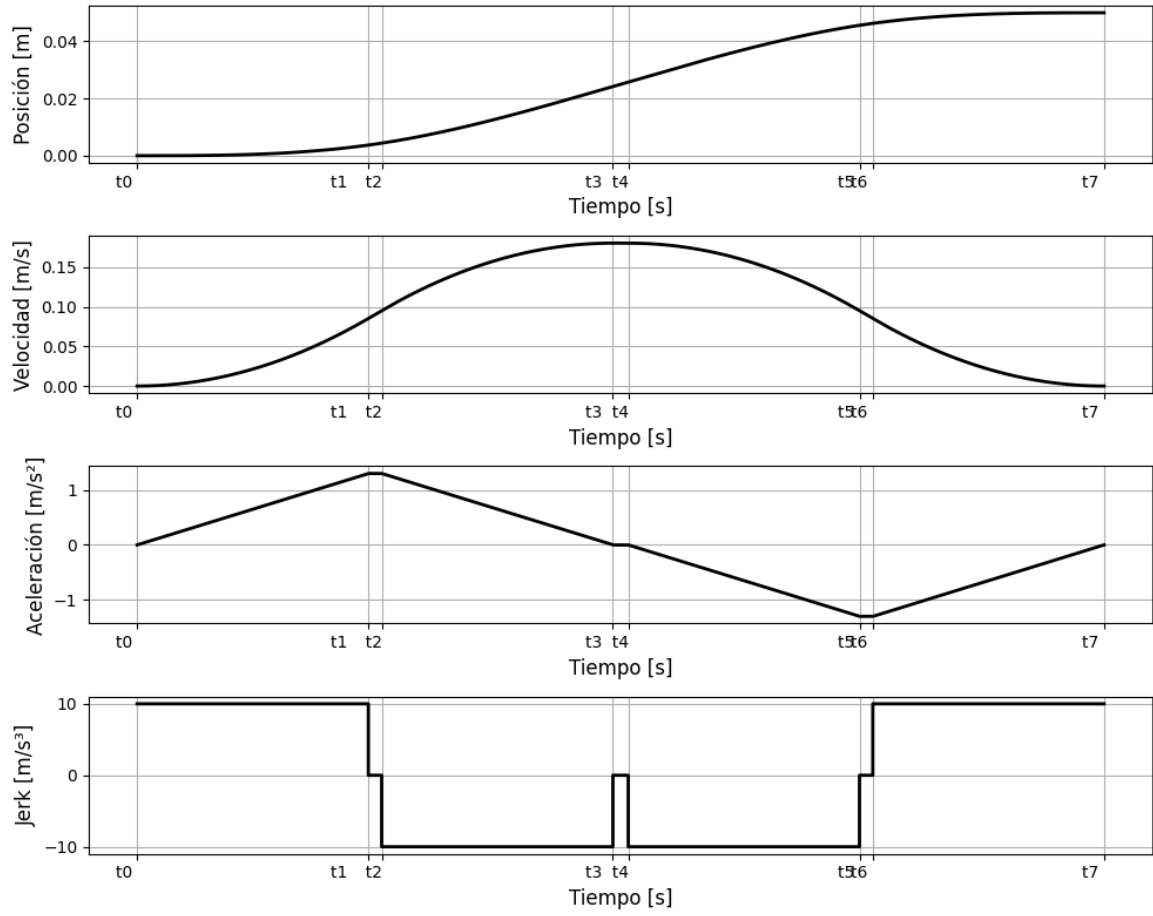


Figura 22: Perfil de aceleración en un caso degenerado.

Con los cálculos de estos perfiles, se realiza un análisis del tiempo que toma cada uno de los movimientos. En función de estos tiempos y del perfil de obstáculos, se determina un solapamiento en los movimientos, de modo que la grúa no se detenga en ningún momento, acortando así los tiempos de translación. La figura 23 muestra un ejemplo de la trayectoria junto con el perfil de obstáculos.

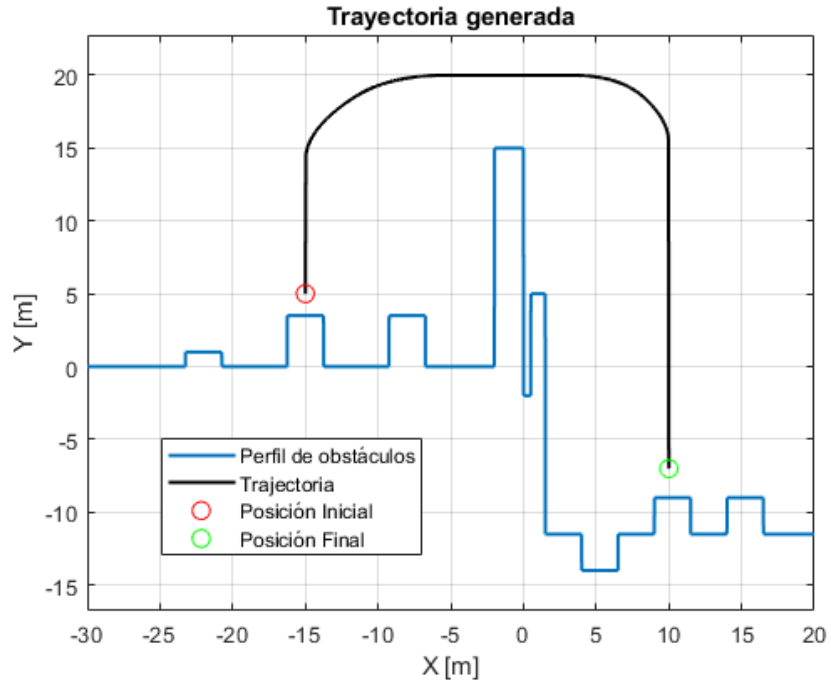


Figura 23: Trayectoria de la carga.

La sincronización del movimiento y la suavidad del desplazamiento se pueden corroborar en las figuras 24 y 25.

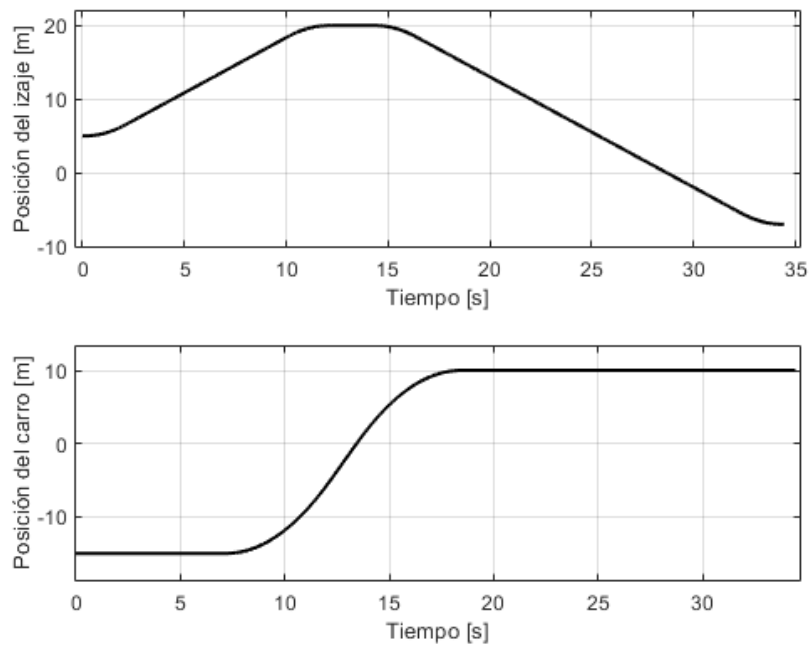


Figura 24: Posición del carro e izaje.

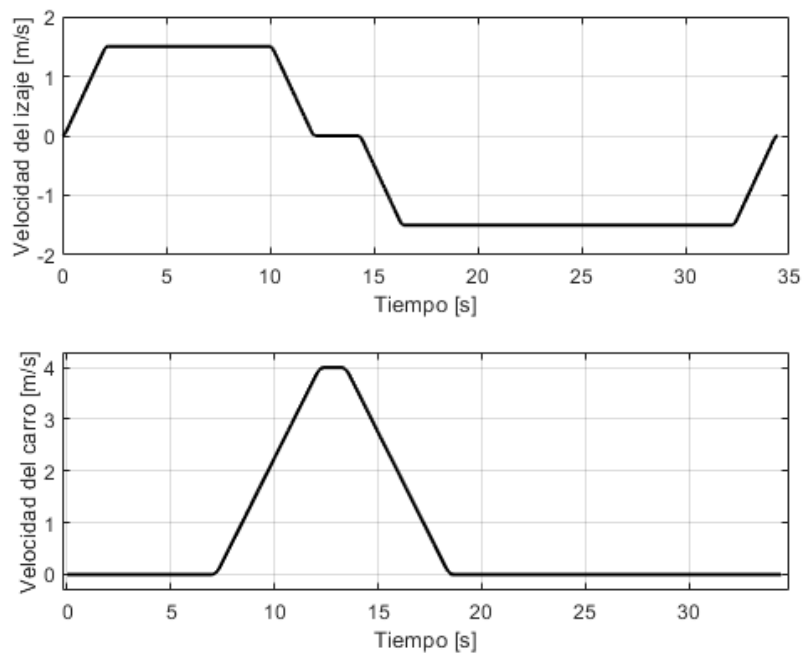


Figura 25: Velocidad del carro e izaje.

### 2.3.1. Nivel 0 – Seguridad y Protección

El **Nivel 0** se encarga de la seguridad y protección del sistema, actuando ante situaciones de emergencia para llevar la grúa a un estado seguro. Este nivel funciona de manera independiente al control operativo normal y toma el mando en caso de fallas críticas o riesgos para la seguridad. El autómata de este nivel supervisa eventos como la activación de pulsadores de emergencia, la detección de sobrevelocidad, y la llegada a los límites de carrera finales. En tales casos, se interrumpe la operación normal y se ejecutan acciones predeterminadas para detener la grúa de manera controlada y segura.

El objetivo del Nivel 0 es minimizar riesgos tanto para los operadores como para la integridad del equipo. Entre las acciones que puede realizar están la activación de frenos de emergencia, el apagado de motores, y el bloqueo de cualquier maniobra hasta que se restablezca una condición segura. En caso de un evento crítico, este nivel debe llevar al sistema a un estado seguro.

A continuación, se muestra la lógica de funcionamiento de este nivel utilizando *Stateflow* para modelar el autómata de estados. El modelo se compone de dos estados que operan en paralelo. A continuación, se describen estos estados.

**Seguridad:**

**Watchdog:**

## 3. Resultados

## 4. Conclusión

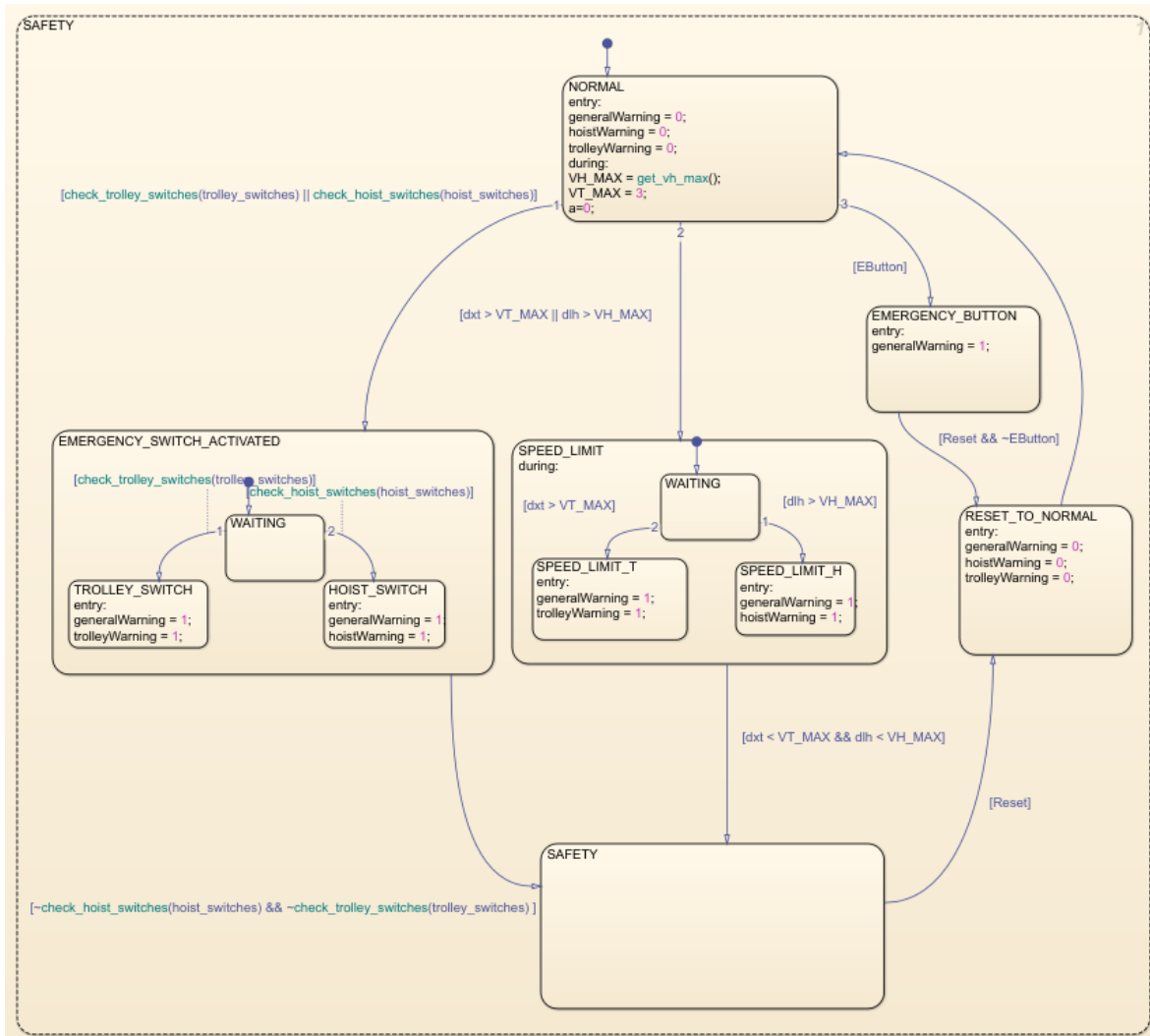


Figura 26: Estado de seguridad del autómata de estados-Nivel 0.



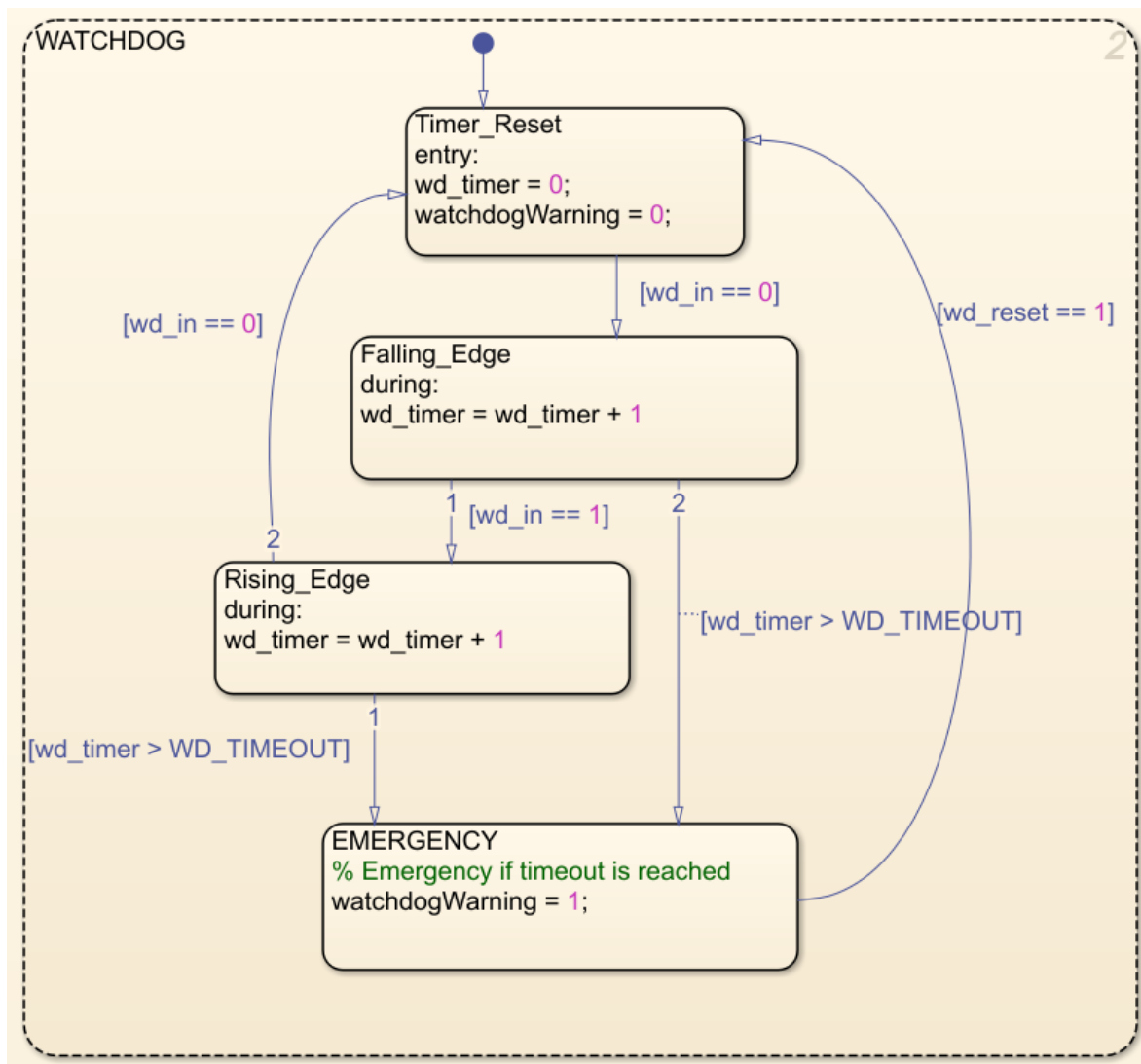


Figura 27: Estado de *Watchdog* del autómata de estados-Nivel 0.