

Proyecto Global de Asignatura

Rubén A. y David A. - Otoño 2024

```
decimales = 5;           % Decimales a mostrar
muestras_para_media = 50; % Muestras del final de las respuestas a tener en cuenta
para las medias
aproximacion = -3;       % Aproximación del  $\ln(0.05)$ 
```

Fase 1 - Caracterización de la planta

Ejercicio 1

Capture y caracterice la salida de la planta en función de la tensión de entrada con un procedimiento análogo al empleado en la Práctica 2 y obtenga su función de transferencia (FdeT) en las unidades que considere más adecuadas para trabajar en el desarrollo de todo el diseño.

```
ficheros = dir("Fase1/sin_perturbacion/");

funciones_de_transferencia = {};
Kms = {};
taus = {};
tiempos_de_interes = {};
salidas_de_interes = {};
amplitudes_escalon = {};

for i = 1:length(ficheros)
    fichero = ficheros(i).name;

    % Saltar '.' y '..'
    if strcmp(fichero, '.') || strcmp(fichero, '..')
        continue;
    end
    str_amplitud_escalon = strsplit(fichero, '_'); str_amplitud_escalon =
str_amplitud_escalon{end};
    amplitud_escalon = str2double(regexprep(str_amplitud_escalon, '^[^\d]', ''));

    escalon_sin_perturbacion = load("Fase1/sin_perturbacion/" + fichero);
    Ts = 35e-3;

    tiempo      = escalon_sin_perturbacion(:, 1);
    referencia   = escalon_sin_perturbacion(:, 2);
    salida       = escalon_sin_perturbacion(:, 3);

    x_ini = find(abs(referencia) > 0, 1, 'first') + 1; % Se suma 1 ya que se tienen
dos retardos y se va a trabajar
% con una aproximación de primer orden
    x_fin = length(referencia);
```

```

t_ini = tiempo(x_ini);
t_fin = tiempo(end);

t_interes = tiempo(x_ini:x_fin) - tiempo(x_ini);
c_interes = salida(x_ini:x_fin);

tiempos_de_interes{end + 1} = t_interes;
salidas_de_interes{end + 1} = c_interes;
amplitudes_escalon{end + 1} = amplitud_escalon;

c_inf = mean(c_interes(end-muestras_para_media, end));

ks = find(abs(c_interes) >= abs(0.95*c_inf), 1, 'first') - 1;
ts = ks*Ts;

figure
hold on
stairs(t_interes, c_interes)
plot([0 t_fin - t_ini], 0.95*c_inf*[1 1], 'r:')
xlim([0 5])
ylim(1.1 * [min(0, c_inf) max(0, c_inf)])
title("Respuesta al escalón de " + str_amplitud_escalon + " de la planta")
xlabel("Tiempo (s)")
ylabel("Velocidad escalera (m/s)")

Km = c_inf/amplitud_escalon;
tau = -ts/aproximacion;

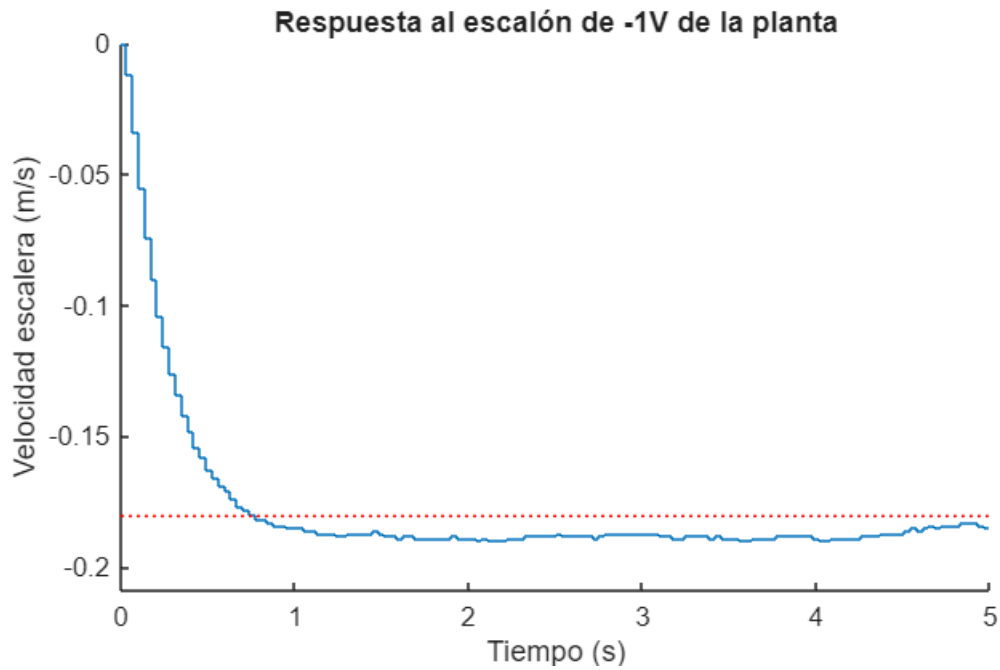
Kms{end + 1} = Km;
taus{end + 1} = tau;

syms s
funciones_de_transferencia{end + 1} = Km/(tau*s + 1);

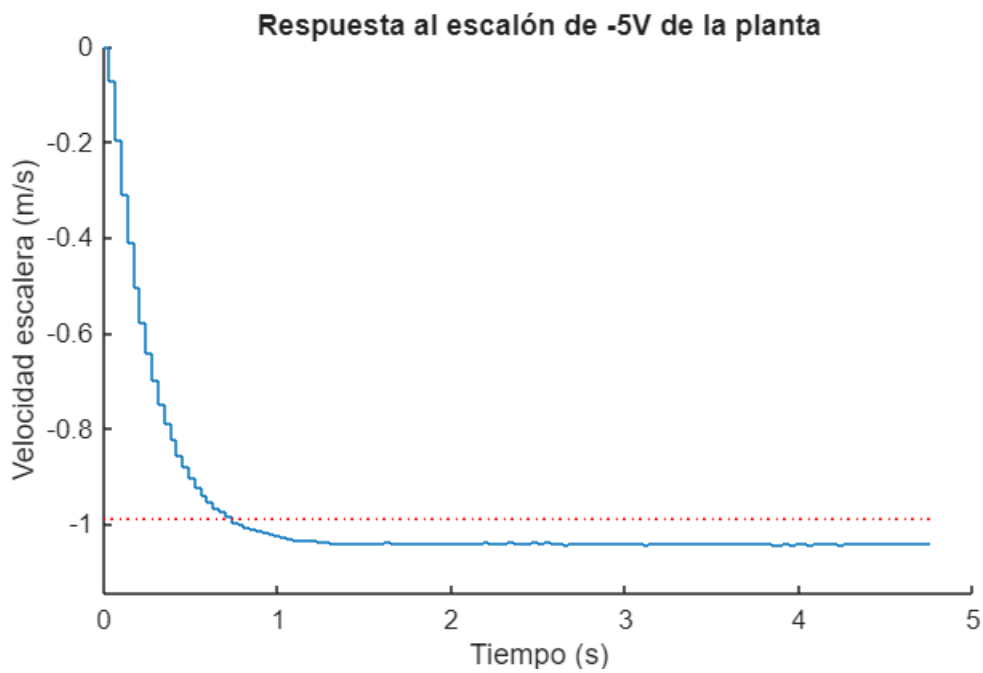
fprintf("Escalón de %d V.\n\tMedia de las últimas %d muestras = %.0" +
decimales + "f m/s\n\tts = %.0" ...
+ decimales + "f s\n\tKm = %.0" + decimales + "f m/s/V\n\ttau = %.0" +
decimales + "f s", ...
amplitud_escalon, muestras_para_media, c_inf, ts, Km, tau)

end

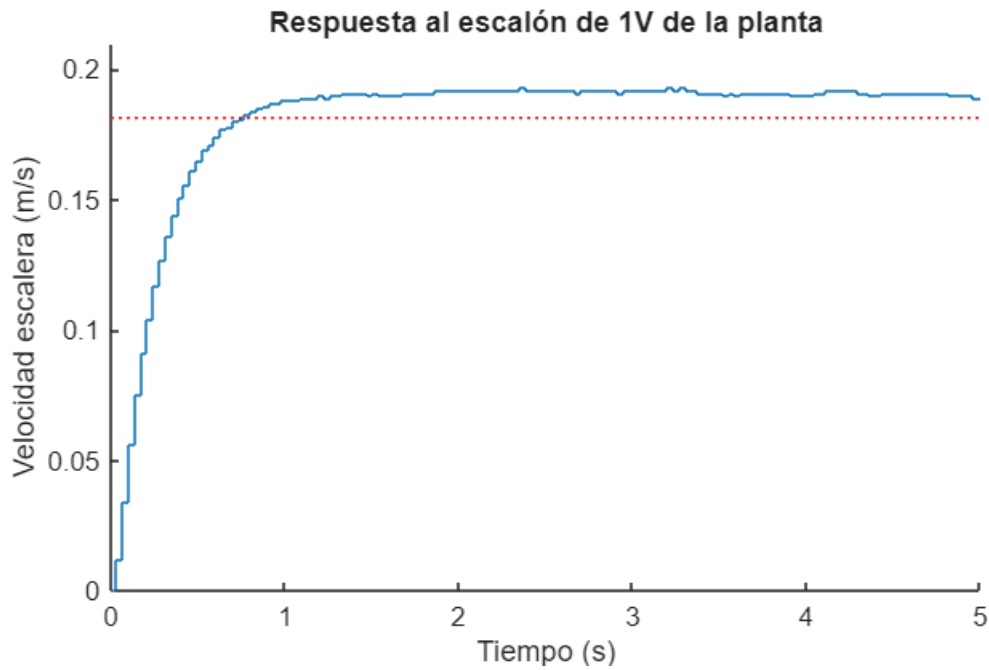
```



Escalón de -1 V.
 Media de las últimas 50 muestras = -0.19000 m/s
 $t_s = 0.77000$ s
 $K_m = 0.19000$ m/s/V
 $\tau = 0.25667$ s



Escalón de -5 V.
 Media de las últimas 50 muestras = -1.04200 m/s
 $t_s = 0.73500$ s
 $K_m = 0.20840$ m/s/V
 $\tau = 0.24500$ s



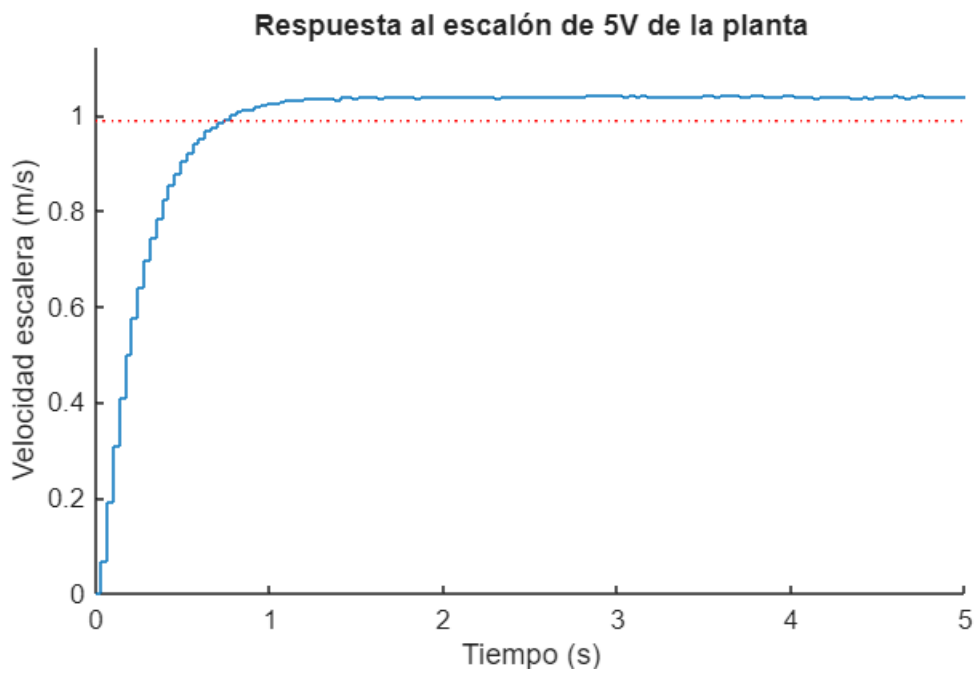
Escalón de 1 V.

Media de las últimas 50 muestras = 0.19100 m/s

$t_s = 0.77000$ s

$K_m = 0.19100$ m/s/V

$\tau = 0.25667$ s



Escalón de 5 V.

Media de las últimas 50 muestras = 1.04000 m/s

$t_s = 0.73500$ s

$K_m = 0.20800$ m/s/V

$\tau = 0.24500$ s

```
fprintf("Funciones de transferencia:")
```

Funciones de transferencia:

```
vpa(funciones_de_transferencia, decimales)
```

```
ans =
```

$$\left(\frac{0.19}{0.25667 s + 1.0} \frac{0.2084}{0.245 s + 1.0} \frac{0.191}{0.25667 s + 1.0} \frac{0.208}{0.245 s + 1.0} \right)$$

```
fprintf("Funcion de transferencia media:")
```

Funcion de transferencia media:

```
Km = mean(cell2mat(Kms));  
tau = mean(cell2mat(taus));
```

```
syms s  
G = Km/(tau*s + 1);  
vpa(G, decimales)
```

```
ans =
```

$$\frac{0.19935}{0.25083 s + 1.0}$$

```
fprintf("Valores de la función media:\n\tKm = %.0" + decimales + "f m/s/V\n\ttau = %.0" + decimales + "f s", Km, tau)
```

Valores de la función media:

```
Km = 0.19935 m/s/V  
tau = 0.25083 s
```

Ejercicio 2

Corrobore la validez del modelo mediante su simulación en Matlab/Simulink, como también se hizo en la Práctica 2

```
if ~exist("tiempos_sim", 'var')  
    open_system("Fase1/ejercicio2")  
    set_param("ejercicio2/Escalera mecánica", 'Numerator', string(Km),  
    'Denominator', '[' + string(tau) + ' 1' + ']')  
  
    sim_len = 150;  
    tiempos_sim = zeros(sim_len, 1);  
    salidas_sim = zeros(sim_len,1);  
    for i = 1:length(ficheros)  
        fichero = ficheros(i).name;  
  
        % Saltar '.' y '..'  
        if strcmp(fichero, '.') || strcmp(fichero, '..')  
            continue;  
        end  
  
        str_amplitud_escalon = strsplit(fichero, '_'); str_amplitud_escalon =  
        str_amplitud_escalon{end};
```

```

    amplitud_escalon = str2double(regexprep(str_amplitud_escalon, '^[^\d]',
''));

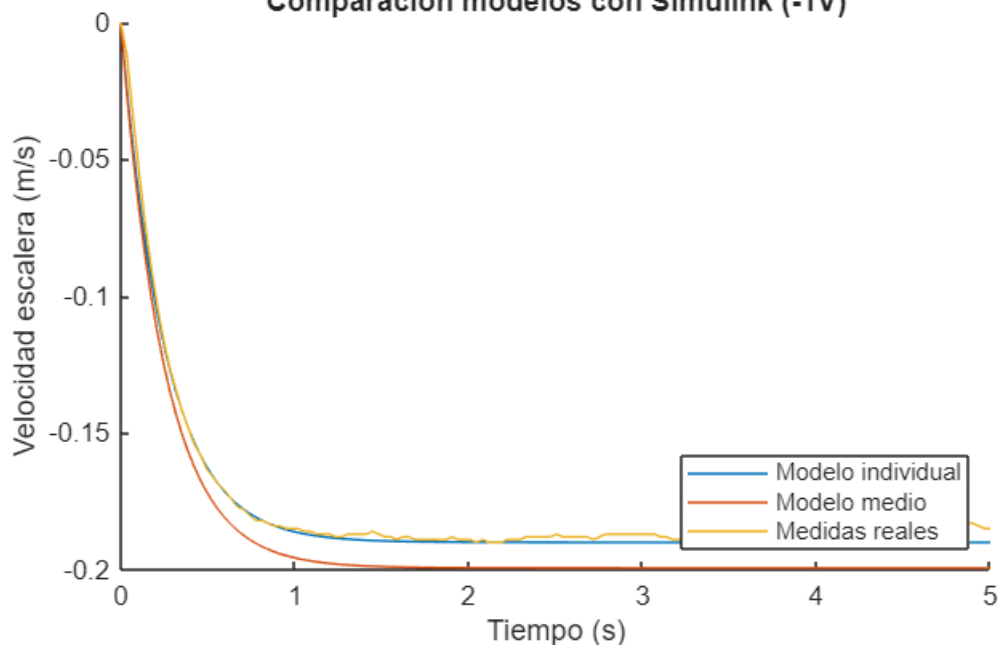
    set_param("ejercicio2/Consigna (V)", 'After', string(amplitud_escalon))
    sim("ejercicio2")

    tiempos_sim(:, i - 2) = salidaEscalonMotor(1:sim_len, 1);
    salidas_sim(:, i - 2) = salidaEscalonMotor(1:sim_len, 2);
end
end
for i = 1:(length(ficheros)-2)
    fichero = ficheros(i+2).name;
    str_amplitud_escalon = strsplit(fichero, '_'); str_amplitud_escalon =
str_amplitud_escalon{end};
    amplitud_escalon = str2double(regexprep(str_amplitud_escalon, '^[^\d]', ''));

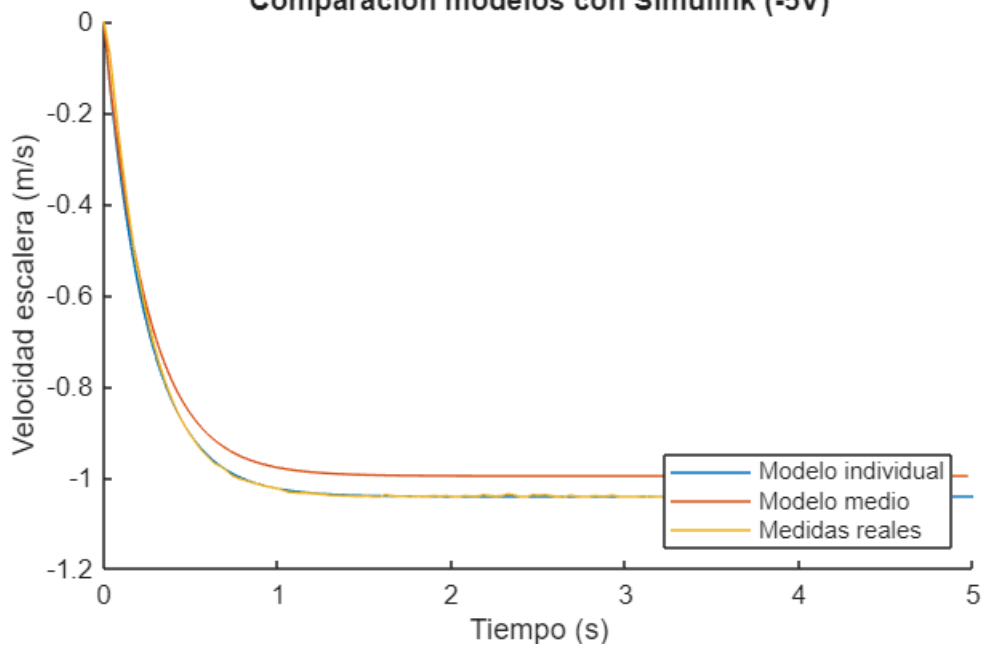
    [c_ind, t_ind] = step(amplitud_escalon * tf(Kms{i}, [taus{i} 1]), 5);
    t_sim = tiempos_sim(:,i);
    c_sim = salidas_sim(:,i);
    figure
    hold on
    title("Comparación modelos con Simulink (" + str_amplitud_escalon + ")")
    plot(t_ind, c_ind)
    plot(t_sim, c_sim)
    plot(tiempos_de_interes{i}, salidas_de_interes{i})
    legend('Modelo individual', 'Modelo medio', 'Medidas
reales', 'Location', 'southeast')
    xlim([0 5])
    xlabel("Tiempo (s)")
    ylabel("Velocidad escalera (m/s)")
end

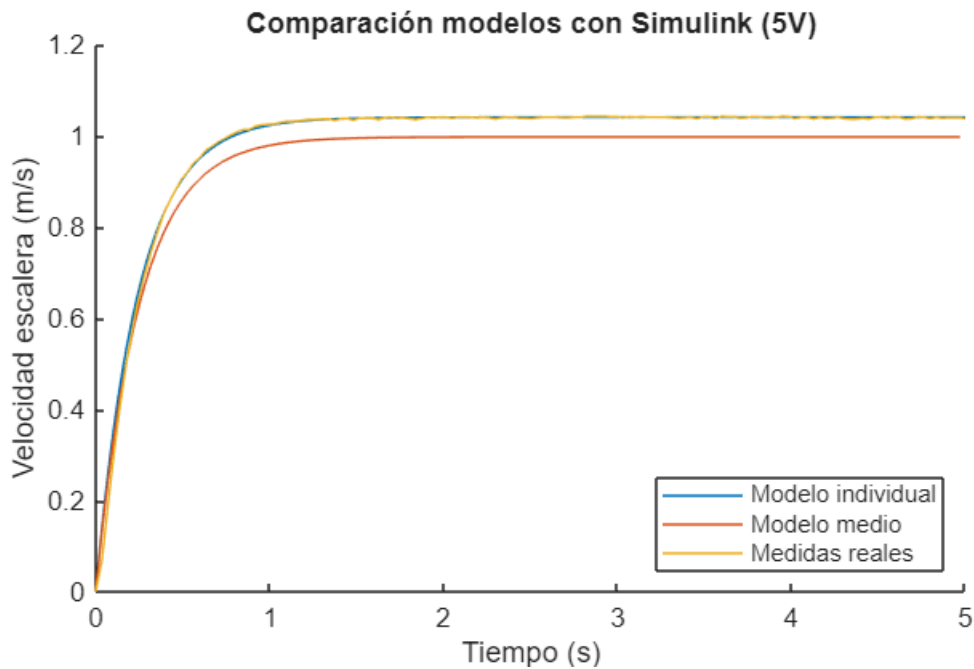
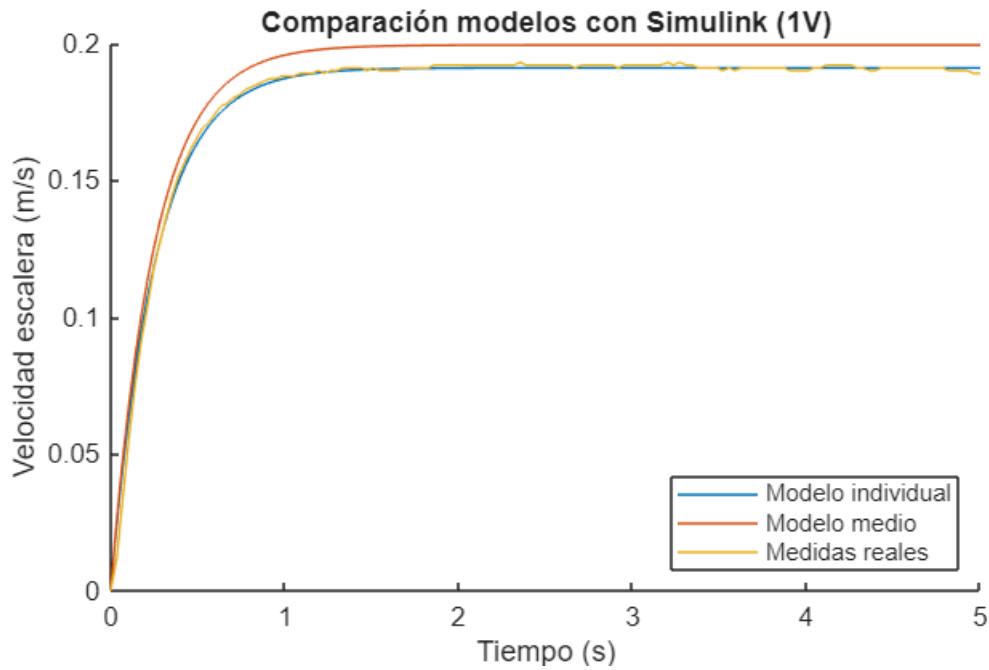
```

Comparación modelos con Simulink (-1V)



Comparación modelos con Simulink (-5V)





Ejercicio 3

Proponga e implemente los siguientes experimentos para caracterizar las alinealidades de la planta:

1. Para modelar la saturación excite el SM con grandes tensiones
2. Para modelar la zona muerta excite el SM con pequeñas tensiones.

Debe de obtener los valores adecuados con sus unidades de la saturación y zona muerta, explicando el procedimiento de obtención, y añadir al modelo de Simulink los bloques de las alinealidades caracterizadas convenientemente colocadas y con coherencia de unidades.


```
% Medida de saturación
% Se incrementa la ganancia y se pone el valor máximo de entrada.
% Se obtiene el valor máximo de salida, que resulta ser igual para ambas
direcciones.
c_max_abs = 2.11;
fprintf("Saturación. Valor máximo de módulo de salida: |c_max| = %.02f m/s",
c_max_abs)
```

Saturación. Valor máximo de módulo de salida: |c_max| = 2.11 m/s

```
% Medida de zona muerta
% Partiendo desde cero, se disminuye la ganancia y se va incrementando la entrada
gradualmente hasta que se tiene movimiento (salida != 0)
deadzone_in_pos = 0.21;
deadzone_in_neg = -0.19;
fprintf("Zona muerta. Valor mínimo de entrada. Asimétrico. DeadZone: (%.02f, %.02f)
V", deadzone_in_neg, deadzone_in_pos)
```

Zona muerta. Valor mínimo de entrada. Asimétrico. DeadZone: (-0.19, 0.21) V

```
% La zona muerta resta un valor fijo a la entrada, por lo que hay que
% compensar otra vez la Km.
```

```
% Km' = salida/(entrada - deadzone) = Km/(1-dz/entrada)
km_comps = zeros(1, length(amplitudes_escalon));
for i = 1:length(km_comps)
    if amplitudes_escalon{i} > 0
        dz = deadzone_in_pos;
    else
        dz = deadzone_in_neg;
    end

    km_comps(i) = Kms{i}/(1-dz/amplitudes_escalon{i});
end

km_comp = mean(km_comps);

fprintf("Nueva aproximación de la planta teniendo en cuenta la zona muerta [m/s/V].
Solo afecta a Km, no la tau")
```

Nueva aproximación de la planta teniendo en cuenta la zona muerta [m/s/V]. Solo afecta a Km, no la tau

```
syms s
Gcomp = km_comp/(tau*s + 1);
vpa(Gcomp, decimales)
```

```
ans =
    0.22752
-----
0.25083 s + 1.0
```

Ejercicio 4

Calcule el equivalente discreto de la planta y caracterice teóricamente su respuesta ante entrada escalón, tanto en régimen permanente como transitorio.

```
% Se añade un retraso para tener en cuenta el que se quitó en el ejercicio
% 1. Esto es debido a que la planta real es de segundo orden mientras que
% se aproxima con una de primero.

% BoG = ((1-z^(-1)) * sum_(Polos G(s)/s) Res[G(s)/(s*(1-exp(s*Ts)*z^(-1))]) * (1/z)
BoG = zpk(series(c2d(tf(Km, [tau 1]), Ts), tf(1, [1 0], Ts)))
```

```
BoG =

    0.025963
    -----
    z (z-0.8698)
```

```
Sample time: 0.035 seconds
Discrete-time zero/pole/gain model.
Model Properties
```

```
[~, zp, ~] = zpkdata(BoG, 'v');

c_disc_inf = dcgain(BoG); % lim(z->1) BoG
ks_disc = ceil(aproximacion / log(abs(max(zp)))) + (length(zp) - 1); % Teniendo en
cuenta el retardo
ts_disc = Ts * ks_disc;

fprintf("Parámetros del BoG: \n\tc_inf = %.0" + decimales + "f m/s\n\tks = %d\n\tts
= %.0"+decimales+"f s", c_disc_inf, ks_disc, ts_disc)
```

```
Parámetros del BoG:
c_inf = 0.19935 m/s
ks = 23
ts = 0.80500 s
```

Ejercicio 5

```
pert10personas = 0.2;

% Con el número máximo de personas, el escalón unidad no es suficiente para iniciar
el movimiento positivo.
% La perturbación máxima para el valor positivo es de -0.2m/s, valor con módulo
superior a la salida frente al escalón unidad (0.1993 m/s).
c_inf_pert_pos = max([0, Km*1 - pert10personas]);
% Sin embargo, en el movimiento negativo, la perturbación ""ayuda"" a bajar, por lo
que el valor final será de:
c_inf_pert_neg = min([0, Km*(-1) - pert10personas]);

fprintf("Desplazamiento con escalón de 1V y 10 personas: c = %.02f",
c_inf_pert_pos)
```

Desplazamiento con escalón de 1V y 10 personas: $c = 0.00$

```
fprintf("Desplazamiento con escalón de -1V y 10 personas: c = %.02f",  
c_inf_pert_neg)
```

Desplazamiento con escalón de -1V y 10 personas: $c = -0.40$

Fase 2

Ejercicio 6

Calcule la FdeT del regulador con sus correspondientes unidades, mediante la aplicación detallada del método directo, el resultado debe presentarlo de la forma factorizada, polinomial y con potencias negativas en z. Presente la ecuación en diferencias del regulador.

```
% Normalizar a planta de la forma Kg/(s-pg) = (Km/tau)/(s - (-1/tau)) = Km/(tau*s +  
1)
```

```
Kg = Km/tau;
```

```
pg = -1/tau;
```

```
clearvars -except Kg pg Ts aproximacion decimales tiempos_sim salidas_sim  
errores_SA acciones_SA salidas_SA errores_CA acciones_CA salidas_CA  
eje_tiempos_fase3 errores_SA_f5 acciones_SA_f5 salidas_SA_f5 errores_CA_f5  
acciones_CA_f5 salidas_CA_f5 eje_tiempos_fase5;
```

```
% Aproximación del BoG, teniendo en cuenta el retardo adicional
```

```
G = tf(Kg, [1 -pg]);
```

```
retardo = tf(1, [1 0], Ts);
```

```
BoG = series(c2d(G, Ts), retardo)
```

BoG =

```
      0.02596  
-----  
z^2 - 0.8698 z
```

Sample time: 0.035 seconds

Discrete-time transfer function.

Model Properties

```
[NumBoG, DenBoG] = tfdata(BoG, 'v');
```

```
% Especificaciones deseadas
```

```
ts = 0.980;
```

```
erpp = 0.03;
```

```
% Calculo diferencias de grados
```

```
grado_Den_BoG = length(DenBoG) - find(DenBoG ~= 0, 1, 'first');
```

```
grado_Num_BoG = length(NumBoG) - find(NumBoG ~= 0, 1, 'first');
```

```
dif_grados_G = grado_Den_BoG - grado_Num_BoG;
```

```
retardos = dif_grados_G - 1;
```

```
% Calculo de parámetros
ts_prima = ts - retardos*Ts;
pm = exp(Ts * aproximacion / ts_prima); % Ignorando solución negativa por "no ser práctica"
km = (1-pm)*(1-erpp);

fprintf("===== MÉTODO DE TRUXAL =====\n" + ...
    "1. Aplicabilidad del modelo. Aplicable por entrada al escalón y singularidades que la unidad\n" + ...
    "2. Causalidad: Diferencia de grados = %d\n" + ...
    "3. Modelo genérico. Se genera un modelo sobreamortiguado teniendo en cuenta la diferencia de grados del paso anterior\n" + ...
    "\t Parámetros: pm = %.0" + decimales + "f\tkm=%.0" + decimales + "f", ...
    dif_grados_G, pm, km);
```

```
===== MÉTODO DE TRUXAL =====
1. Aplicabilidad del modelo. Aplicable por entrada al escalón y singularidades que la unidad
2. Causalidad: Diferencia de grados = 2
3. Modelo genérico. Se genera un modelo sobreamortiguado teniendo en cuenta la diferencia de grados del paso anterior
   Parámetros: pm = 0.89484    km=0.10201
```

```
M_obj = zpk(tf(km, [1, -pm], Ts)*(tf(1, [1 0], Ts)^retardos))
```

```
M_obj =
```

```
    0.10201
-----
z (z-0.8948)
```

```
Sample time: 0.035 seconds
Discrete-time zero/pole/gain model.
Model Properties
```

```
fprintf("4. Cálculo del controlador")
```

```
4. Cálculo del controlador
```

```
F = zpk(minreal(M_obj/(BoG * (1 - M_obj))))
```

```
F =
```

```
    3.9289 z (z-0.8698)
-----
(z+0.1023) (z-0.9971)
```

```
Sample time: 0.035 seconds
Discrete-time zero/pole/gain model.
Model Properties
```

```
F = tf(F)
```

```
F =
```

```
    3.929 z^2 - 3.417 z
-----
```

$$z^2 - 0.8948z - 0.102$$

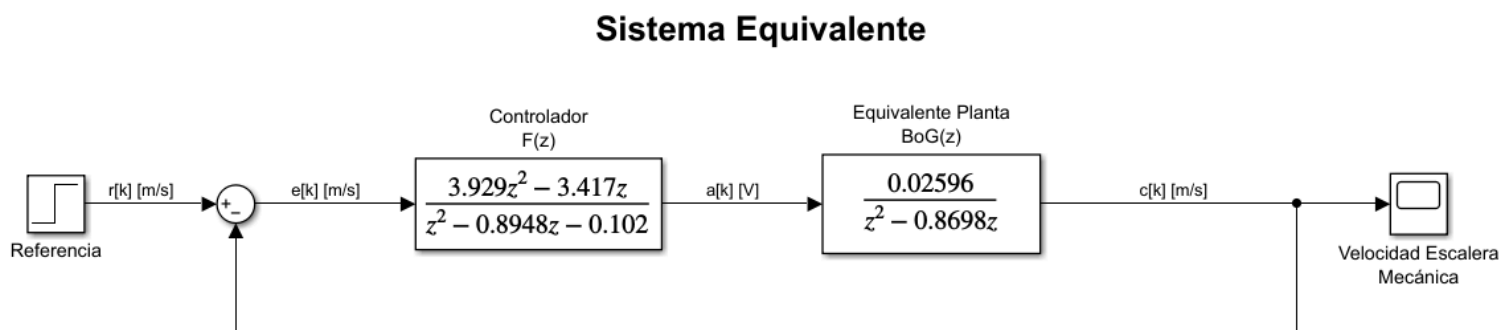
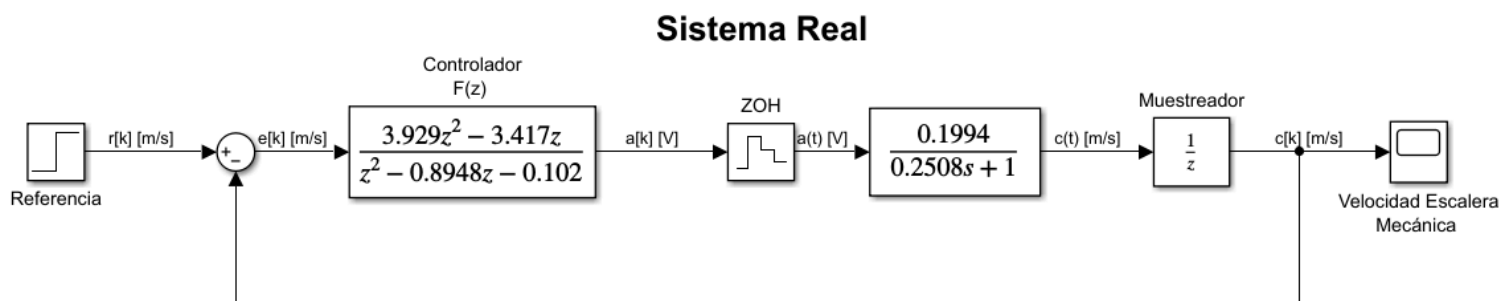
Sample time: 0.035 seconds
Discrete-time transfer function.
Model Properties

```
get_eq_diff(F, decimales, 'e', 'a')
```

Eq en diferencias: $c[k] = 3.92892 \cdot e[k] - 3.41723 \cdot e[k-1] + 0.89484 \cdot a[k-1] + 0.10201 \cdot a[k-2]$
Eq para GUI: $y0 = 3.92892 \cdot x0 - 3.41723 \cdot x1 + 0.89484 \cdot y1 + 0.10201 \cdot y2;$

Ejercicio 7

Dibuje una topología (diagrama de bloques) del sistema de acuerdo a las unidades elegidas, que deberán aparecer indicadas en cada señal o secuencia del dibujo.



Ejercicio 8

Calcule la FdeT del sistema y caracterice teóricamente su salida ante entradas escalón de ± 1 m/s.

```
fprintf("Valor de FdeT del sistema controlado")
```

Valor de FdeT del sistema controlado

```
M_final = zpk(minreal(feedback(series(F, BoG), 1)))
```

```
M_final =
```

```
    0.10201  
-----  
z (z-0.8948)
```

```
Sample time: 0.035 seconds  
Discrete-time zero/pole/gain model.  
Model Properties
```

```
p_m_final = pole(M_final);  
  
ts_M = Ts * (ceil(round(aproximacion / log(max(p_m_final)), 10)) +  
(length(p_m_final) - 1)); % ceil(round(...)) para evitar que x + 1e-12 suba a x  
+ 1  
Km_final = dcgain(M_final); % lim_(z->1) M_final  
c_inf_mas_1V = Km_final * 1;  
c_inf_menos_1V = Km_final * (-1);  
  
fprintf("Caracterización de salida frente a escalón")
```

Caracterización de salida frente a escalón

```
fprintf("Escalón de 1 m/s:\tc(inf)=%.0"+decimales+"f m/s, ts=%.0"+decimales+"f s",  
c_inf_mas_1V, ts_M);
```

Escalón de 1 m/s: c(inf)=0.97000 m/s, ts=0.98000 s

```
fprintf("Escalón de -1 m/s:\tc(inf)=%.0"+decimales+"f m/s, ts=%.0"+decimales+"f s",  
c_inf_menos_1V, ts_M);
```

Escalón de -1 m/s: c(inf)=-0.97000 m/s, ts=0.98000 s

Ejercicio 9

Calcule en régimen permanente tanto la acción de control como la velocidad de la escalera subiendo y bajando en diferentes condiciones de carga de la escalera, para consignas de ± 1 m/s. Relacione los incrementos o decrementos de acción de control con el peso de las personas presentes en la escalera.

```
pert_5personas = -0.1;  
pert_10personas = -0.2;
```

Aplicando teorema de superposición

```
M_r = M_final;  
M_p = zpk(minreal(feedback(1, series(F, BoG))));
```

La perturbación no afecta al tiempo de establecimiento, solo al valor final (y por tanto al error)

```
K0_ref = dcgain(M_r);  
K0_pert= dcgain(M_p);  
K0_F = dcgain(F);
```

```

entradas          = [1 -1];
perturbaciones    = [0 -0.1 -0.2];
perturbaciones_names = [" 0 personas", " 5 personas", "10 personas"];

fprintf("Cálculo de error y acción en régimen permanente para distintas
perturbaciones\n")

```

Cálculo de error y acción en régimen permanente para distintas perturbaciones

```

for i = 1:length(entradas)
    fprintf("\nPara escalón de %.2fV:\n", entradas(i))
    for j = 1:length(perturbaciones)
        r = entradas(i);
        p = perturbaciones(j);

        c = K0_ref * r + K0_pert * p;
        e = r - c;
        a = K0_F * e;

        fprintf("\tCon %s:\tc_inf=%.0" + decimales + "f\terpp=%.0" + decimales +
"f\ta_inf=%.0" + decimales + "f\n", perturbaciones_names(j), c, e, a);
    end
end

```

Para escalón de 1.00V:

| | | | |
|------------------|---------------|--------------|---------------|
| Con 0 personas: | c_inf=0.97000 | erpp=0.03000 | a_inf=4.86581 |
| Con 5 personas: | c_inf=0.96700 | erpp=0.03300 | a_inf=5.35240 |
| Con 10 personas: | c_inf=0.96400 | erpp=0.03600 | a_inf=5.83898 |

Para escalón de -1.00V:

| | | | |
|------------------|----------------|---------------|----------------|
| Con 0 personas: | c_inf=-0.97000 | erpp=-0.03000 | a_inf=-4.86581 |
| Con 5 personas: | c_inf=-0.97300 | erpp=-0.02700 | a_inf=-4.37923 |
| Con 10 personas: | c_inf=-0.97600 | erpp=-0.02400 | a_inf=-3.89265 |

Fase 3

Ejercicio 10

Simule en Simulink el sistema de control con y sin alinealidades. Mida las características de las respuestas (tanto el régimen permanente como el transitorio) ante entradas escalón de ± 1 m/s. Analice las diferencias de las medidas respecto de los cálculos teóricos.

```

if ~exist("acciones_SA", 'var') % evita simular todo el rato
    open_system("Fase3/modeloSimulink");

    % Controlador
    % SA ≡ Sin Alinealidades, CA ≡ Con Alinealidades
    [N,D] = tfdata(F, 'v');
    set_param("modeloSimulink/Controlador - SA", 'Numerator', '[' +
strjoin(string(N)) + ']', 'Denominator', '[' + strjoin(string(D)) + ']')
    set_param("modeloSimulink/Controlador - CA", 'Numerator', '[' +
strjoin(string(N)) + ']', 'Denominator', '[' + strjoin(string(D)) + ']')

```

```

% Planta
[N,D] = tfdata(G, 'v');
set_param("modeloSimulink/Planta - CA", 'Numerator', '[' + strjoin(string(N)) +
'],'', 'Denominator', '[' + strjoin(string(D)) + ']')
set_param("modeloSimulink/Planta - CA", 'Numerator', '[' + strjoin(string(N)) +
'],'', 'Denominator', '[' + strjoin(string(D)) + ']')

sim("modeloSimulink")
errores_SA = zeros(length(SA.signals(1).values),1);
acciones_SA = zeros(length(SA.signals(1).values),1);
salidas_SA = zeros(length(SA.signals(1).values),1);
errores_CA = zeros(length(CA.signals(1).values),1);
acciones_CA = zeros(length(CA.signals(1).values),1);
salidas_CA = zeros(length(CA.signals(1).values),1);
eje_tiempos_fase3 = CA.time; % Igual para Con/Sin Alinealidades y las distintas
señales

% Simular
for i = 1:length(entradas) %  $\pm 1\text{m/s}$ 
    set_param("modeloSimulink/Ref - SA", 'After', string(entradas(i)))
    set_param("modeloSimulink/Ref - CA", 'After', string(entradas(i)))

    for j = 1:length(perturbaciones)
        perturbacion = perturbaciones(j);
        set_param("modeloSimulink/Personas - SA", 'After', string(perturbacion))
        set_param("modeloSimulink/Personas - CA", 'After', string(perturbacion))
        sim("modeloSimulink")

        errores_SA(:, 3*(i - 1) + j) = SA.signals(1).values;
        acciones_SA(:, 3*(i - 1) + j) = SA.signals(2).values;
        salidas_SA(:, 3*(i - 1) + j) = SA.signals(3).values;

        errores_CA(:, 3*(i - 1) + j) = CA.signals(1).values;
        acciones_CA(:, 3*(i - 1) + j) = CA.signals(2).values;
        salidas_CA(:, 3*(i - 1) + j) = CA.signals(3).values;
    end
end
end

% Representar
for i = 1:length(entradas)
    for j = 1:length(perturbaciones)
        % Sin alinealidades
        figure
        sgtitle("Consigna " + entradas(i) + " m/s, carga de " +
perturbaciones_names(j) + " - Sin alinealidades")
        subplot(1, 2, 1)
        hold on
        stairs(eje_tiempos_fase3, errores_SA(:, 3*(i - 1) + j))
    end
end

```



```

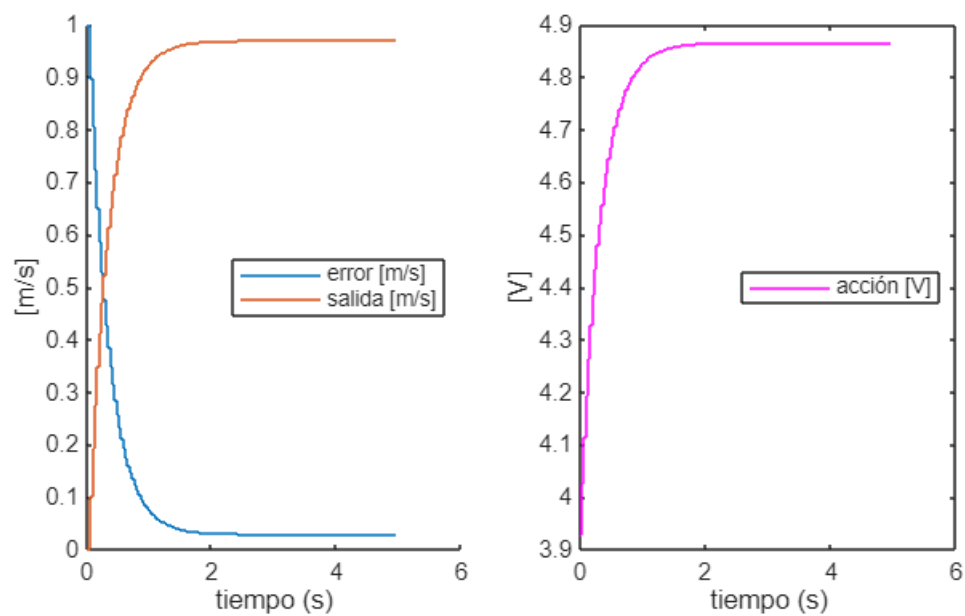
stairs(eje_tiempos_fase3, salidas_SA(:, 3*(i - 1) + j))
legend("error [m/s]", "salida [m/s]", Location="east")
xlabel("tiempo (s)")
ylabel("[m/s]")
subplot(1, 2, 2)
stairs(eje_tiempos_fase3, acciones_SA(:, 3*(i - 1) + j), Color="m")
legend("acción [V]", Location="east")
xlabel("tiempo (s)")
ylabel("[V]")

% Con alinealidades
figure
sgtitle("Consigna " + entradas(i) + " m/s, carga de " +
perturbaciones_names(j) + " - Con alinealidades")
subplot(1, 2, 1)
hold on
stairs(eje_tiempos_fase3, errores_CA(:, 3*(i - 1) + j))
stairs(eje_tiempos_fase3, salidas_CA(:, 3*(i - 1) + j))
legend("error [m/s]", "salida [m/s]", Location="east")
xlabel("tiempo (s)")
ylabel("[m/s]")
subplot(1, 2, 2)
stairs(eje_tiempos_fase3, acciones_CA(:, 3*(i - 1) + j), Color="m")
legend("acción [V]", Location="east")
xlabel("tiempo (s)")
ylabel("[V]")

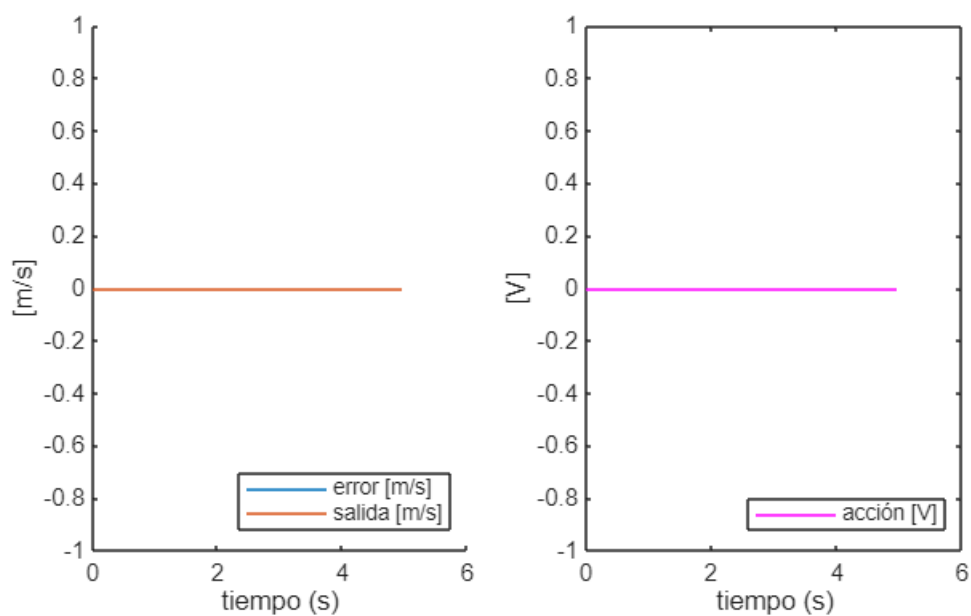
% Diferencia entre Sin y Con alinealidades
figure
sgtitle("Consigna " + entradas(i) + " m/s, carga de " +
perturbaciones_names(j) + " - Diferencia")
subplot(1, 2, 1)
hold on
stairs(eje_tiempos_fase3, errores_CA(:, 3*(i - 1) + j) - errores_SA(:,
3*(i - 1) + j))
stairs(eje_tiempos_fase3, salidas_CA(:, 3*(i - 1) + j) - salidas_SA(:,
3*(i - 1) + j))
legend("error [m/s]", "salida [m/s]", Location="southeast")
xlabel("tiempo (s)")
ylabel("[m/s]")
subplot(1, 2, 2)
stairs(eje_tiempos_fase3, acciones_CA(:, 3*(i - 1) + j) - acciones_SA(:,
3*(i - 1) + j), Color="m")
legend("acción [V]", Location="southeast")
xlabel("tiempo (s)")
ylabel("[V]")
end
end

```

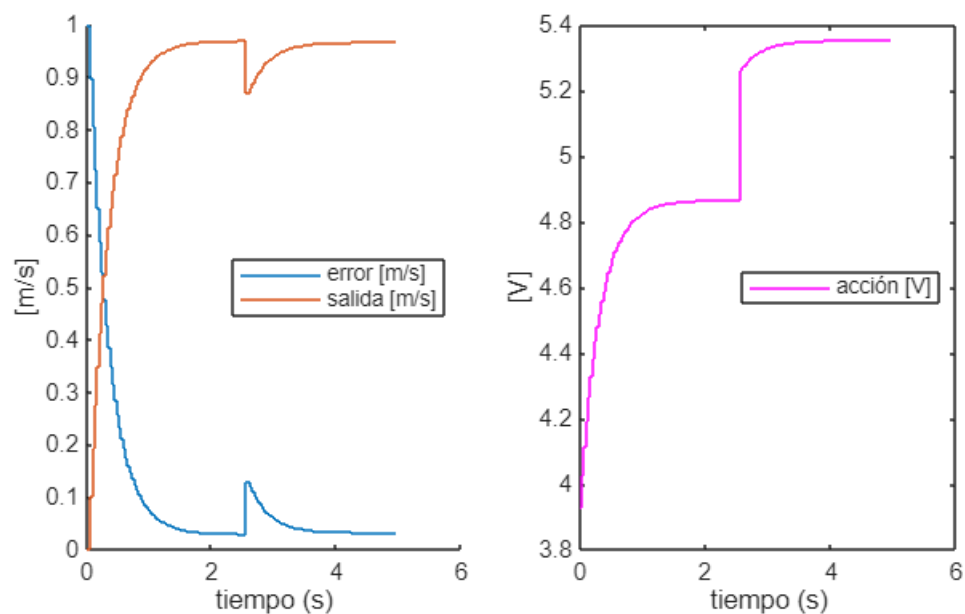
Consigna 1 m/s, carga de 0 personas - Con alinealidades



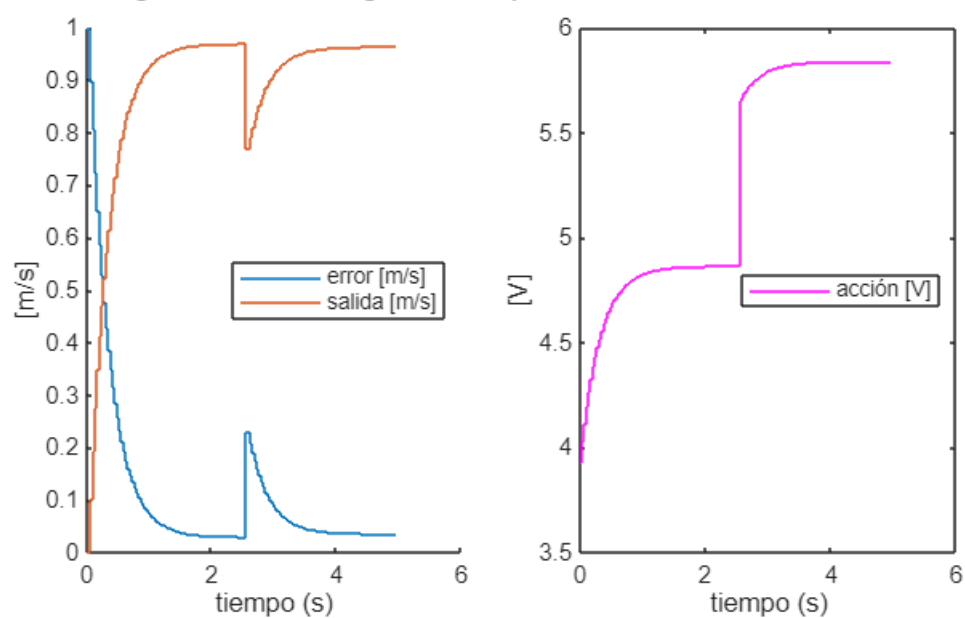
Consigna 1 m/s, carga de 0 personas - Diferencia



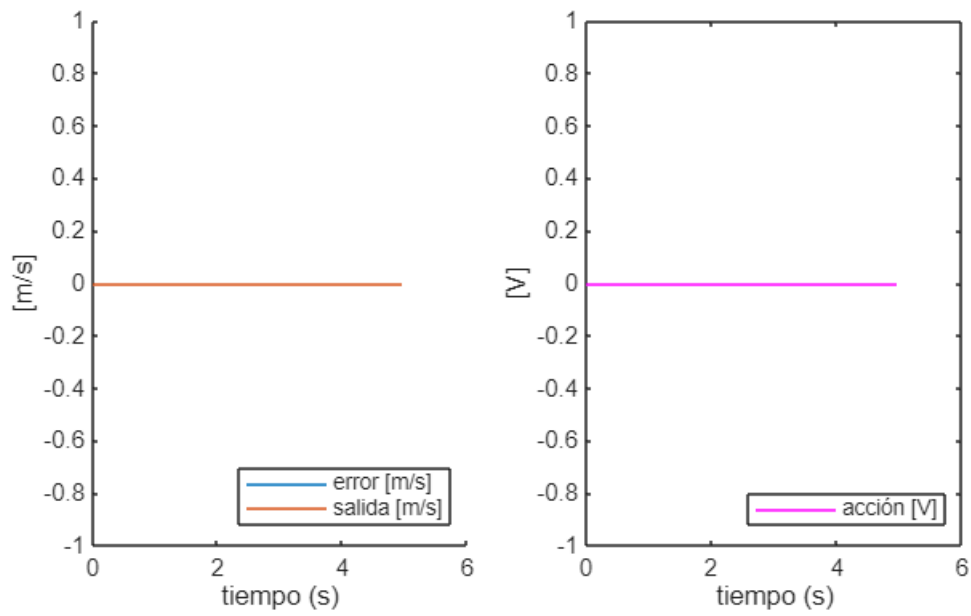
Consigna 1 m/s, carga de 5 personas - Sin alinealidades



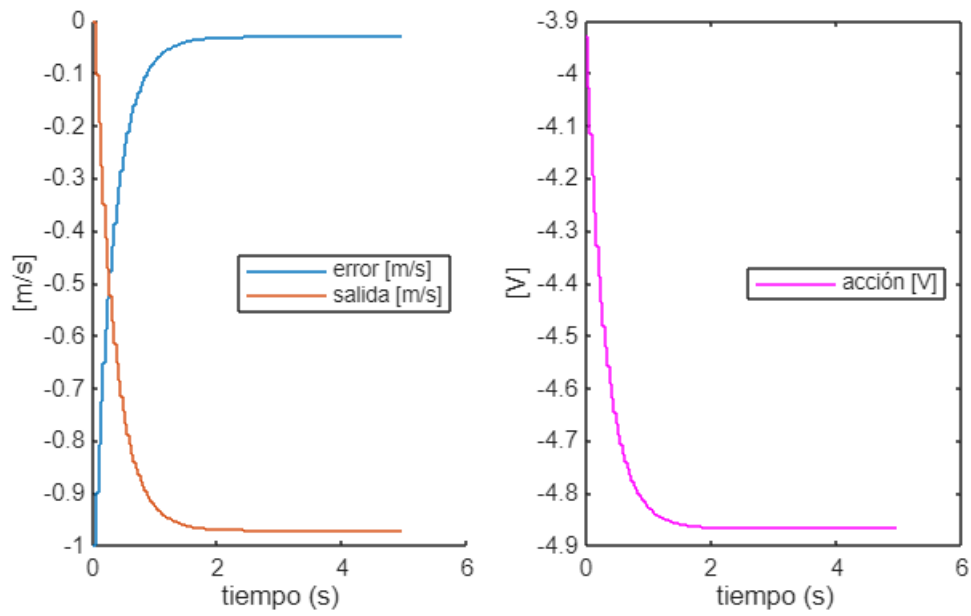
Consigna 1 m/s, carga de 10 personas - Sin alinealidades



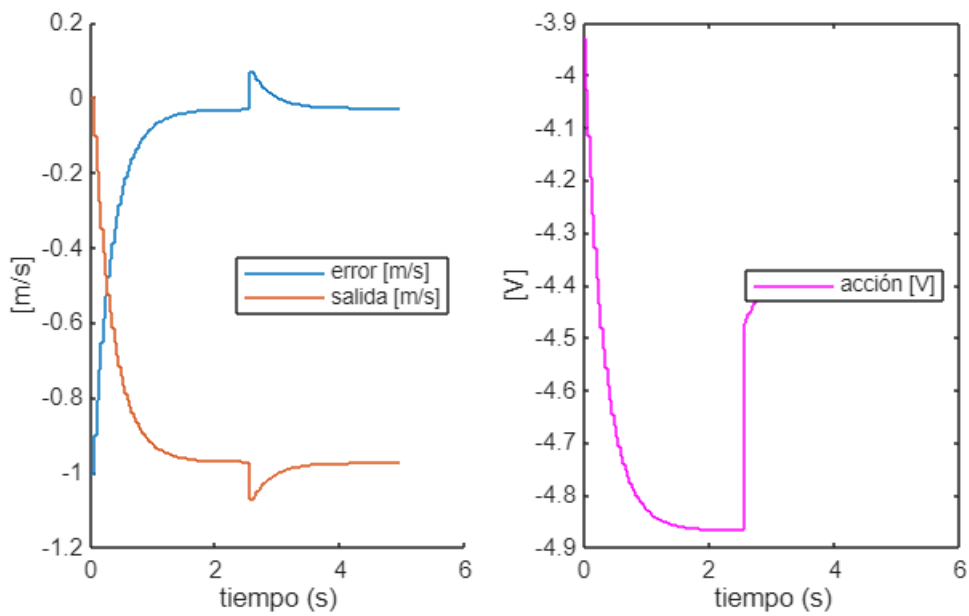
Consigna 1 m/s, carga de 5 personas - Diferencia



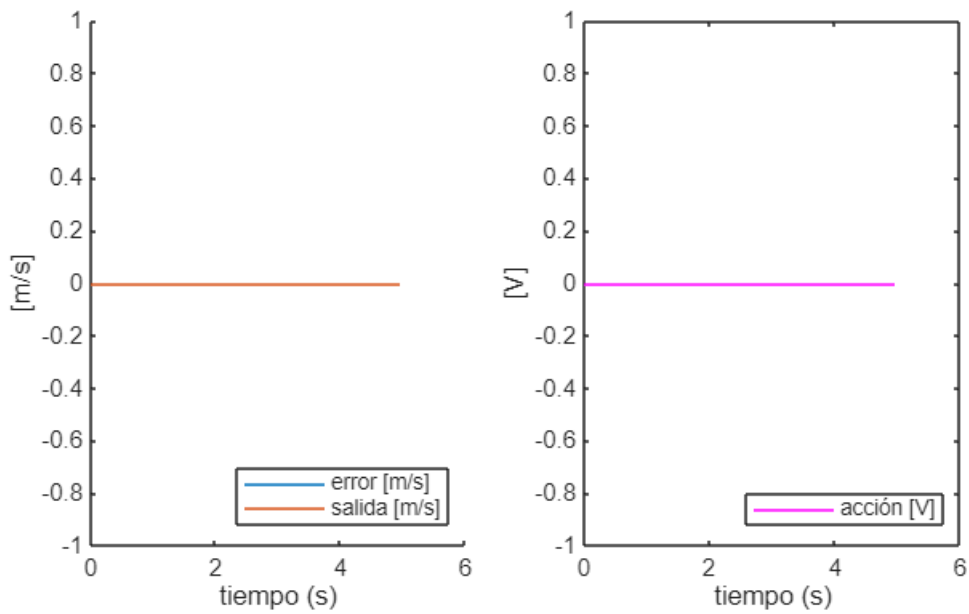
Consigna -1 m/s, carga de 0 personas - Sin alinealidades



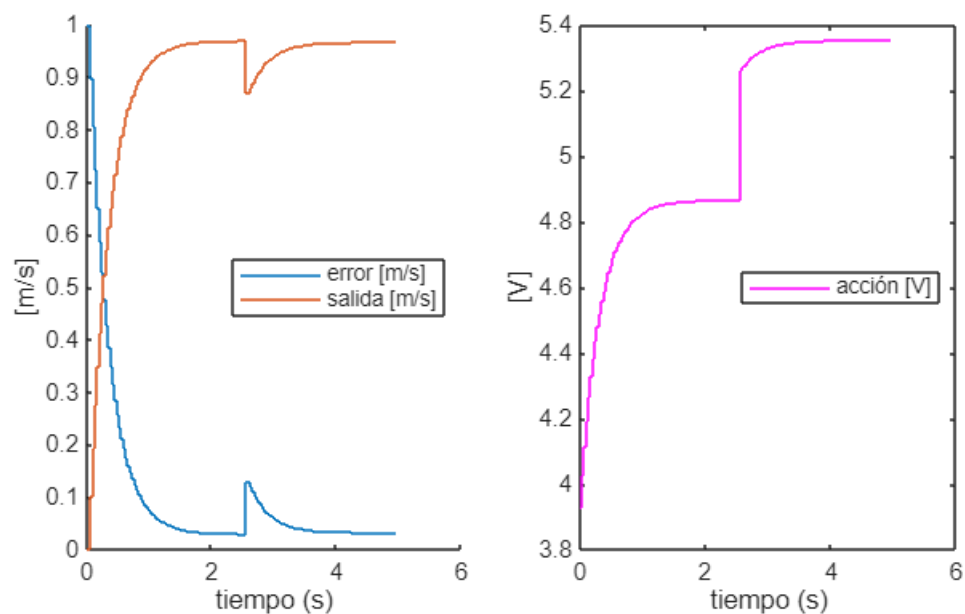
Consigna -1 m/s, carga de 5 personas - Sin alinealidades



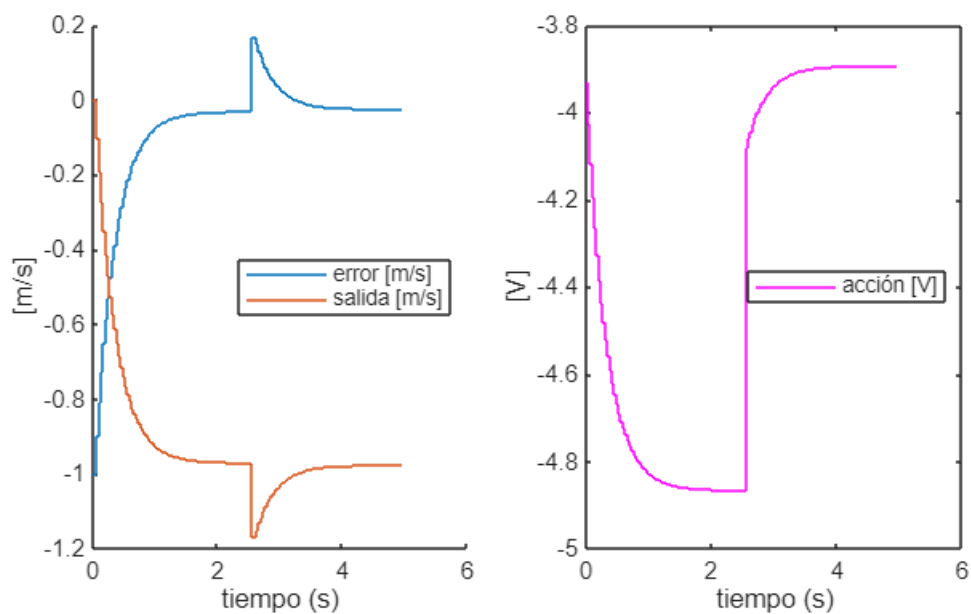
Consigna 1 m/s, carga de 10 personas - Diferencia



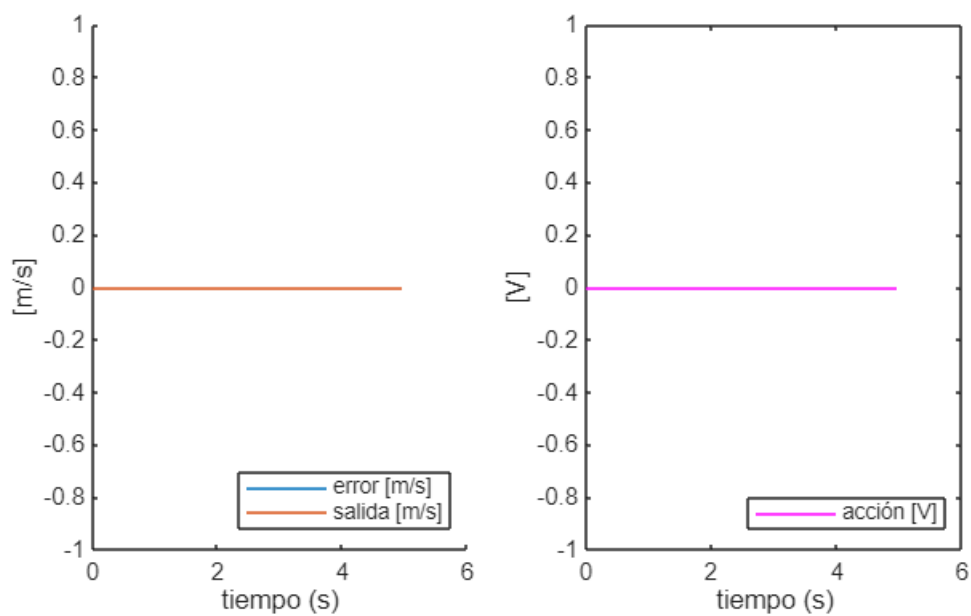
Consigna 1 m/s, carga de 5 personas - Con alinealidades



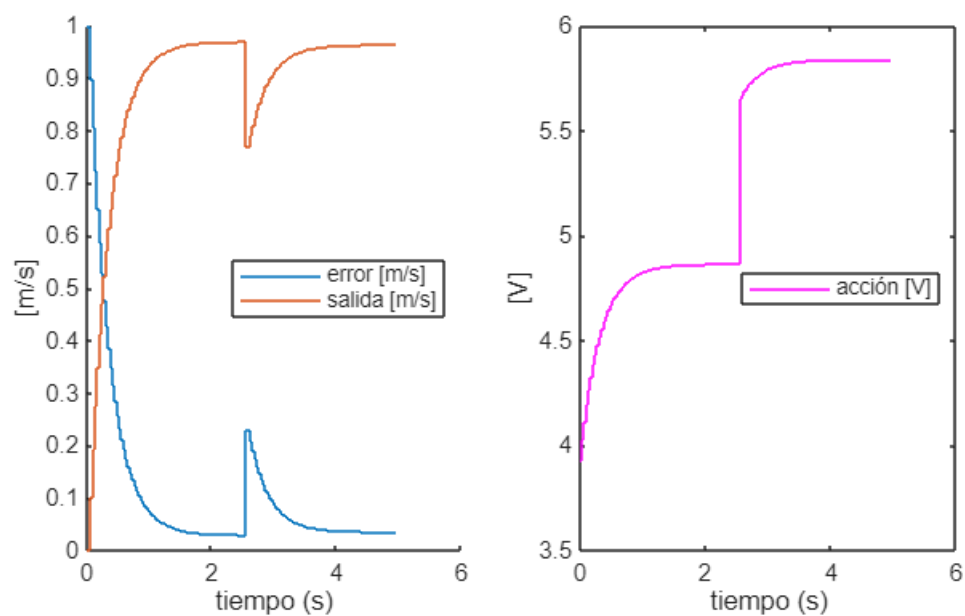
Consigna -1 m/s, carga de 10 personas - Sin alinealidades



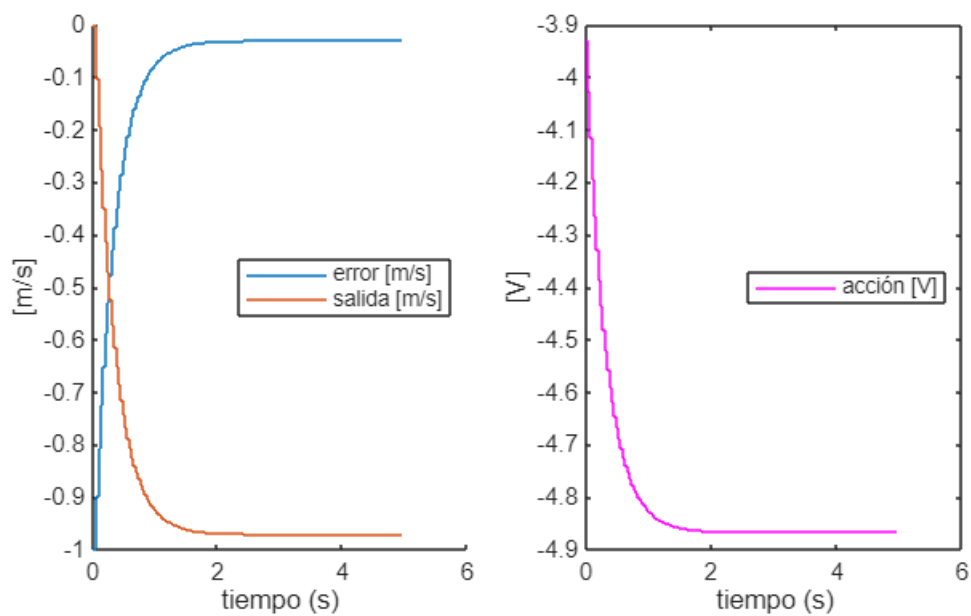
Consigna -1 m/s, carga de 0 personas - Diferencia



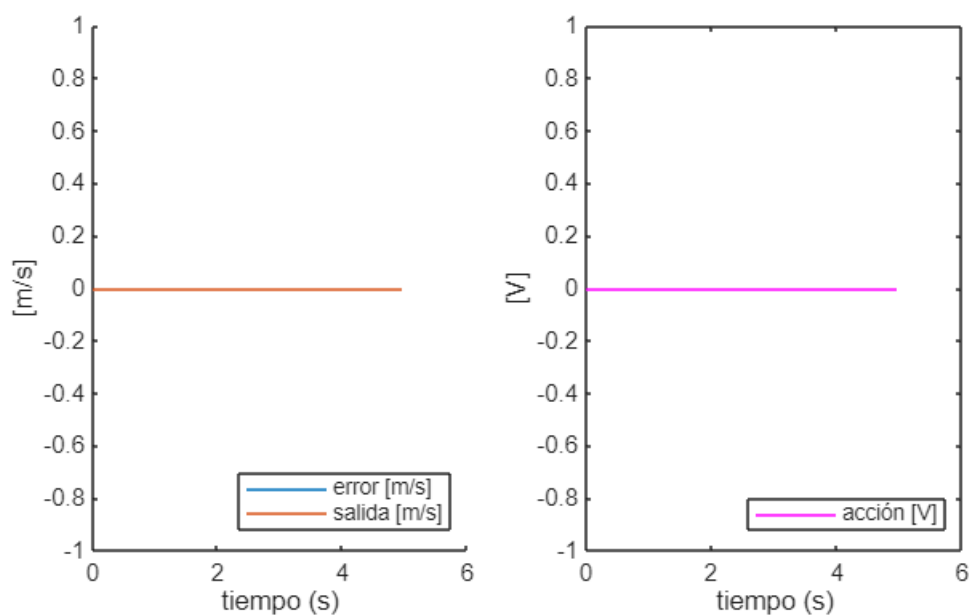
Consigna 1 m/s, carga de 10 personas - Con alinealidades



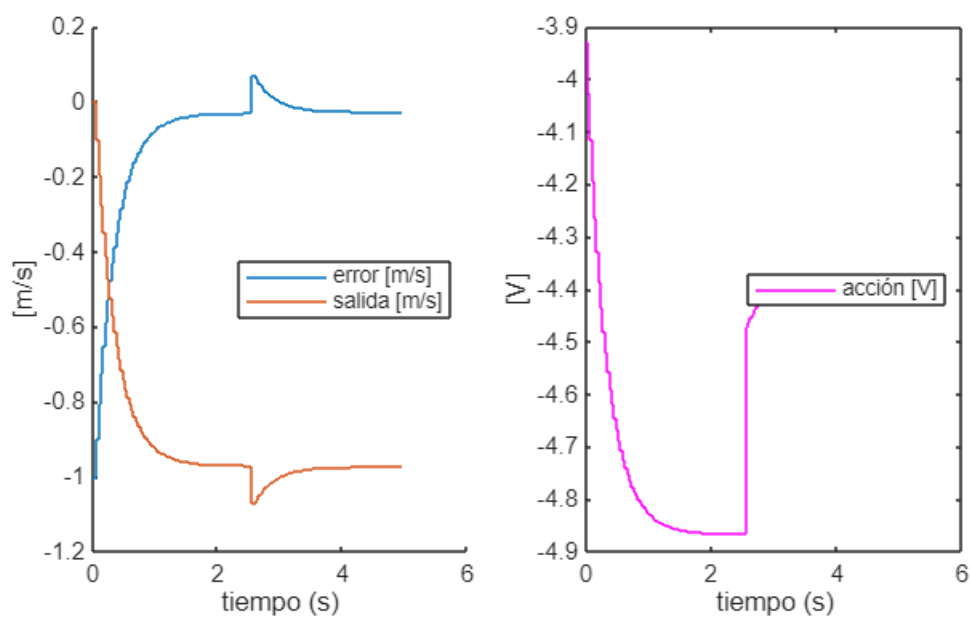
Consigna -1 m/s, carga de 0 personas - Con alinealidades



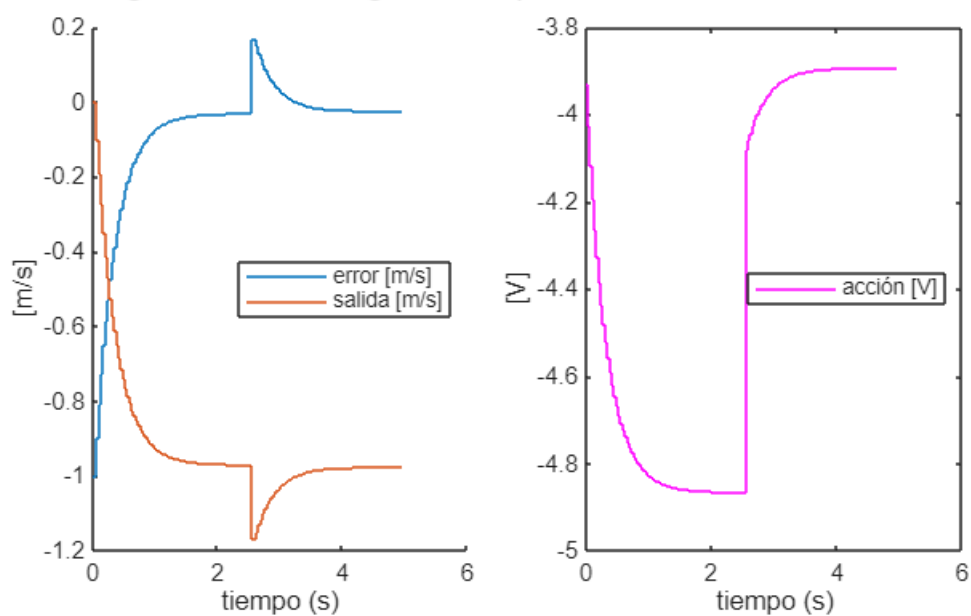
Consigna -1 m/s, carga de 5 personas - Diferencia



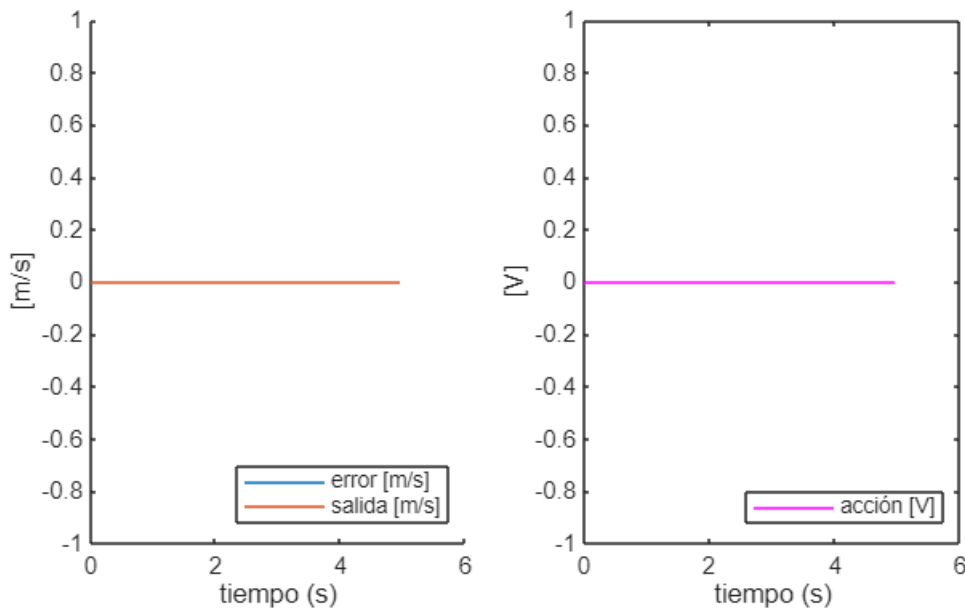
Consigna -1 m/s, carga de 5 personas - Con alinealidades



Consigna -1 m/s, carga de 10 personas - Con alinealidades



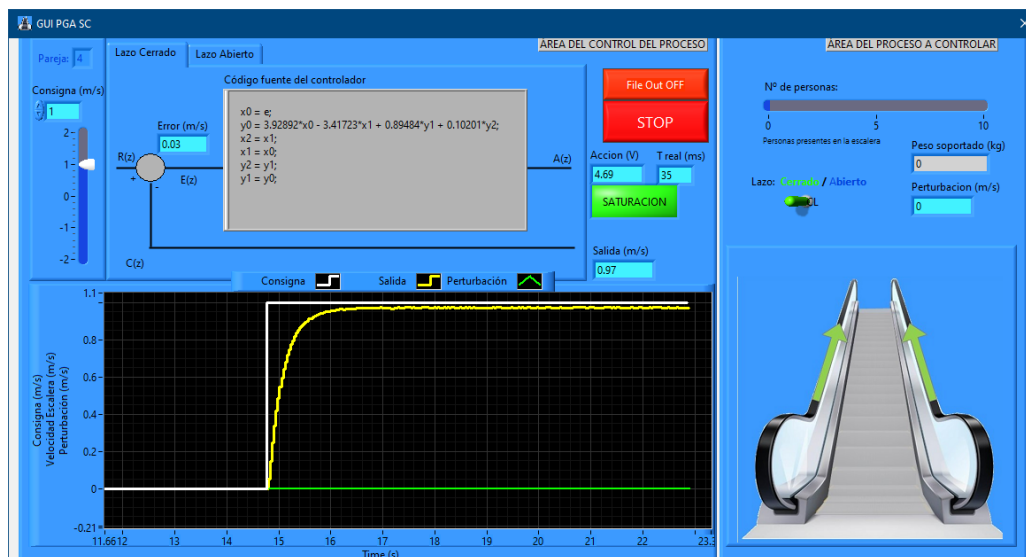
Consigna -1 m/s, carga de 10 personas - Diferencia



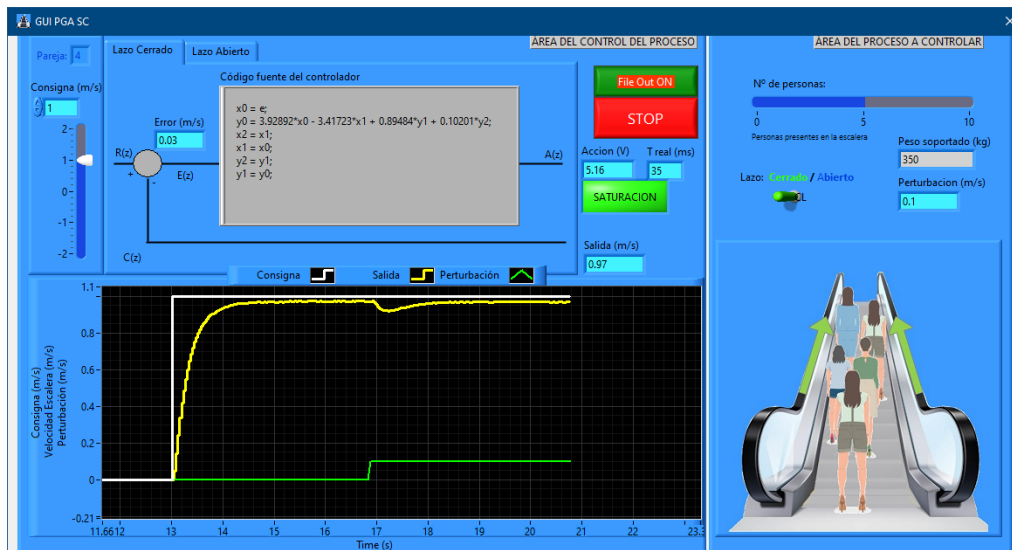
Ejercicio 11

Mediante el GUI, obtenga la captura de la respuesta del sistema para una consigna de 1 m/s (subiendo) y sin personas transportadas (sin carga). Adjunte una captura de pantalla de la ventana de gestión del sistema de control en la que se aprecie con la suficiente nitidez el código fuente del regulador (Figura 2.b). Mida las características de la respuesta y coteje los resultados con los de la simulación del ejercicio anterior, así como con los cálculos teóricos realizados en la fase de diseño del regulador.

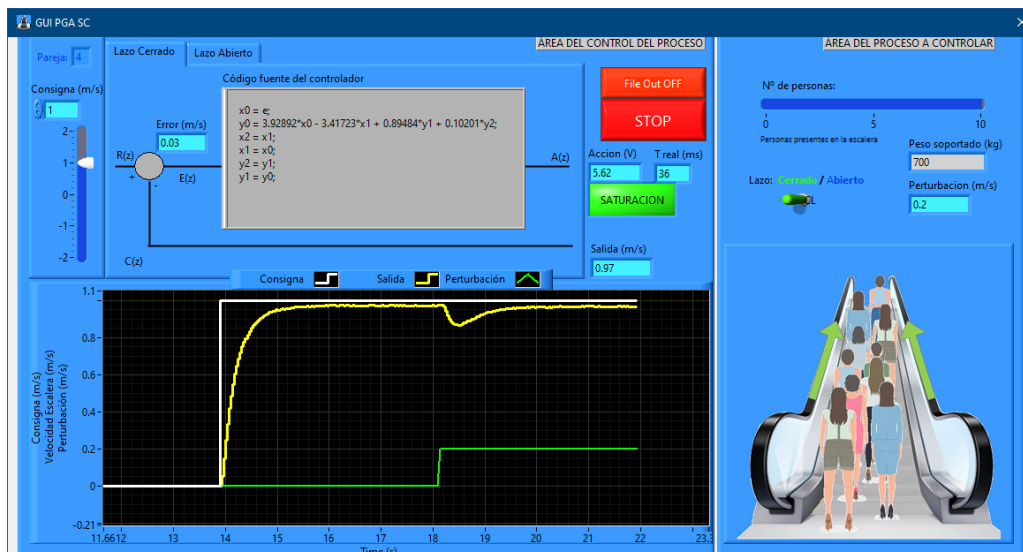
1 m/s sin perturbación



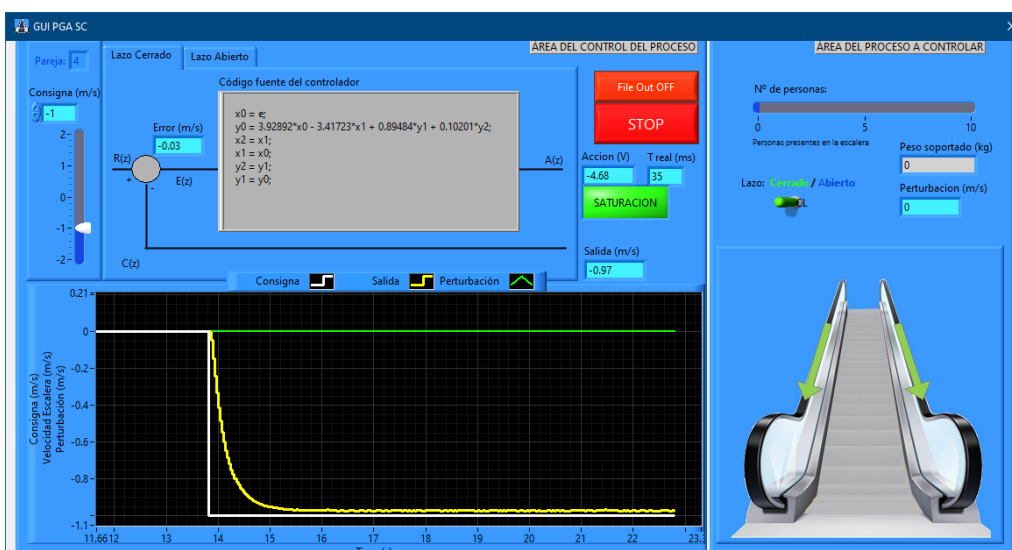
1 m/s 5 personas



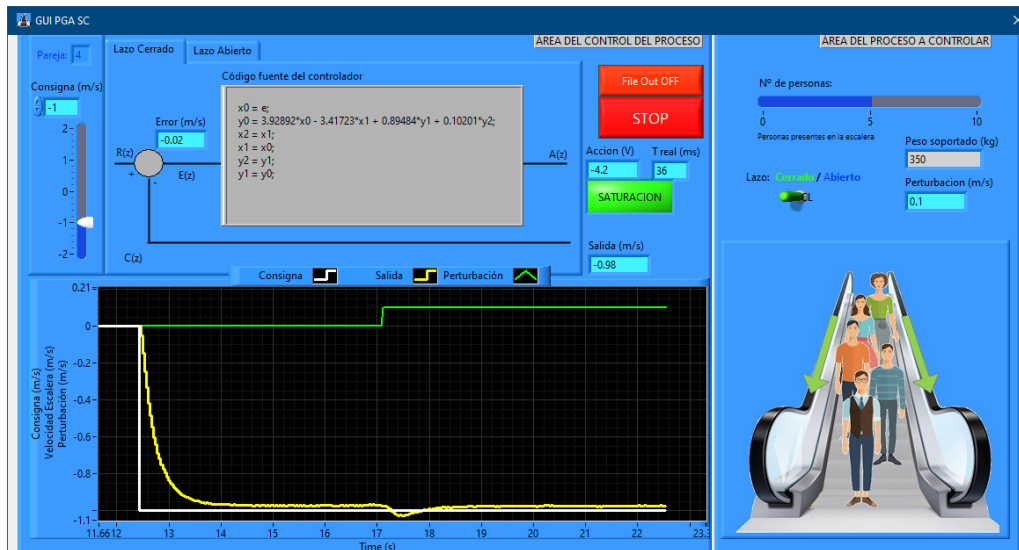
1 m/s 10 personas



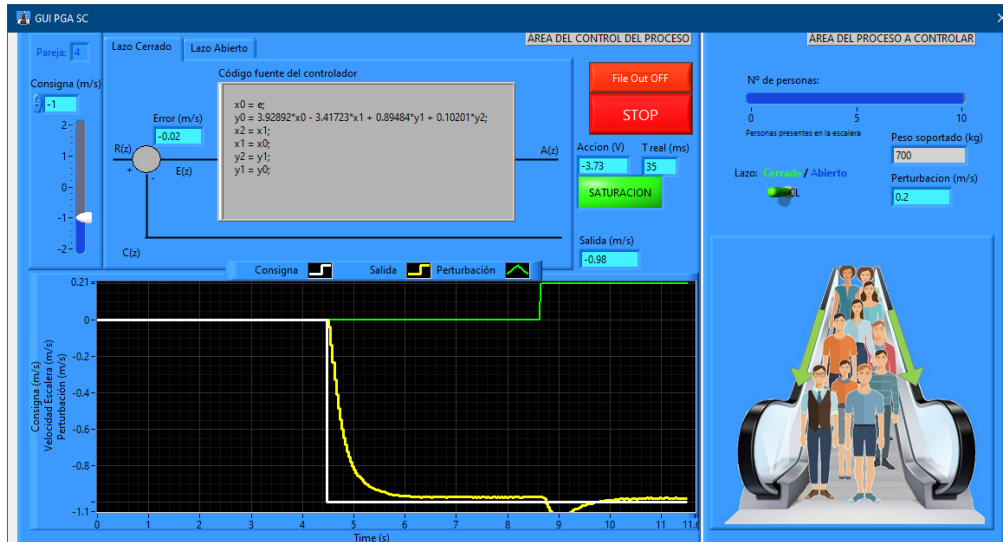
-1 m/s sin perturbación



-1 m/s 5 personas



-1 m/s 10 personas

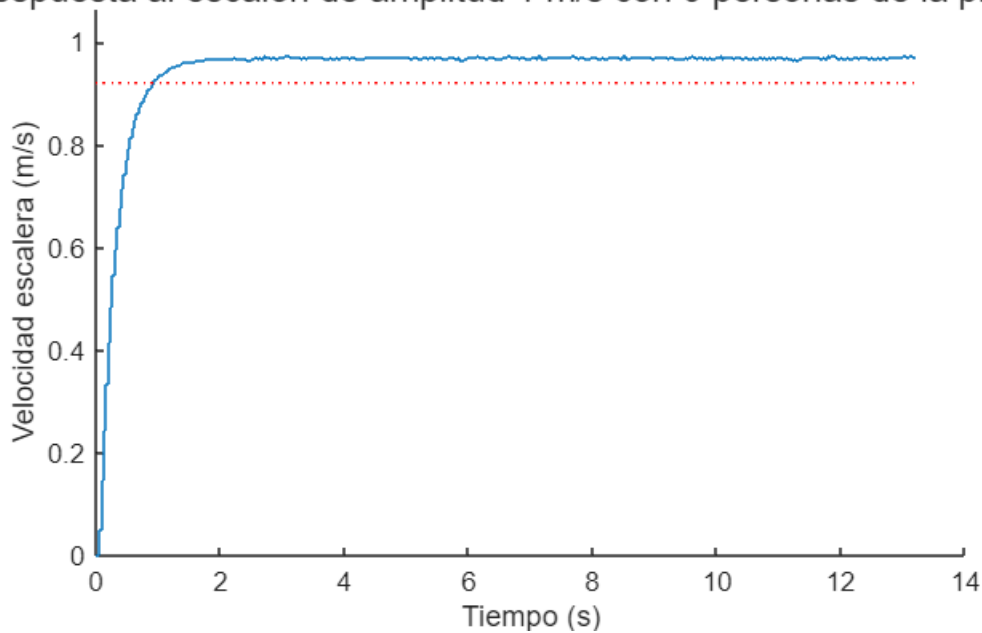


```
ficheros = dir("Fase3/ficherosGUI/positivo");

c_inf = zeros(1,length(ficheros) - 2);

[~, ~, c_inf(1)] = procesar_resultados("Fase3/ficherosGUI/positivo/" +
ficheros(3).name, 0.035, 20, 3); % Sin añadir la simulacion
```

Respuesta al escalón de amplitud 1 m/s con 0 personas de la planta



Escalón de 1 m/s, con 0 personas.

Media de las últimas 20 muestras = 0.970 m/s

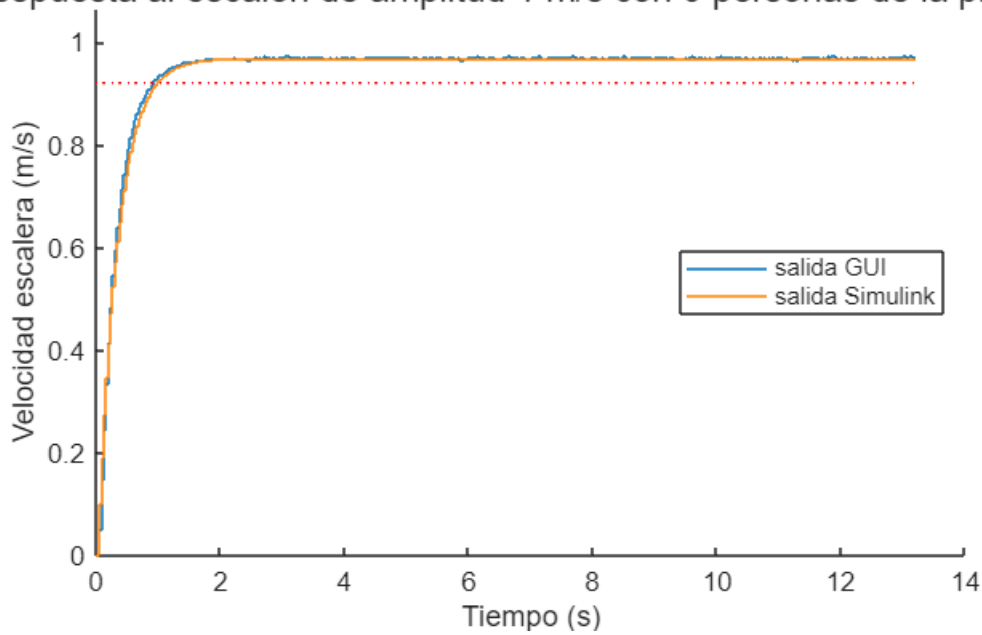
ts = 0.945 s

```
for i = 1:length(ficheros)
    fichero = ficheros(i).name;

    % Saltar '.' y '..'
    if strcmp(fichero, '.') || strcmp(fichero, '..')
        continue;
    end

    [~, erpp, c_inf(i-2)] = procesar_resultados("Fase3/ficherosGUI/positivo/" +
fichero, 0.035, 20, 3, salidas_CA(:, i - 2));
    ts_sim = (find(salidas_SA(:, i-2) >= 0.95*salidas_CA(end, i-2), 1, "first") -
1)* 0.035;
    erpp_sim = entradas(1) - salidas_CA(end, i-2);
    fprintf("\tc_inf_sim: " + salidas_SA(end, i - 2) + ...
        "\n\terpp: " + erpp + ...
        "\n\tts_sim: " + ts_sim + ...
        "\n\terpp_sim: " + erpp_sim)
end
```

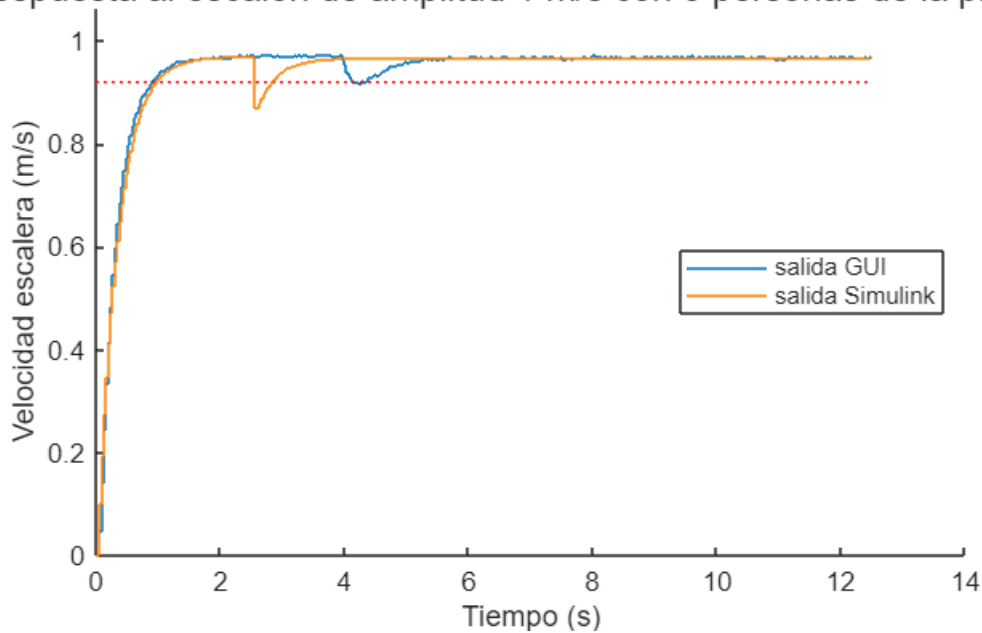
Respuesta al escalón de amplitud 1 m/s con 0 personas de la planta



Escalón de 1 m/s, con 0 personas.

Media de las últimas 20 muestras = 0.970 m/s
 ts = 0.945 s
 c_inf_sim: 0.97004
 erpp: 0.03
 ts_sim: 0.98
 erpp_sim: 0.029955

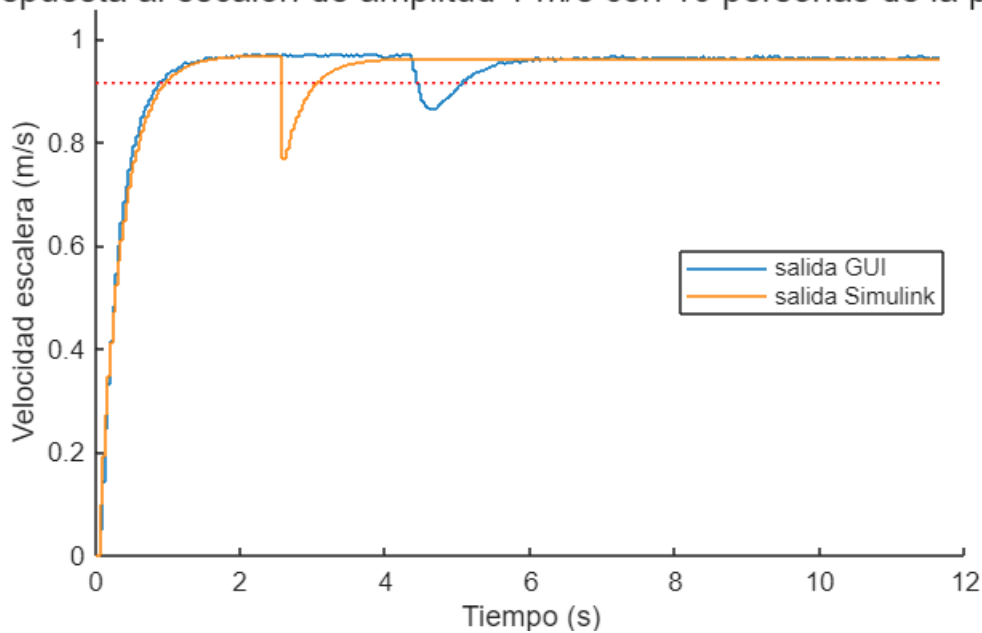
Respuesta al escalón de amplitud 1 m/s con 5 personas de la planta



Escalón de 1 m/s, con 5 personas.

Media de las últimas 20 muestras = 0.968 m/s
 ts = 0.910 s
 c_inf_sim: 0.967
 erpp: 0.032
 ts_sim: 0.98
 erpp_sim: 0.033001

Respuesta al escalón de amplitud 1 m/s con 10 personas de la planta



Escalón de 1 m/s, con 10 personas.

Media de las últimas 20 muestras = 0.964 m/s

ts = 0.875 s

c_inf_sim: 0.96395

erpp: 0.036

ts_sim: 0.945

erpp_sim: 0.036048

Ejercicio 12

Igual que el ejercicio 11 pero con consigna de -1 m/s (bajando).

```
ficheros = dir("Fase3/ficherosGUI/negativo");

idx = length(c_inf);
c_inf = [c_inf zeros(1, length(ficheros) - 2)];

for i = 1:length(ficheros)
    fichero = ficheros(i).name;

    % Saltar '.' y '..'
    if strcmp(fichero, '.') || strcmp(fichero, '..')
        continue;
    end

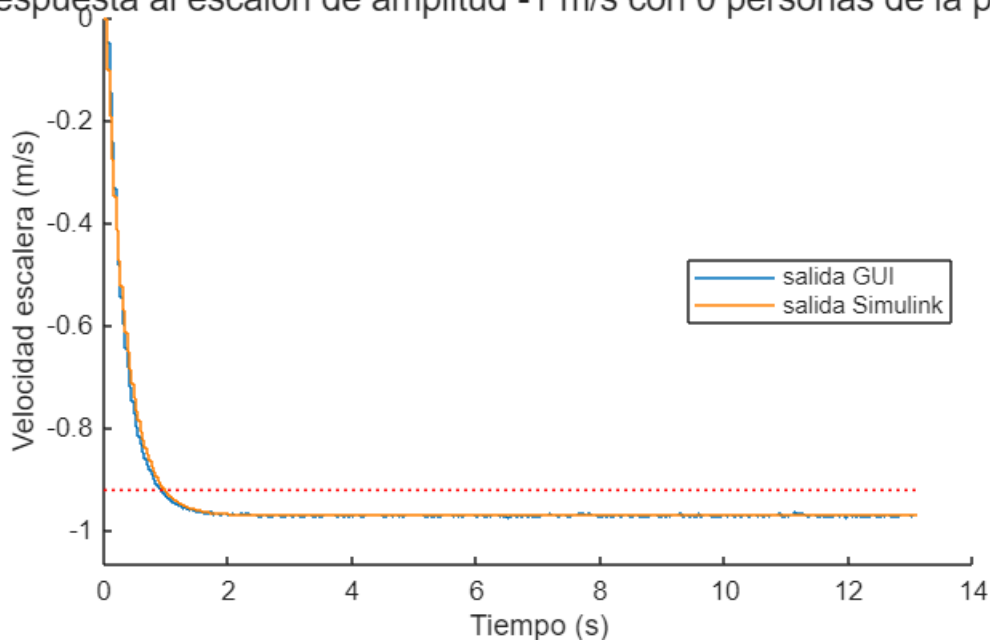
    [~, erpp, c_inf(idx + i - 2)] = procesar_resultados("Fase3/ficherosGUI/
negativo/" + fichero, 0.035, 20, 3, salidas_CA(:, idx + i - 2));
    ts_sim = (find(salidas_SA(:, idx + i - 2) <= 0.95*salidas_CA(end, idx + i - 2),
1, "first") - 1) * 0.035;
    erpp_sim = entradas(2) - salidas_CA(end, idx + i - 2);
    fprintf("\tc_inf_sim: " + salidas_SA(end, idx + i - 2) + ...
        "\n\terpp: " + erpp + ...
        "\n\tts_sim: " + ts_sim + ...
```

```

        "\n\terpp_sim: " + erpp_sim)
end

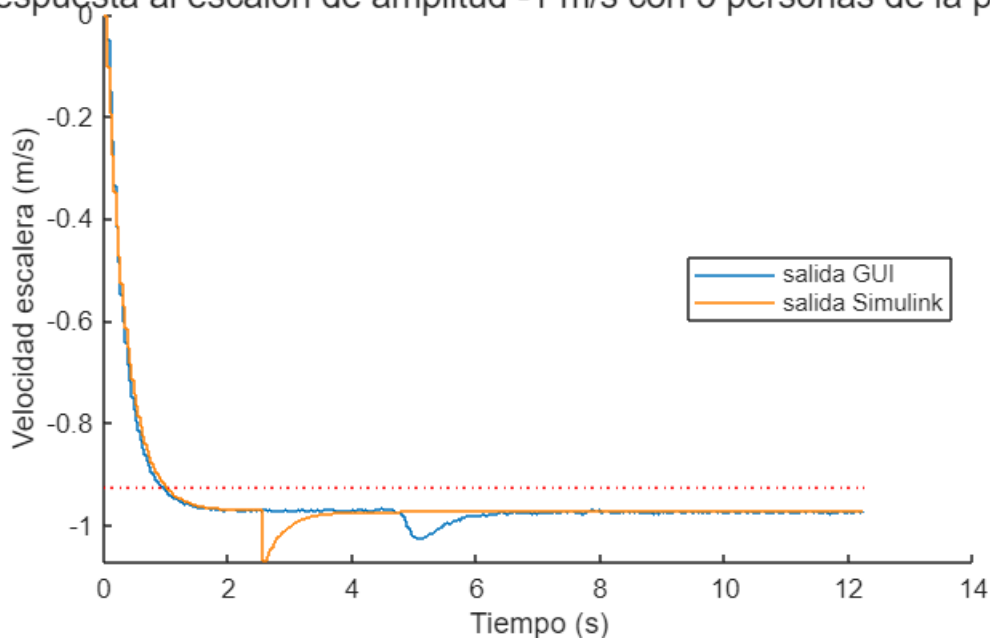
```

Respuesta al escalón de amplitud -1 m/s con 0 personas de la planta



Escalón de -1 m/s, con 0 personas.
 Media de las últimas 20 muestras = -0.971 m/s
 ts = 0.945 s
 c_inf_sim: -0.97004
 erpp: -0.029
 ts_sim: 0.98
 erpp_sim: -0.029955

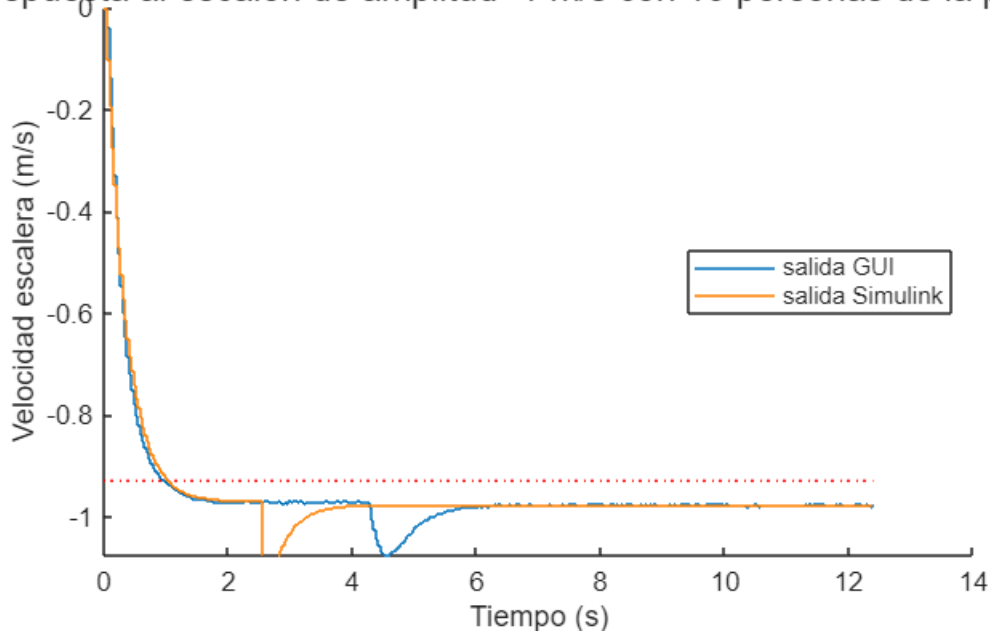
Respuesta al escalón de amplitud -1 m/s con 5 personas de la planta



Escalón de -1 m/s, con 5 personas.
 Media de las últimas 20 muestras = -0.976 m/s
 ts = 0.980 s
 c_inf_sim: -0.97309
 erpp: -0.024


```
ts_sim: 1.015
erpp_sim: -0.026909
```

Respuesta al escalón de amplitud -1 m/s con 10 personas de la planta



```
Escalón de -1 m/s, con 10 personas.
Media de las últimas 20 muestras = -0.979 m/s
ts = 0.980 s
c_inf_sim: -0.97614
erpp: -0.021
ts_sim: 1.05
erpp_sim: -0.023863
```

Ejercicio 13

Mida en régimen permanente tanto la acción de control como la velocidad de la escalera, subiendo y bajando en diferentes condiciones de carga de la escalera, para consignas de ± 1 m/s. Relacione los incrementos o decrementos de acción de control con el peso de las personas presentes en la escalera.

```
% En la GUI se tiene la señal real de control. En este caso son:
gui_control = [
    1, 4.69; ... 1m/s, 0 personas
    1, 5.16; ... 1m/s, 5 personas
    1, 5.62; ... 1m/s, 10 personas
    -1, -4.68; ... -1m/s, 0 personas
    -1, -4.20; ... -1m/s, 5 personas
    -1, -3.73; ... -1m/s, 10 personas
];
for i = 1:length(entradas)
    for j = 1:length(perturbaciones)
        fprintf("Consigna " + entradas(i) + " m/s, carga de " +
perturbaciones_names(j) + " - Sin alinealidades: " + ...
            "\n\tSalida de: " + salidas_SA(end, 3*(i - 1) + j) + " m/s en RP" + ...
            "\n\tAcción de: " + acciones_SA(end, 3*(i - 1) + j) + " V en RP\n")

        fprintf("Consigna " + entradas(i) + " m/s, carga de " +
perturbaciones_names(j) + " - Con alinealidades:" + ...
```

```

        "\n\tSalida de: " + salidas_CA(end, 3*(i - 1) + j) + " m/s en RP" + ...
        "\n\tAcción de: " + acciones_CA(end, 3*(i - 1) + j) + " V en RP\n")

        fprintf("Consigna " + gui_control(3*(i - 1) + j, 1) + " m/s, carga de " +
perturbaciones_names(j) + " - Resultado GUI:" + ...
        "\n\tSalida de: " + c_inf(3*(i-1) + j) + " m/s en RP (media ultimas 20
muestras)" + ...
        "\n\tAcción de: " + gui_control(3*(i - 1) + j, 2) + " V en última
muestra\n \n ")
    end
end
end

```

Consigna 1 m/s, carga de 0 personas - Sin alinealidades:

Salida de: 0.97004 m/s en RP

Acción de: 4.866 V en RP

Consigna 1 m/s, carga de 0 personas - Con alinealidades:

Salida de: 0.97004 m/s en RP

Acción de: 4.866 V en RP

Consigna 1 m/s, carga de 0 personas - Resultado GUI:

Salida de: 0.97 m/s en RP (media ultimas 20 muestras)

Acción de: 4.69 V en última muestra

Consigna 1 m/s, carga de 5 personas - Sin alinealidades:

Salida de: 0.967 m/s en RP

Acción de: 5.3526 V en RP

Consigna 1 m/s, carga de 10 personas - Sin alinealidades:

Salida de: 0.96395 m/s en RP

Acción de: 5.8391 V en RP

Consigna 1 m/s, carga de 5 personas - Resultado GUI:

Salida de: 0.968 m/s en RP (media ultimas 20 muestras)

Acción de: 5.16 V en última muestra

Consigna -1 m/s, carga de 0 personas - Sin alinealidades:

Salida de: -0.97004 m/s en RP

Acción de: -4.866 V en RP

Consigna -1 m/s, carga de 5 personas - Sin alinealidades:

Salida de: -0.97309 m/s en RP

Acción de: -4.3795 V en RP

Consigna 1 m/s, carga de 10 personas - Resultado GUI:

Salida de: 0.964 m/s en RP (media ultimas 20 muestras)

Acción de: 5.62 V en última muestra

Consigna 1 m/s, carga de 5 personas - Con alinealidades:

Salida de: 0.967 m/s en RP

Acción de: 5.3526 V en RP

Consigna -1 m/s, carga de 10 personas - Sin alinealidades:

Salida de: -0.97614 m/s en RP

Acción de: -3.8929 V en RP

Consigna -1 m/s, carga de 0 personas - Resultado GUI:

Salida de: -0.971 m/s en RP (media ultimas 20 muestras)

Acción de: -4.68 V en última muestra

Consigna 1 m/s, carga de 10 personas - Con alinealidades:

Salida de: 0.96395 m/s en RP

Acción de: 5.8391 V en RP

Consigna -1 m/s, carga de 0 personas - Con alinealidades:
 Salida de: -0.97004 m/s en RP
 Acción de: -4.866 V en RP
 Consigna -1 m/s, carga de 5 personas - Resultado GUI:
 Salida de: -0.976 m/s en RP (media ultimas 20 muestras)
 Acción de: -4.2 V en última muestra

Consigna -1 m/s, carga de 5 personas - Con alinealidades:
 Salida de: -0.97309 m/s en RP
 Acción de: -4.3795 V en RP
 Consigna -1 m/s, carga de 10 personas - Con alinealidades:
 Salida de: -0.97614 m/s en RP
 Acción de: -3.8929 V en RP
 Consigna -1 m/s, carga de 10 personas - Resultado GUI:
 Salida de: -0.979 m/s en RP (media ultimas 20 muestras)
 Acción de: -3.73 V en última muestra

Ejercicio 14

Extraiga cuantas reflexiones estime oportunas sobre el cumplimiento de los objetivos y sobre la comparativa entre especificaciones de diseño y resultados teóricos, simulados y reales.

Fase 4

Ejercicio 15

Calcule la FdeT del regulador así como sus unidades, aplicando la teoría que corresponda y justificando adecuadamente la variante de PID elegida, el resultado debe presentarlo de la forma factorizada, polinomial y con potencias negativas en z. Presente la ecuación en diferencias del regulador. Obtenga las ganancias proporcional, integral y derivativa, así como los tiempos de acción integral y derivativa del PID (parámetros de sintonía).

```
clearvars -except Kg pg Ts aproximacion decimales errores_SA acciones_SA salidas_SA
errores_CA acciones_CA salidas_CA eje_tiempos_fase3 errores_SA_f5 acciones_SA_f5
salidas_SA_f5 errores_CA_f5 acciones_CA_f5 salidas_CA_f5 eje_tiempos_fase5
tiempos_sim salidas_sim;

G = tf(Kg, [1 -pg]);
retardo = tf(1, [1 0], Ts);
BoG = series(c2d(G, Ts), retardo)

BoG =

      0.02596
-----
z^2 - 0.8698 z

Sample time: 0.035 seconds
Discrete-time transfer function.
Model Properties

[NumBoG, DenBoG] = tfdata(BoG, 'v');
```

```
% Especificaciones
```

```
ts = 1.12;
```

```
erpp = 0;
```

```
% Calculo diferencias de grados
```

```
grado_Den_BoG = length(DenBoG) - find(DenBoG ~= 0, 1, 'first');
```

```
grado_Num_BoG = length(NumBoG) - find(NumBoG ~= 0, 1, 'first');
```

```
dif_grados_G = grado_Den_BoG - grado_Num_BoG;
```

```
retardos = dif_grados_G - 1;
```

```
% Calculo de parámetros. Como no puede haber oscilación, el polo debe ser
```

```
% real y positivo
```

```
ts_prima = ts - retardos*Ts;
```

```
pm = exp(Ts * aproximacion / ts_prima);
```

```
km = (1-pm)*(1-erpp);
```

```
% Primer intento: Controlador I
```

```
baseControladorI = Ts/2 * tf([1 1], [1 -1], Ts);
```

```
figure
```

```
rlocus(baseControladorI * BoG)
```

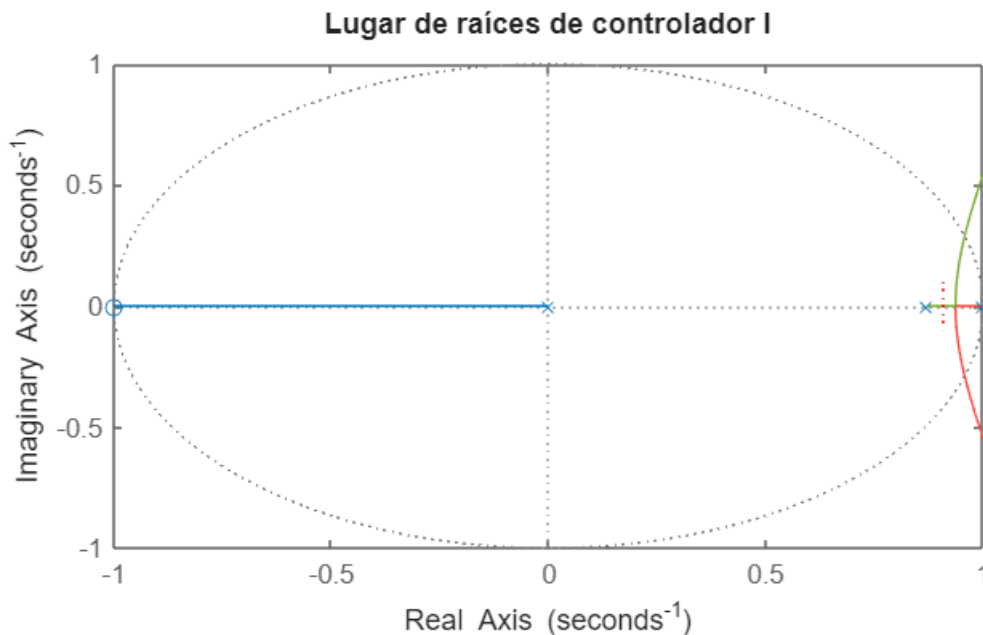
```
hold on
```

```
plot(pm * [1 1], 0.1 * [1 -1], 'r:')
```

```
title("Lugar de raíces de controlador I")
```

```
xlim([-1 1])
```

```
ylim([-1 1])
```



```
% El valor pertenece al lugar de raíces, pero no sería dominante.
```

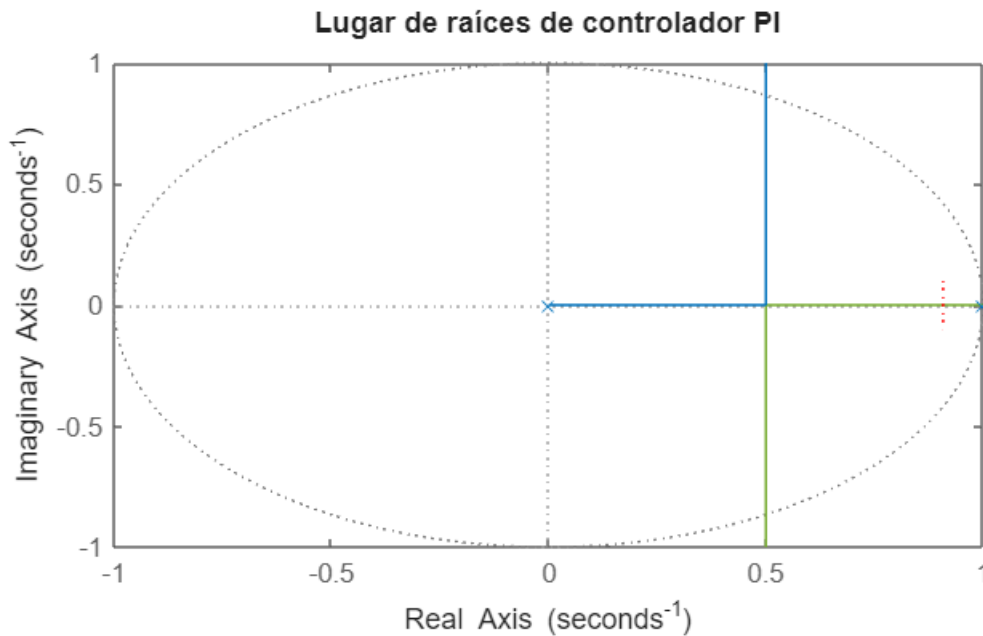
```
% Segundo intento: Controlador PI. Se coloca el cero para cancelar el polo
```

```

% de la planta
ceroPI = max(abs(pole(BoG)));
baseControladorPI = tf([1 -ceroPI], [1 -1], Ts);

figure
rlocus(minreal(baseControladorPI * BoG))
hold on
plot(pm * [1 1], 0.1 * [1 -1], 'r:')
title("Lugar de raíces de controlador PI")
xlim([-1 1])
ylim([-1 1])
hold off

```



```

% Es posible -> Obtener parámetros
[~, pX, kX] = zpndata(minreal(baseControladorPI * BoG), 'v');
K = prod(abs(pm - pX))/kX; % Condición modular

F = K * baseControladorPI;
[NumF, ~] = tfdata(F, 'v'); q0 = NumF(1); q1 = NumF(2);

% q0 = kp + ki*Ts/2
% q1 = -kp + ki*Ts/2
Ki = (q0 + q1)/Ts;
Kp = (q0 - q1)/2;

fprintf("Parámetros del PI: \n\tKp = %.0" + decimales + "f \n\tKi = %.0" +
decimales + "f \n\tq0 = %.0" + decimales + "f\n\tq1 = %.0" + decimales +
"f\n\nControlador:", Kp, Ki, q0, q1)

```

Parámetros del PI:

Kp = 3.01503
Ki = 12.00057
q0 = 3.22504
q1 = -2.80502

Controlador:

F

F =

$$\frac{3.225 z - 2.805}{z - 1}$$

Sample time: 0.035 seconds
Discrete-time transfer function.
Model Properties

F = zpk(F)

F =

$$\frac{3.225 (z-0.8698)}{(z-1)}$$

Sample time: 0.035 seconds
Discrete-time zero/pole/gain model.
Model Properties

Ejercicio 16

Calcule la FdeT del sistema y obtenga el/los polo/s del sistema.

M = minreal(feedback(series(F, BoG), 1))

M =

$$\frac{0.083731}{(z-0.09224)(z-0.9078)}$$

Sample time: 0.035 seconds
Discrete-time zero/pole/gain model.
Model Properties

```
[Zm, Pm, ~] = zpkdata(M, 'v');  
fprintf("Ceros del sistema: [ "); fprintf("%.0" + decimales + "f ", Zm);  
fprintf("]");
```

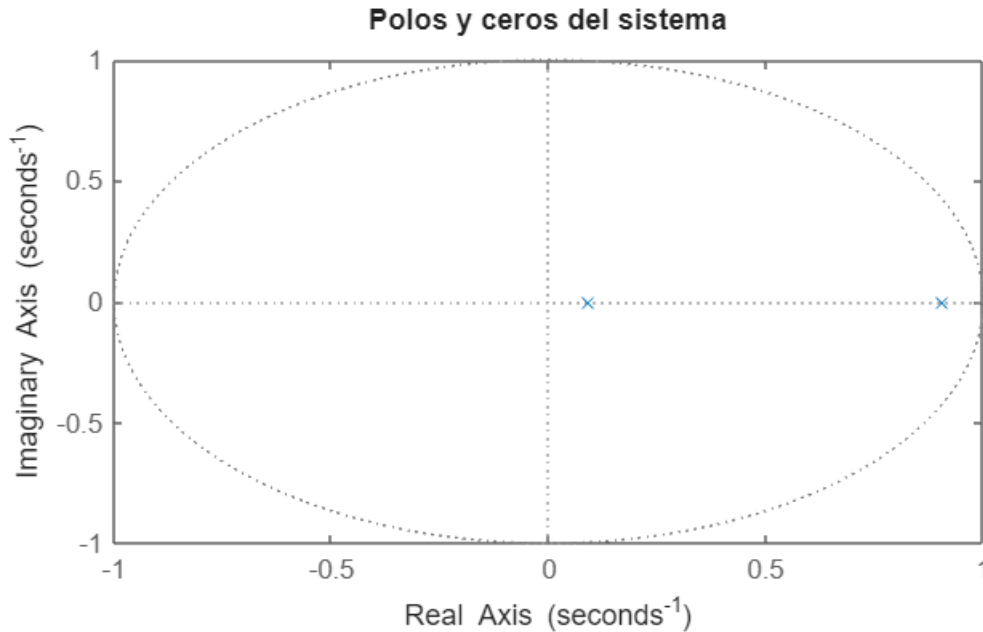
Ceros del sistema: []

```
fprintf("Polos del sistema: [ "); fprintf("%.0" + decimales + "f ", Pm);  
fprintf("]");
```

Polos del sistema: [0.09224 0.90776]

figure

```
pzplot(M)
title("Polos y ceros del sistema")
```



Ejercicio 17

Caracterice teóricamente, tanto en régimen transitorio como permanente, la salida del sistema ante entradas escalón de ± 1 m/s

```
retardos = length(Pm) - 1;

ks = ceil(aproximacion/log(max(abs(Pm)))) + retardos;
ts = ks * Ts;

for entrada = [1 -1]
    c_inf = entrada * dcgain(M);

    fprintf("Caracterización ante escalón de %.02f m/s: \n\tc_inf = %.0" +
    decimales + "f m/s\n\tks = %d\n\tts = %.0" + decimales + "f s\n\terpp = %.0" +
    decimales + "f\n\n", entrada, c_inf, ks, ts, abs(entrada - c_inf))
end
```

Caracterización ante escalón de 1.00 m/s:

```
c_inf = 1.00000 m/s
ks = 32
ts = 1.12000 s
erpp = 0.00000
```

Caracterización ante escalón de -1.00 m/s:

```
c_inf = -1.00000 m/s
ks = 32
ts = 1.12000 s
erpp = 0.00000
```

Ejercicio 18

Calcule en régimen permanente tanto la acción de control como la velocidad de la escalera subiendo y bajando en diferentes condiciones de carga de la escalera, para consignas de ± 1 m/s. Relacione los incrementos o decrementos de acción de control con el peso de las personas presentes en la escalera.

```
entradas          = [1 -1];
perturbaciones    = [0 -0.1 -0.2];
perturbaciones_names = [" 0 personas", " 5 personas", "10 personas"];

M_r = M;
M_p = zpk(minreal(feedback(1, series(F, BoG))));
K0_ref = dcgain(M_r);
K0_pert = dcgain(M_p);
K0_g = dcgain(BoG);

% Cálculo de error y acción en régimen permanente para distintas perturbaciones

for i = 1:length(entradas)
    fprintf("\nPara escalón de %.2fV:\n", entradas(i))
    for j = 1:length(perturbaciones)
        r = entradas(i);
        p = perturbaciones(j);

        c = K0_ref * r + K0_pert * p;
        e = r - c;
        a = (c - p) / K0_g;

        fprintf("\tCon %s:\tc_inf=%.0" + decimales + "f m/s\terpp=%.0" + decimales
+ "f\ta_inf=%.0" + decimales + "f\n", perturbaciones_names(j), c, abs(e), a);
    end
end
```

Para escalón de 1.00V:

| | | | |
|------------------|-------------------|--------------|---------------|
| Con 0 personas: | c_inf=1.00000 m/s | erpp=0.00000 | a_inf=5.01630 |
| Con 5 personas: | c_inf=1.00000 m/s | erpp=0.00000 | a_inf=5.51793 |
| Con 10 personas: | c_inf=1.00000 m/s | erpp=0.00000 | a_inf=6.01956 |

Para escalón de -1.00V:

| | | | |
|------------------|--------------------|--------------|----------------|
| Con 0 personas: | c_inf=-1.00000 m/s | erpp=0.00000 | a_inf=-5.01630 |
| Con 5 personas: | c_inf=-1.00000 m/s | erpp=0.00000 | a_inf=-4.51467 |
| Con 10 personas: | c_inf=-1.00000 m/s | erpp=0.00000 | a_inf=-4.01304 |

Fase 5

Ejercicio 19

Simule en Simulink el sistema de control con y sin alinealidades. Mida las características de las respuestas (tanto el régimen permanente como el transitorio) ante entradas escalón de ± 1 m/s. Analice las diferencias de las medidas respecto de los cálculos teóricos.

```
if ~exist("acciones_SA_f5", 'var') % evita simular todo el rato
```



```

    open_system("Fase3/modeloSimulink"); % se usa el modelo de la fase 3 editando
    el controlador

    % Controlador
    % SA ≡ Sin Alinealidades, CA ≡ Con Alinealidades
    [N,D] = tfdata(F, 'v');
    set_param("modeloSimulink/Controlador - SA", 'Numerator', '[' +
    strjoin(string(N)) + ']', 'Denominator', '[' + strjoin(string(D)) + ']')
    set_param("modeloSimulink/Controlador - CA", 'Numerator', '[' +
    strjoin(string(N)) + ']', 'Denominator', '[' + strjoin(string(D)) + ']')

    % Planta
    [N,D] = tfdata(G, 'v');
    set_param("modeloSimulink/Planta - CA", 'Numerator', '[' + strjoin(string(N)) +
    ']', 'Denominator', '[' + strjoin(string(D)) + ']')
    set_param("modeloSimulink/Planta - CA", 'Numerator', '[' + strjoin(string(N)) +
    ']', 'Denominator', '[' + strjoin(string(D)) + ']')

    sim("modeloSimulink")
    errores_SA_f5 = zeros(length(SA.signals(1).values),1);
    acciones_SA_f5 = zeros(length(SA.signals(1).values),1);
    salidas_SA_f5 = zeros(length(SA.signals(1).values),1);
    errores_CA_f5 = zeros(length(CA.signals(1).values),1);
    acciones_CA_f5 = zeros(length(CA.signals(1).values),1);
    salidas_CA_f5 = zeros(length(CA.signals(1).values),1);
    eje_tiempos_fase5 = CA.time; % Igual para Con/Sin Alinealidades y las distintas
    señales

    % Simular
    for i = 1:length(entradas) % ± 1m/s
        set_param("modeloSimulink/Ref - SA", 'After', string(entradas(i)))
        set_param("modeloSimulink/Ref - CA", 'After', string(entradas(i)))

        for j = 1:length(perturbaciones)
            perturbacion = perturbaciones(j);
            set_param("modeloSimulink/Personas - SA", 'After', string(perturbacion))
            set_param("modeloSimulink/Personas - CA", 'After', string(perturbacion))
            sim("modeloSimulink")

            errores_SA_f5(:, 3*(i - 1) + j) = SA.signals(1).values;
            acciones_SA_f5(:, 3*(i - 1) + j) = SA.signals(2).values;
            salidas_SA_f5(:, 3*(i - 1) + j) = SA.signals(3).values;

            errores_CA_f5(:, 3*(i - 1) + j) = CA.signals(1).values;
            acciones_CA_f5(:, 3*(i - 1) + j) = CA.signals(2).values;
            salidas_CA_f5(:, 3*(i - 1) + j) = CA.signals(3).values;
        end
    end
end

```

```

% Representar
for i = 1:length(entradas)
    for j = 1:length(perturbaciones)
        % Sin alinealidades
        figure
        sgtitle("Consigna " + entradas(i) + " m/s, carga de " +
perturbaciones_names(j) + " - Sin alinealidades")
        subplot(1, 2, 1)
        hold on
        stairs(eje_tiempos_fase5, errores_SA_f5(:, 3*(i - 1) + j))
        stairs(eje_tiempos_fase5, salidas_SA_f5(:, 3*(i - 1) + j))
        legend("error [m/s]", "salida [m/s]", Location="east")
        xlabel("tiempo (s)")
        ylabel("[m/s]")
        subplot(1, 2, 2)
        stairs(eje_tiempos_fase5, acciones_SA_f5(:, 3*(i - 1) + j), Color="m")
        legend("acción [V]", Location="east")
        xlabel("tiempo (s)")
        ylabel("[V]")

        % Con alinealidades
        figure
        sgtitle("Consigna " + entradas(i) + " m/s, carga de " +
perturbaciones_names(j) + " - Con alinealidades")
        subplot(1, 2, 1)
        hold on
        stairs(eje_tiempos_fase5, errores_CA_f5(:, 3*(i - 1) + j))
        stairs(eje_tiempos_fase5, salidas_CA_f5(:, 3*(i - 1) + j))
        legend("error [m/s]", "salida [m/s]", Location="east")
        xlabel("tiempo (s)")
        ylabel("[m/s]")
        subplot(1, 2, 2)
        stairs(eje_tiempos_fase5, acciones_CA_f5(:, 3*(i - 1) + j), Color="m")
        legend("acción [V]", Location="east")
        xlabel("tiempo (s)")
        ylabel("[V]")

        % Diferencia entre Sin y Con alinealidades
        figure
        sgtitle("Consigna " + entradas(i) + " m/s, carga de " +
perturbaciones_names(j) + " - Diferencia")
        subplot(1, 2, 1)
        hold on
        stairs(eje_tiempos_fase5, errores_CA_f5(:, 3*(i - 1) + j) - errores_SA_f5
(:, 3*(i - 1) + j))
        stairs(eje_tiempos_fase5, salidas_CA_f5(:, 3*(i - 1) + j) - salidas_SA_f5
(:, 3*(i - 1) + j))
        legend("error [m/s]", "salida [m/s]", Location="southeast")
        xlabel("tiempo (s)")
        ylabel("[m/s]")

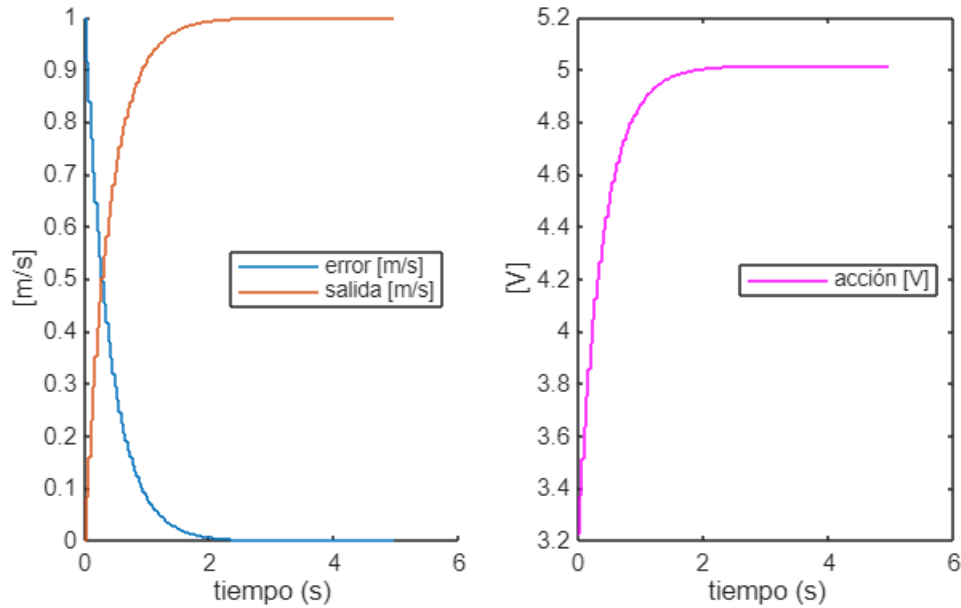
```

```

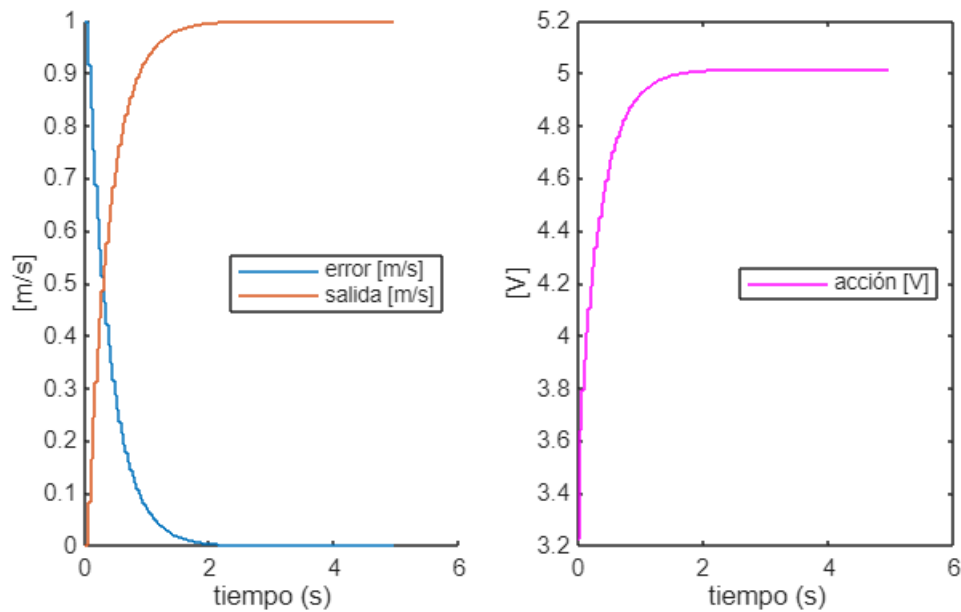
subplot(1, 2, 2)
stairs(eje_tiempos_fase5, acciones_CA_f5(:, 3*(i - 1) + j) - acciones_SA_f5
(:, 3*(i - 1) + j), Color="m")
legend("acción [V]", Location="southeast")
xlabel("tiempo (s)")
ylabel("[V]")
end
end

```

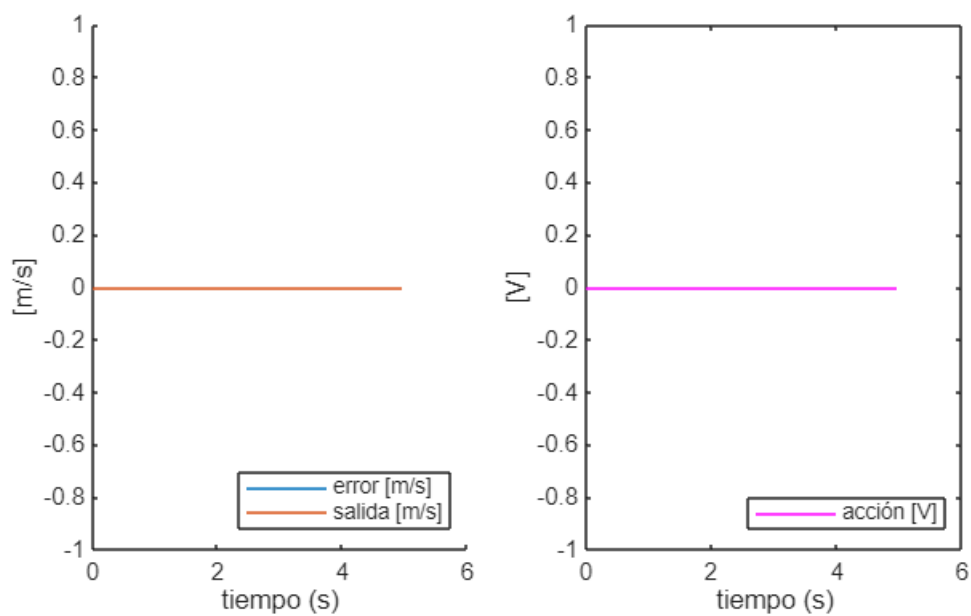
Consigna 1 m/s, carga de 0 personas - Sin alinealidades



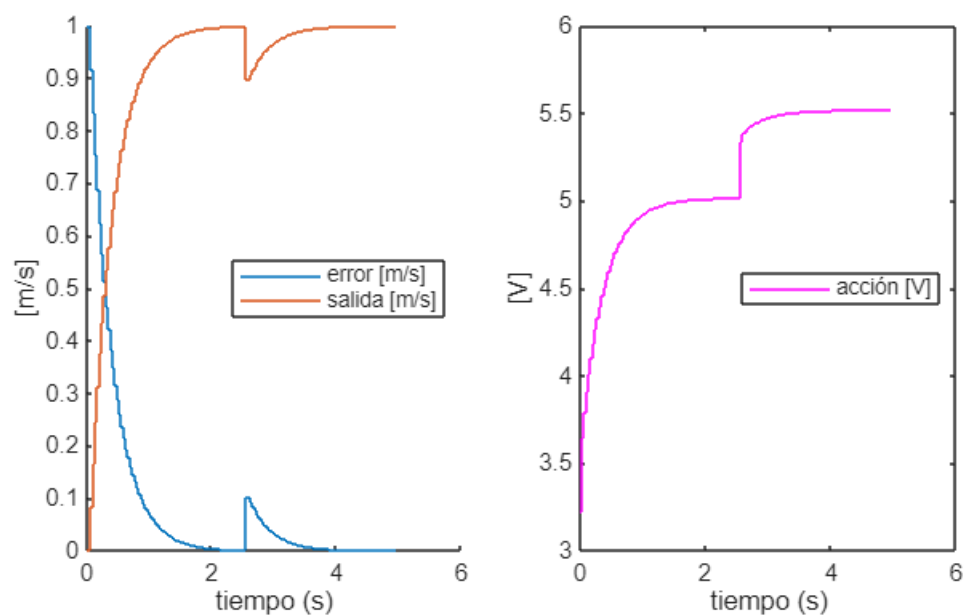
Consigna 1 m/s, carga de 0 personas - Con alinealidades



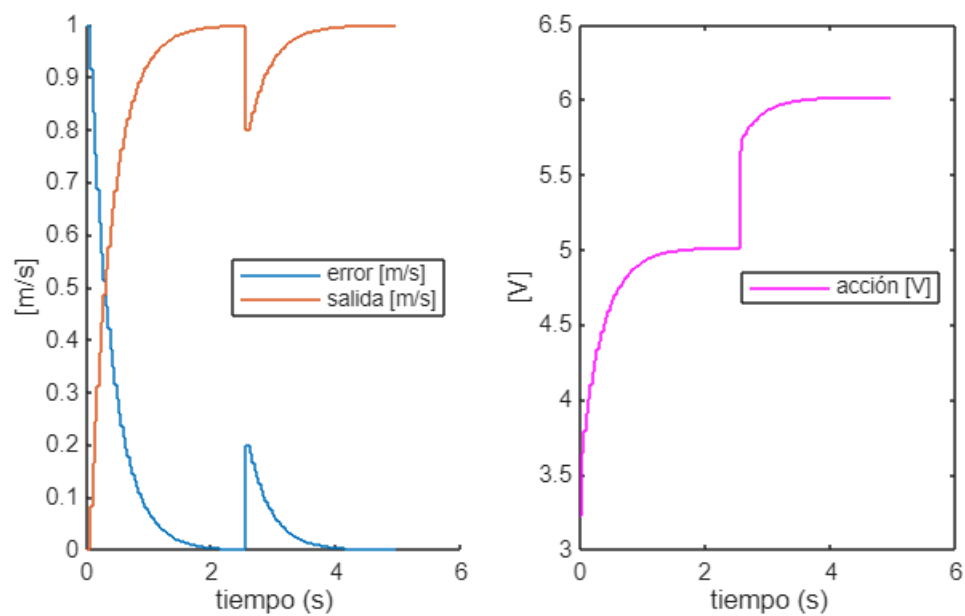
Consigna 1 m/s, carga de 0 personas - Diferencia



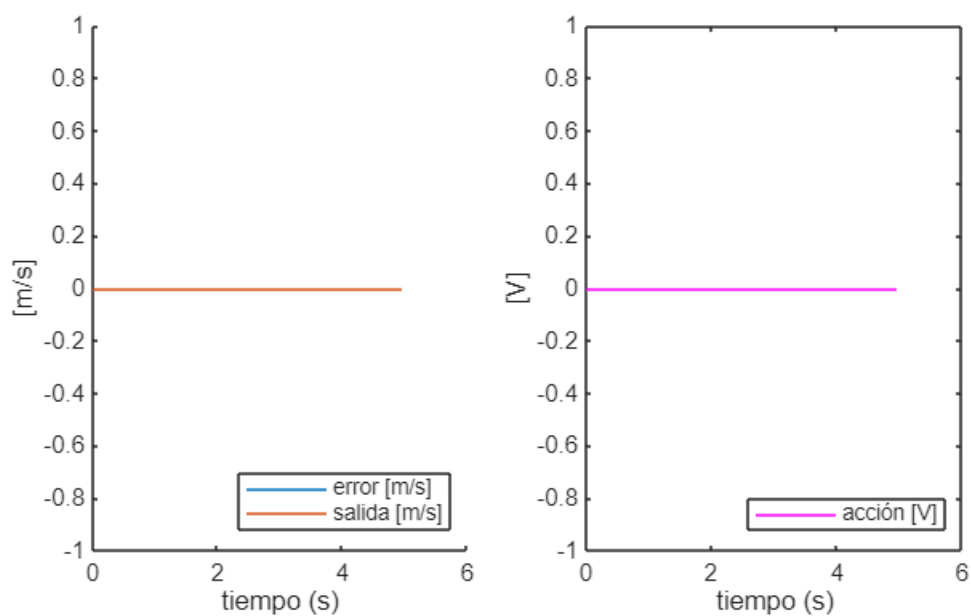
Consigna 1 m/s, carga de 5 personas - Sin alinealidades



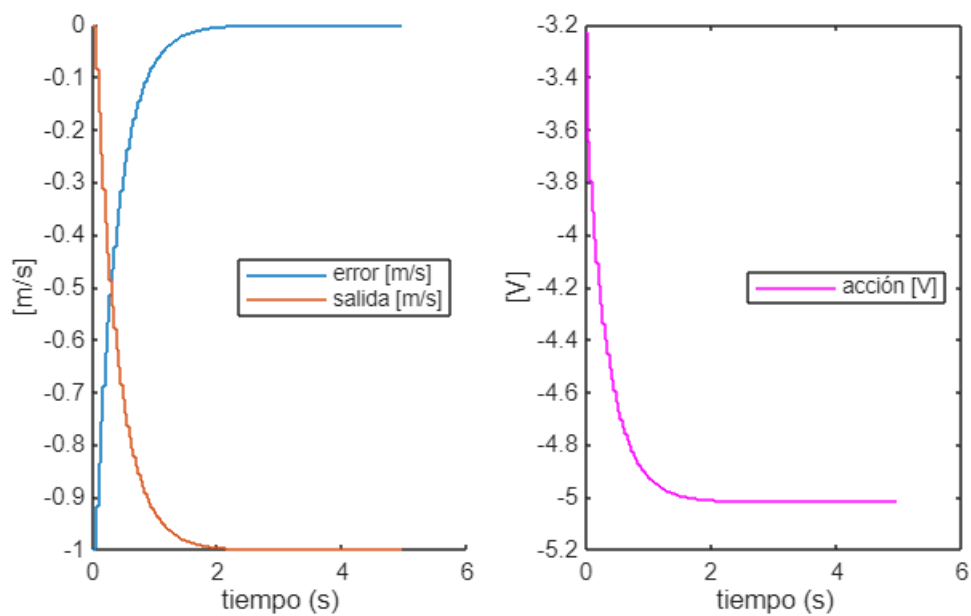
Consigna 1 m/s, carga de 10 personas - Sin alinealidades



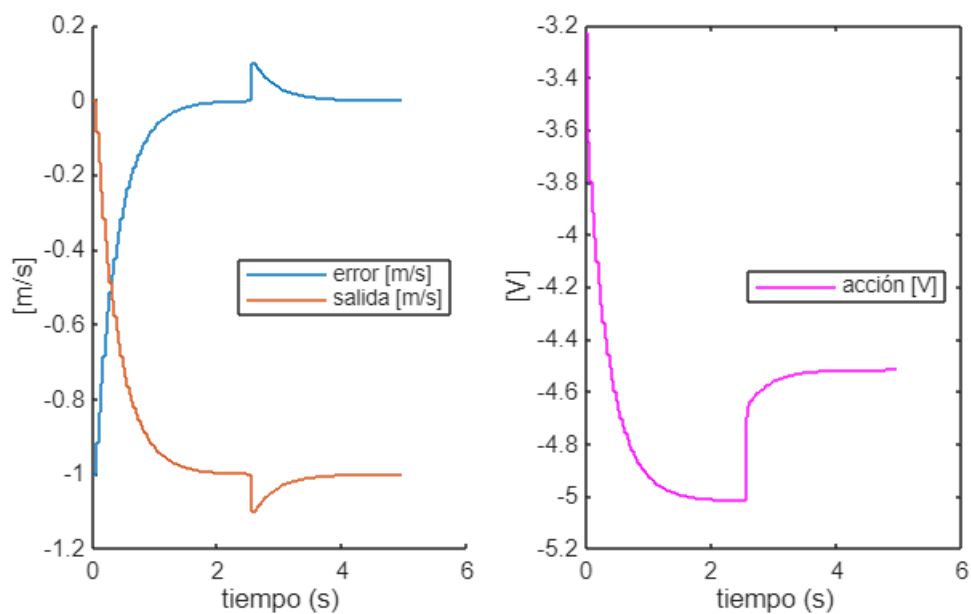
Consigna 1 m/s, carga de 5 personas - Diferencia



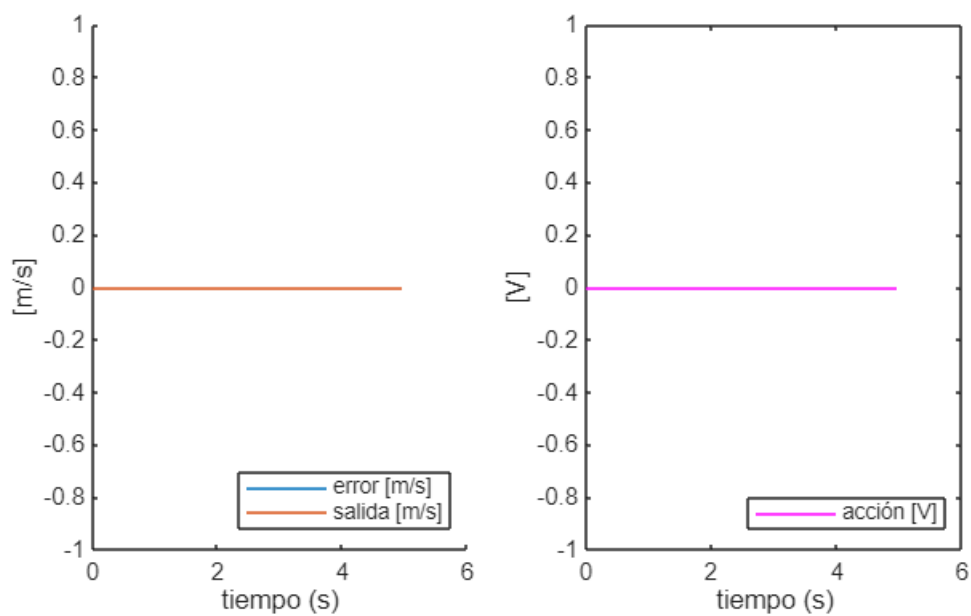
Consigna -1 m/s, carga de 0 personas - Sin alinealidades



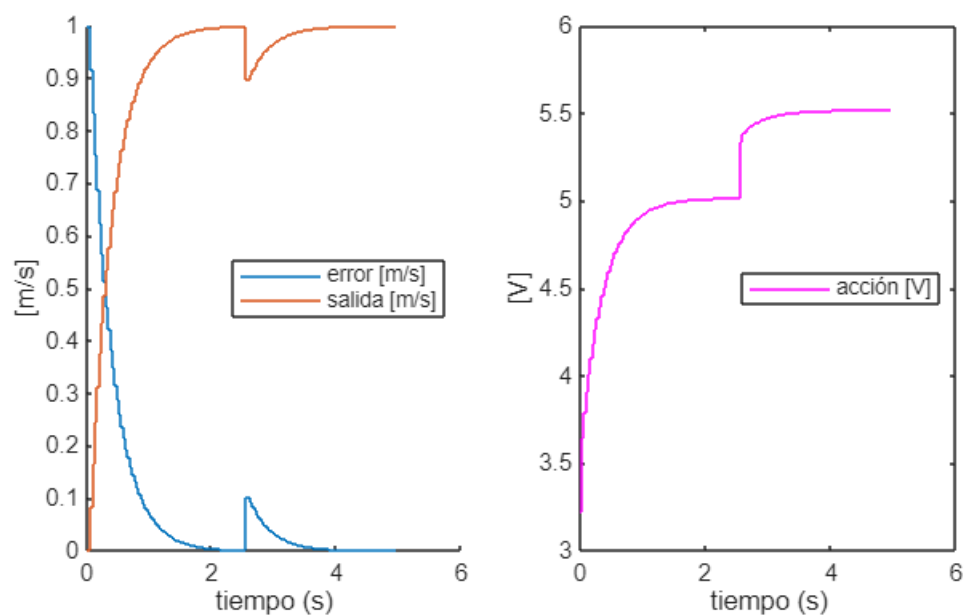
Consigna -1 m/s, carga de 5 personas - Sin alinealidades



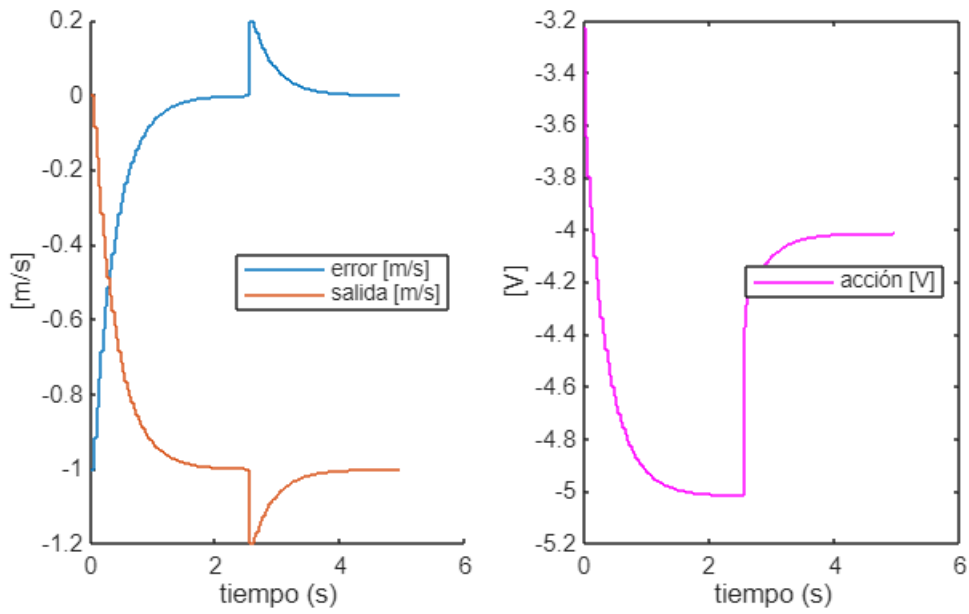
Consigna 1 m/s, carga de 10 personas - Diferencia



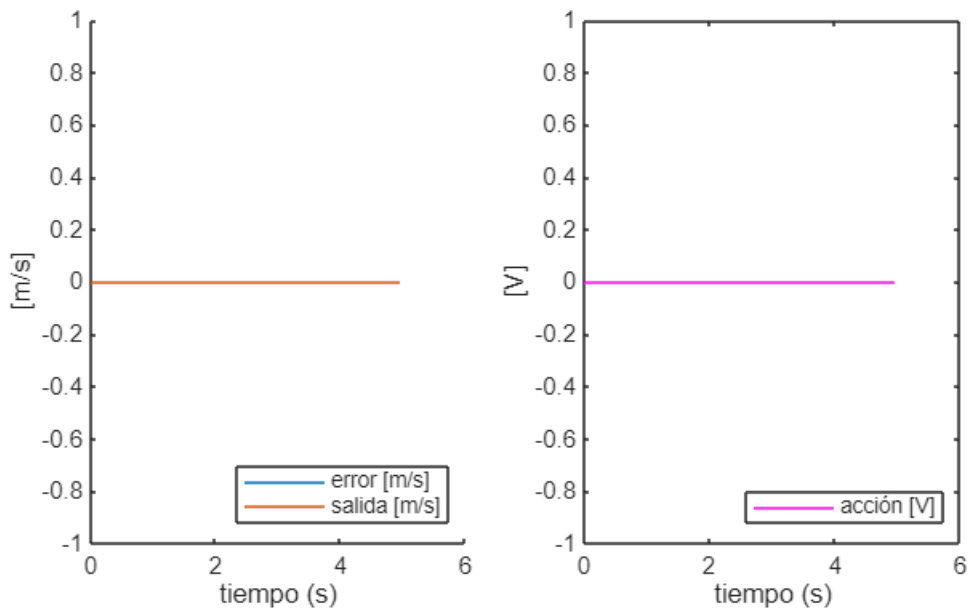
Consigna 1 m/s, carga de 5 personas - Con alinealidades



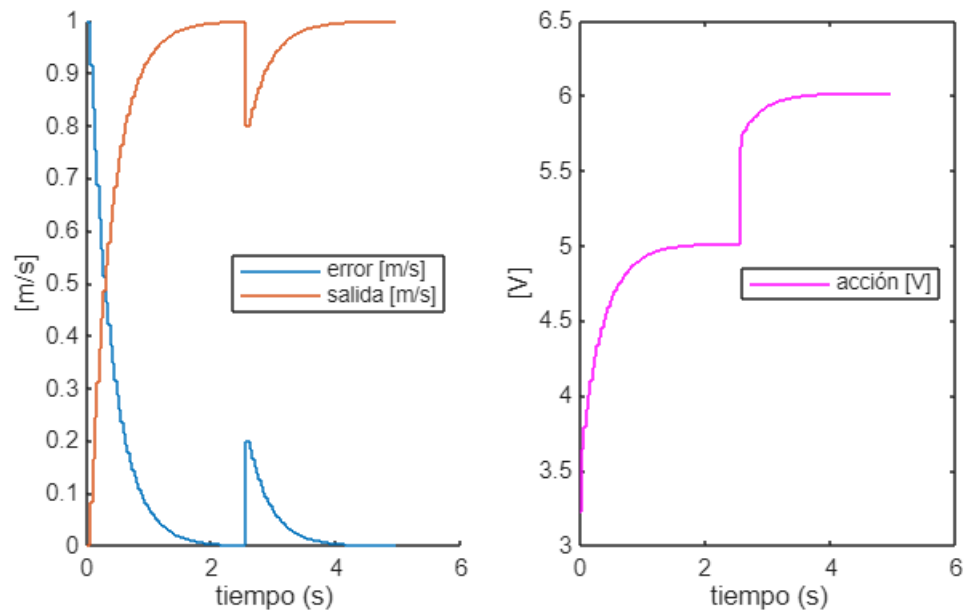
Consigna -1 m/s, carga de 10 personas - Sin alinealidades



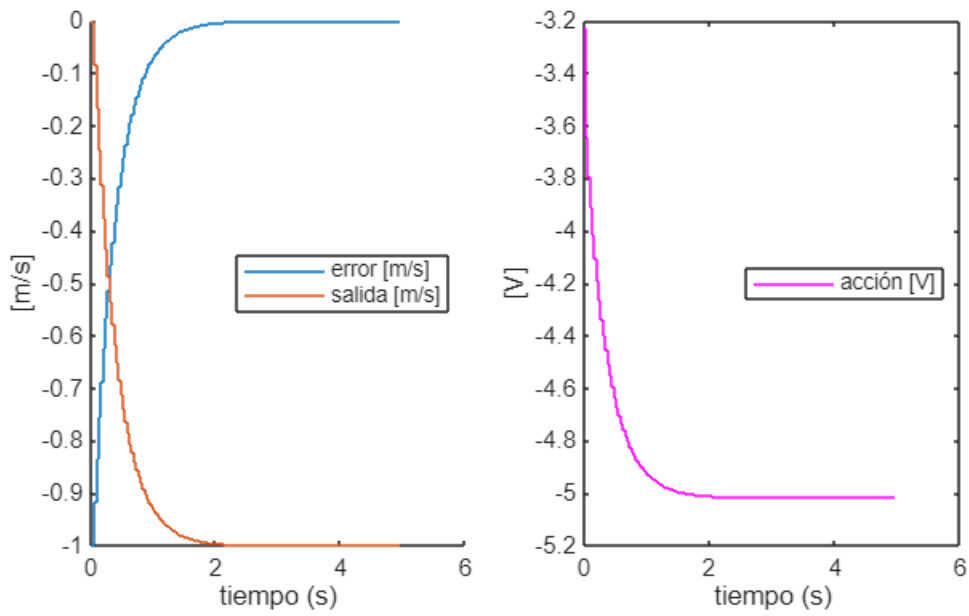
Consigna -1 m/s, carga de 0 personas - Diferencia



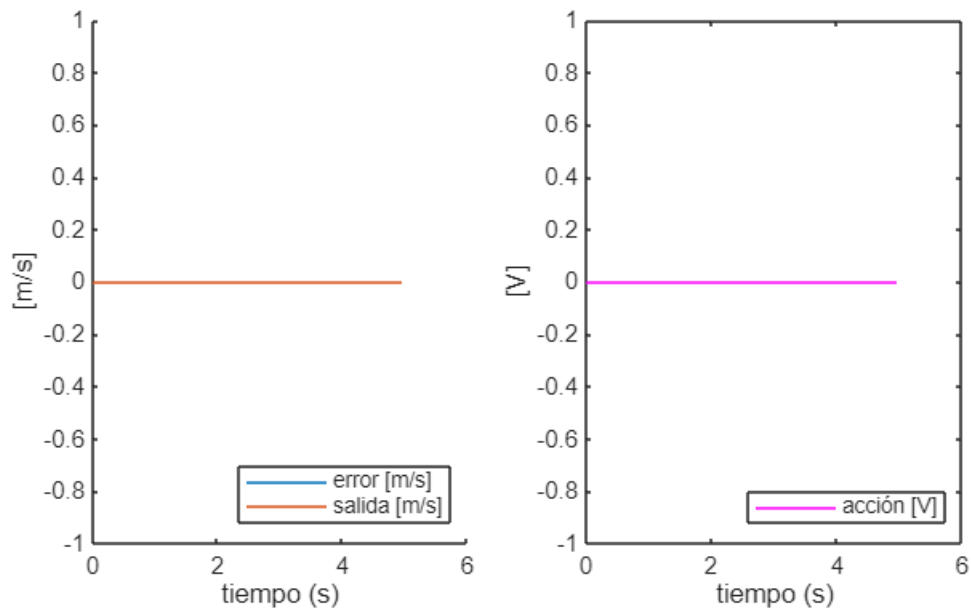
Consigna 1 m/s, carga de 10 personas - Con alinealidades



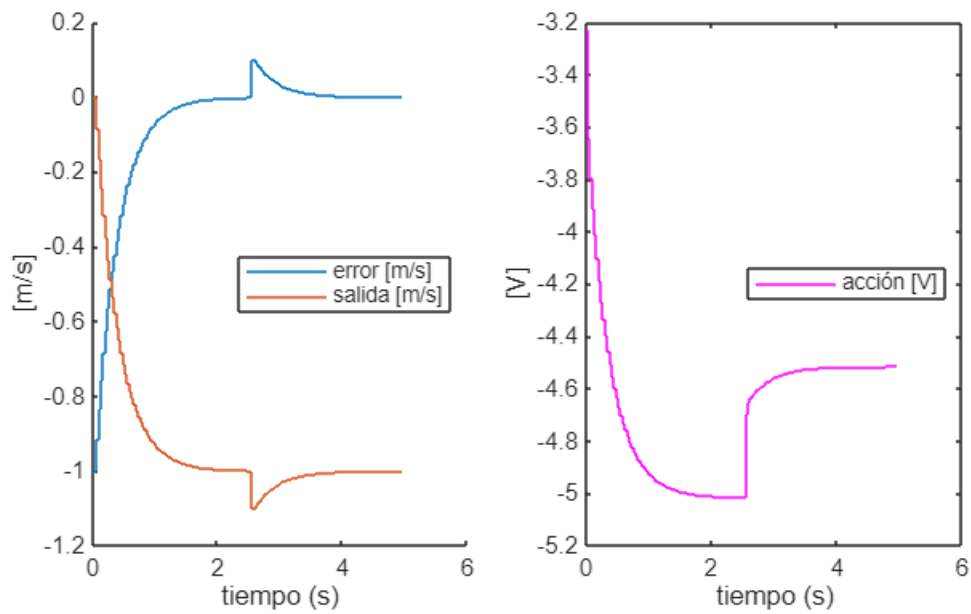
Consigna -1 m/s, carga de 0 personas - Con alinealidades



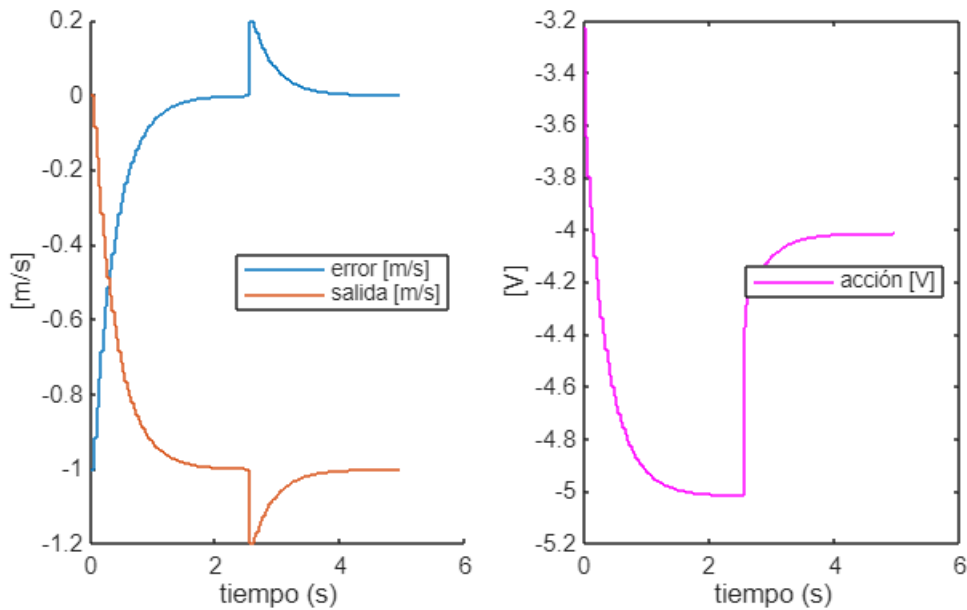
Consigna -1 m/s, carga de 5 personas - Diferencia



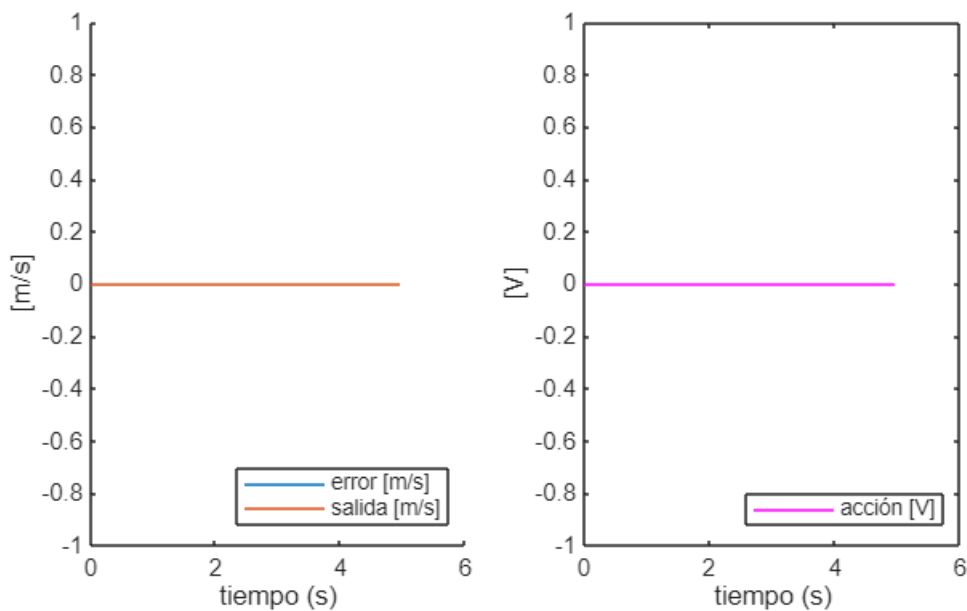
Consigna -1 m/s, carga de 5 personas - Con alinealidades



Consigna -1 m/s, carga de 10 personas - Con alinealidades

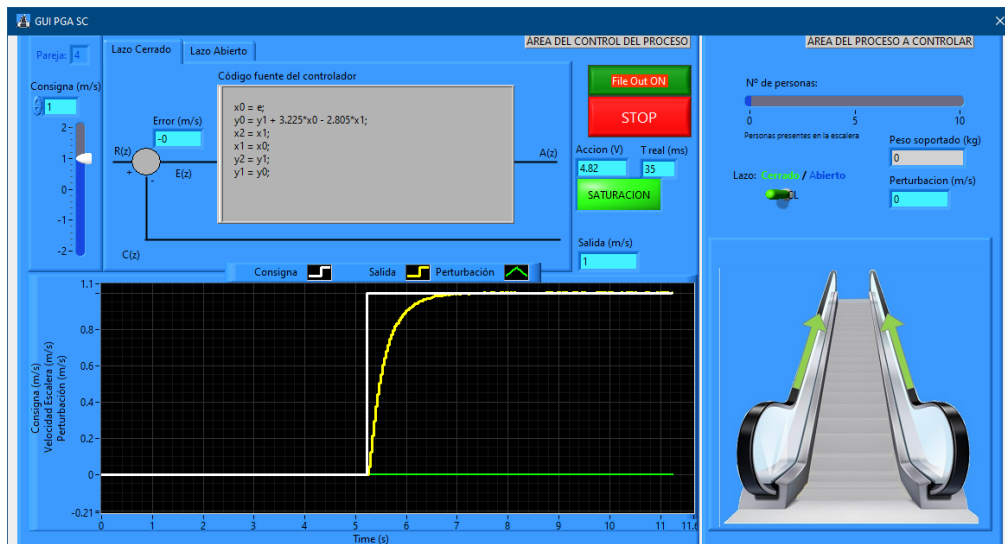


Consigna -1 m/s, carga de 10 personas - Diferencia



Ejercicio 20

Mediante el GUI, obtenga la captura de la respuesta del sistema para una consigna de 1 m/s (subiendo) y sin personas transportadas (sin carga). Adjunte una captura de pantalla de la ventana de gestión del sistema de control en la que se aprecie con la suficiente nitidez el código fuente del regulador (Figura 2.b). Mida las características de la respuesta y coteje los resultados con los de la simulación del ejercicio anterior, así como con los cálculos teóricos realizados en la fase de diseño del regulador.

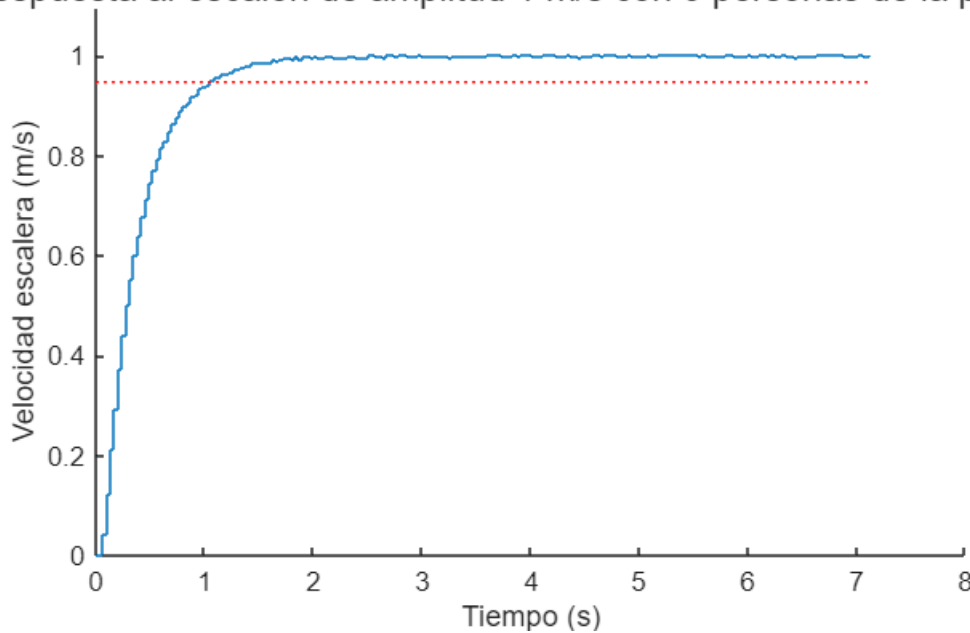


```
ficheros = dir("Fase5/ficherosGUI/positivo");

c_inf = zeros(1,length(ficheros) - 2);

procesar_resultados("Fase5/ficherosGUI/positivo/" + ficheros(3).name, 0.035, 20, 3);
```

Respuesta al escalón de amplitud 1 m/s con 0 personas de la planta



Escalón de 1 m/s, con 0 personas.
 Media de las últimas 20 muestras = 0.997 m/s
 $t_s = 1.050$ s

```
for i = 1:length(ficheros)
    fichero = ficheros(i).name;

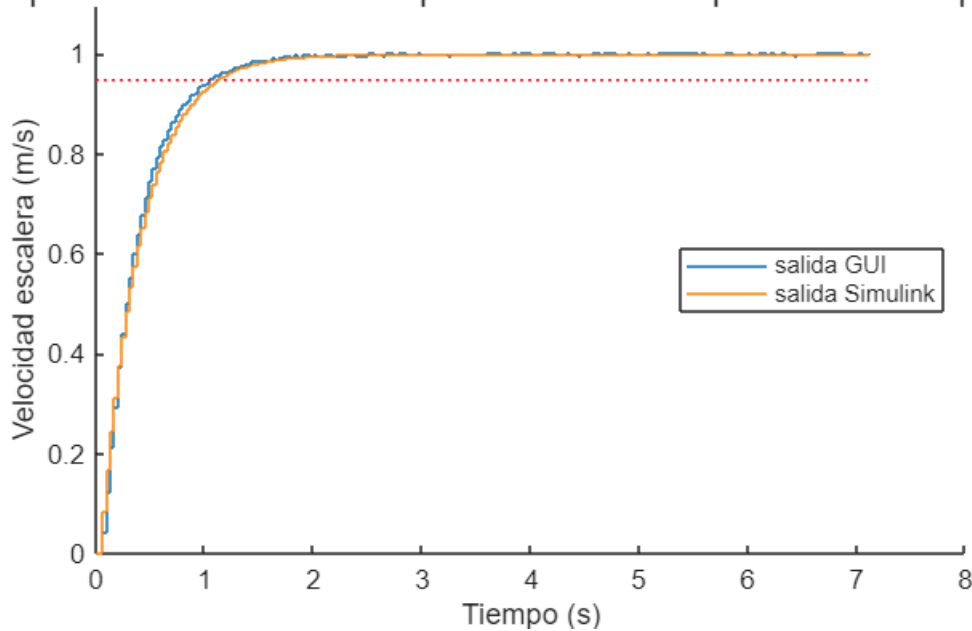
    % Saltar '.' y '..'
    if strcmp(fichero, '.') || strcmp(fichero, '..')
        continue;
    end
end
```

end

```
[~, erpp, c_inf(i-2)] = procesar_resultados("Fase5/ficherosGUI/positivo/" +  
fichero, 0.035, 20, 3, salidas_CA_f5(:, i - 2));  
ts_sim = (find(salidas_SA_f5(:, i-2) >= 0.95*salidas_CA_f5(end, i-2), 1,  
"first") - 1) * 0.035;  
erpp_sim = entradas(1) - salidas_CA_f5(end, i-2);  
fprintf("\tc_inf_sim: " + salidas_SA_f5(end, i - 2) + ...  
        "\n\terpp: " + erpp + ...  
        "\n\tts_sim: " + ts_sim + ...  
        "\n\terpp_sim: " + erpp_sim + ...  
        "\n\taccion_sim: " + acciones_CA_f5(end, i-2))
```

end

Respuesta al escalón de amplitud 1 m/s con 0 personas de la planta



Escalón de 1 m/s, con 0 personas.

Media de las últimas 20 muestras = 0.997 m/s

ts = 1.050 s

c_inf_sim: 1

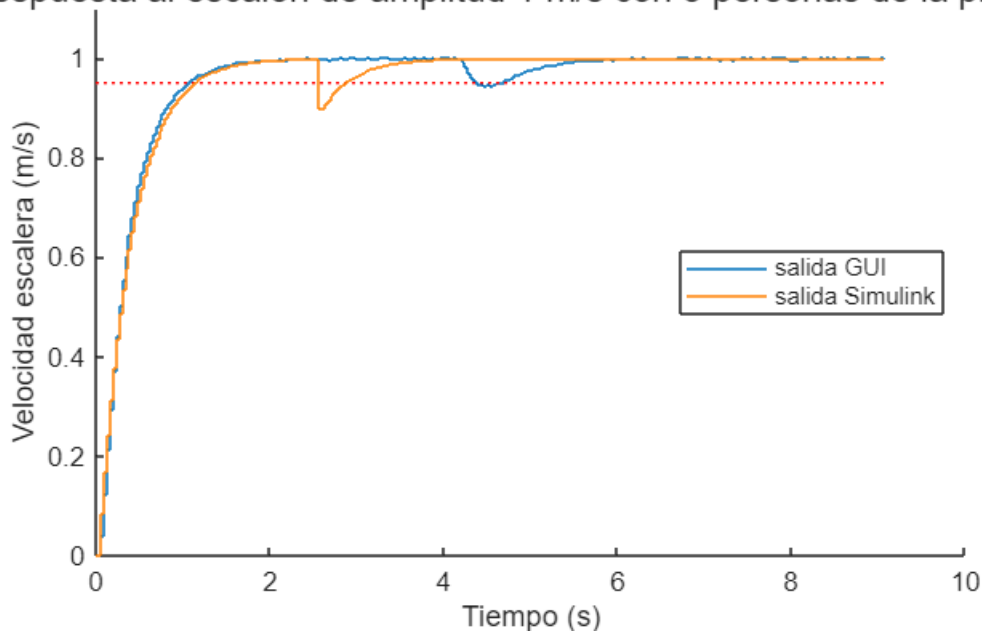
erpp: 0.003

ts_sim: 1.155

erpp_sim: 1.2002e-06

accion_sim: 5.0163

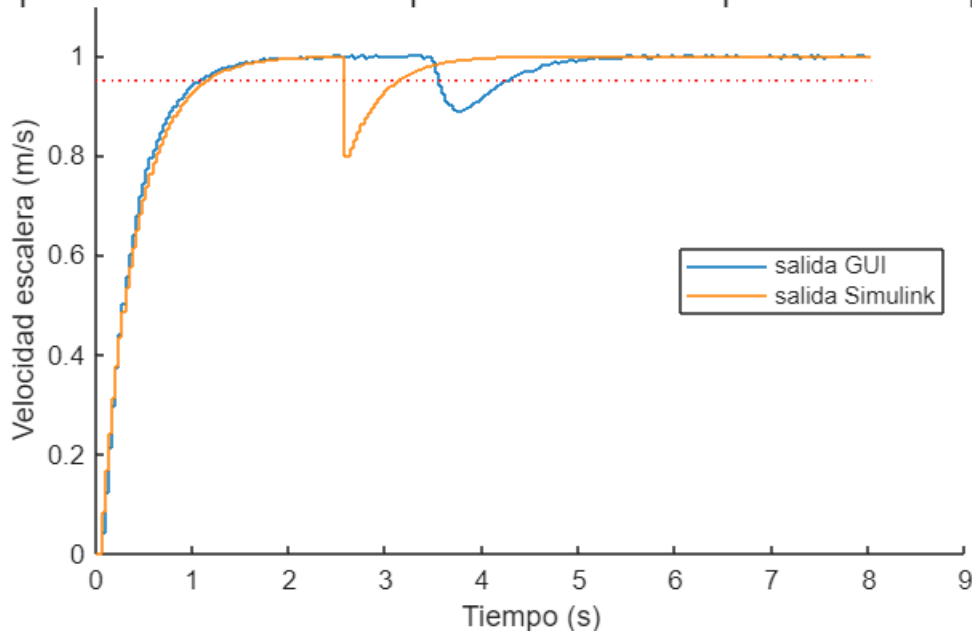
Respuesta al escalón de amplitud 1 m/s con 5 personas de la planta



Escalón de 1 m/s, con 5 personas.

Media de las últimas 20 muestras = 1.000 m/s
 ts = 1.050 s
 c_inf_sim: 0.99986
 erpp: 0
 ts_sim: 1.155
 erpp_sim: 0.00014144
 accion_sim: 5.5177

Respuesta al escalón de amplitud 1 m/s con 10 personas de la planta

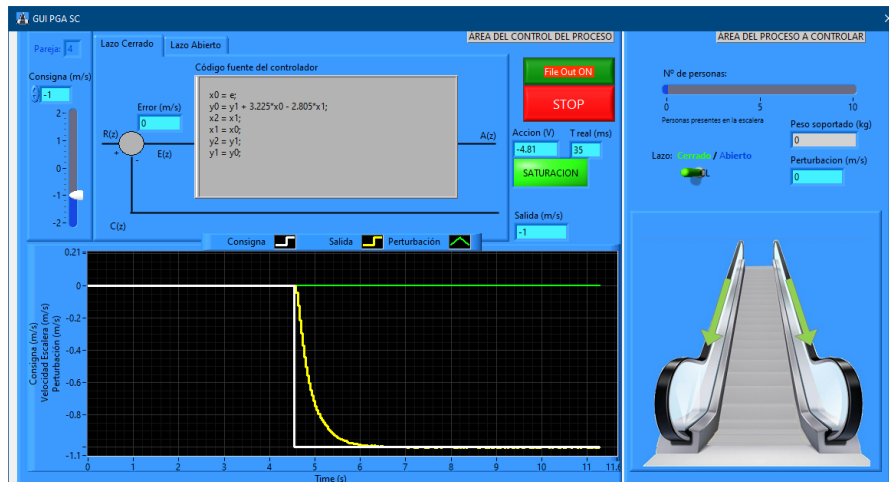


Escalón de 1 m/s, con 10 personas.

Media de las últimas 20 muestras = 1.000 m/s
 ts = 1.085 s
 c_inf_sim: 0.99972
 erpp: 0
 ts_sim: 1.155
 erpp_sim: 0.00028168
 accion_sim: 6.0192

Ejercicio 21

Igual que el ejercicio 20 pero con consigna de -1 m/s (bajando).



```
ficheros = dir("Fase5/ficherosGUI/negativo");

idx = length(c_inf);
c_inf = [c_inf zeros(1, length(ficheros) - 2)];

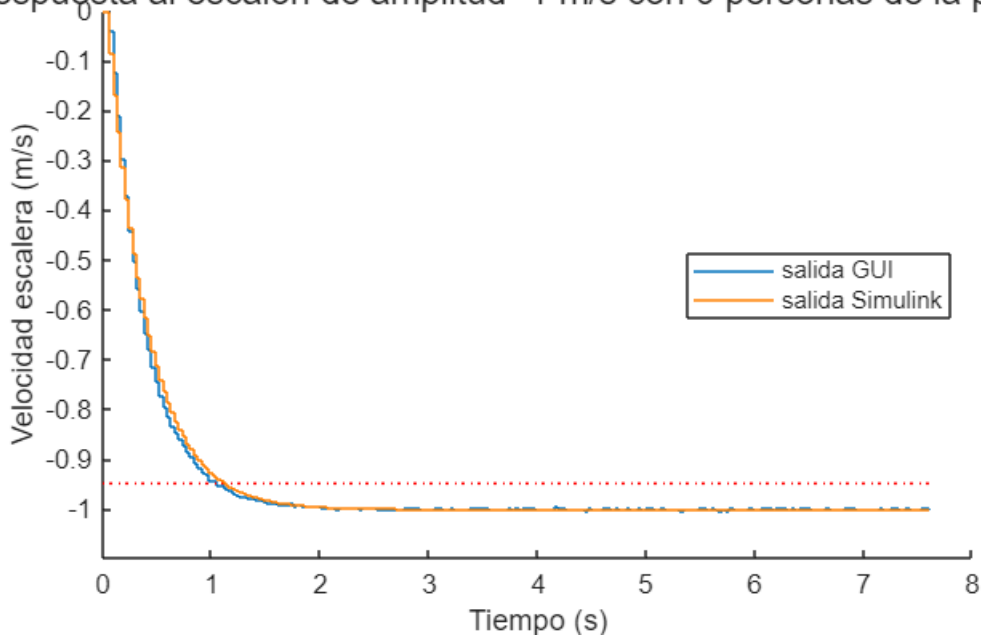
for i = 1:length(ficheros)
    fichero = ficheros(i).name;

    % Saltar '.' y '..'
    if strcmp(fichero, '.') || strcmp(fichero, '..')
        continue;
    end

    [~, ~, c_inf(idx + i - 2)] = procesar_resultados("Fase5/ficherosGUI/negativo/"
+ fichero, 0.035, 20, 3, salidas_CA_f5(:, idx + i - 2));
    ts_sim = (find(salidas_SA_f5(:, idx + i - 2) <= 0.95*salidas_CA_f5(end, idx + i
- 2), 1, "first") - 1) * 0.035;
    erpp_sim = entradas(2) - salidas_CA_f5(end, idx + i - 2);
    fprintf("\tc_inf_sim: " + salidas_SA_f5(end, idx + i - 2) + ...
        "\n\terpp: " + erpp + ...
        "\n\tts_sim: " + ts_sim + ...
        "\n\terpp_sim: " + erpp_sim + ...
        "\n\taccion_sim: " + acciones_CA_f5(end, idx + i - 2))

end
```

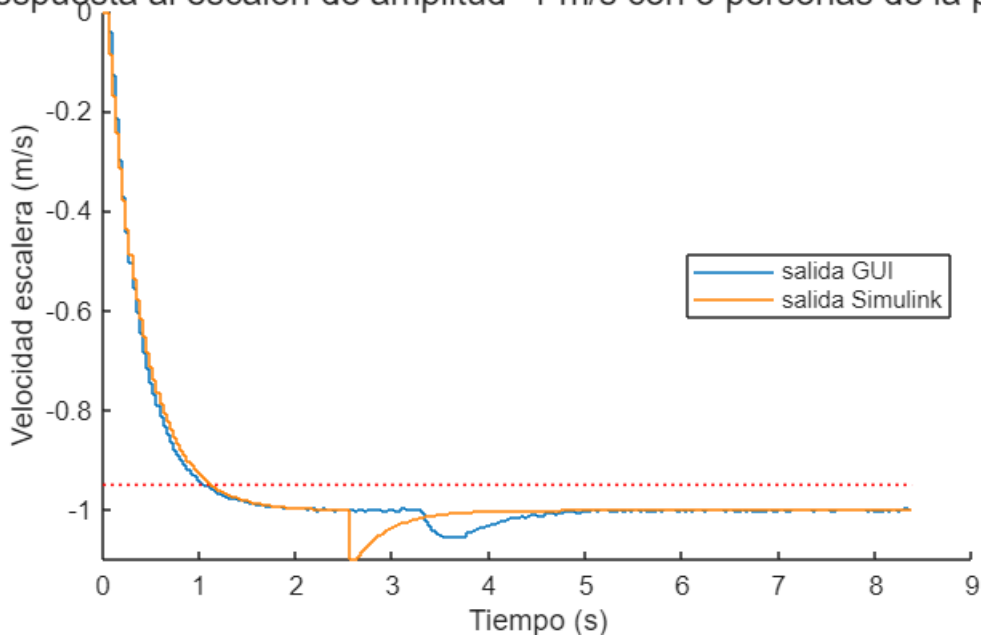
Respuesta al escalón de amplitud -1 m/s con 0 personas de la planta



Escalón de -1 m/s, con 0 personas.

Media de las últimas 20 muestras = -0.999 m/s
 ts = 1.050 s
 c_inf_sim: -1
 erpp: 0
 ts_sim: 1.155
 erpp_sim: -1.2002e-06
 accion_sim: -5.0163

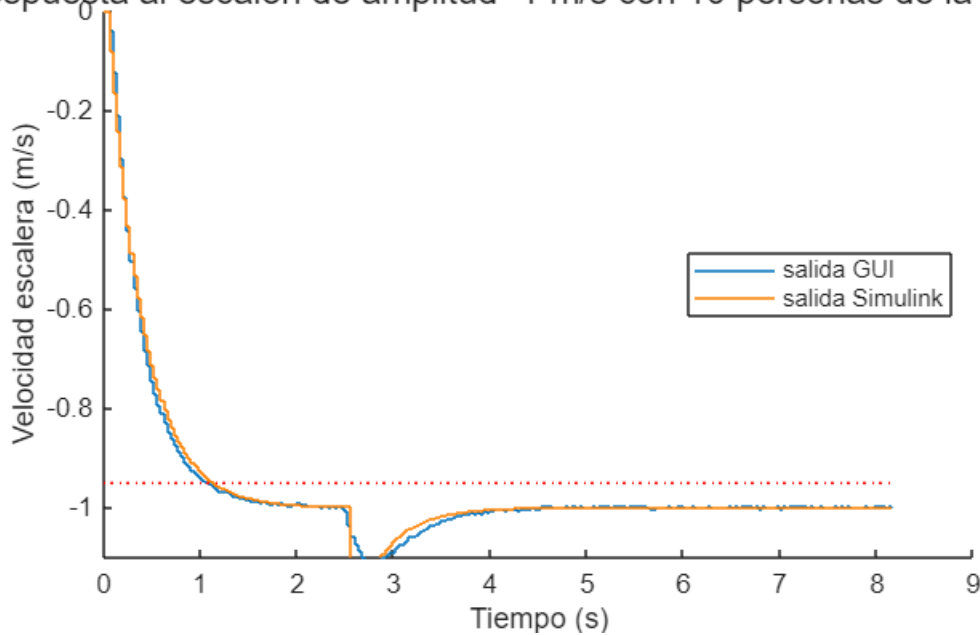
Respuesta al escalón de amplitud -1 m/s con 5 personas de la planta



Escalón de -1 m/s, con 5 personas.

Media de las últimas 20 muestras = -1.000 m/s
 ts = 1.050 s
 c_inf_sim: -1.0001
 erpp: 0
 ts_sim: 1.155
 erpp_sim: 0.00013904
 accion_sim: -4.5148

Respuesta al escalón de amplitud -1 m/s con 10 personas de la planta



Escalón de -1 m/s, con 10 personas.

Media de las últimas 20 muestras = -1.002 m/s

ts = 1.085 s

c_inf_sim: -1.0003

erpp: 0

ts_sim: 1.155

erpp_sim: 0.00027928

accion_sim: -4.0134

Ejercicio 22

Mida en régimen permanente tanto la acción de control como la velocidad de la escalera, subiendo y bajando en diferentes condiciones de carga de la escalera, para consignas de ± 1 m/s. Relacione los incrementos o decrementos de acción de control con el peso de las personas presentes en la escalera.

% En la GUI se tiene la señal real de control. En este caso son:

```
gui_control = [
    1, 4.82; ... 1m/s, 0 personas
    1, 5.29; ... 1m/s, 5 personas
    1, 5.78; ... 1m/s, 10 personas
    -1, -4.81; ... -1m/s, 0 personas
    -1, -4.32; ... -1m/s, 5 personas
    -1, -3.84; ... -1m/s, 10 personas
];
for i = 1:length(entradas)
    for j = 1:length(perturbaciones)
        fprintf("Consigna " + entradas(i) + " m/s, carga de " +
perturbaciones_names(j) + " - Sin alinealidades: " + ...
        "\n\tSalida de: " + salidas_SA_f5(end, 3*(i - 1) + j) + " m/s en RP" +
...
        "\n\tAcción de: " + acciones_SA_f5(end, 3*(i - 1) + j) + " V en RP\n")
    end
end
```

```

        fprintf("Consigna " + entradas(i) + " m/s, carga de " +
perturbaciones_names(j) + " - Con alinealidades:" + ...
        "\n\tSalida de: " + salidas_CA_f5(end, 3*(i - 1) + j) + " m/s en RP" +
...
        "\n\tAcción de: " + acciones_CA_f5(end, 3*(i - 1) + j) + " V en RP\n")

        fprintf("Consigna " + gui_control(3*(i - 1) + j, 1) + " m/s, carga de " +
perturbaciones_names(j) + " - Resultado GUI:" + ...
        "\n\tSalida de: " + c_inf(3*(i-1) + j) + " m/s en RP (media ultimas 20
muestras)" + ...
        "\n\tAcción de: " + gui_control(3*(i - 1) + j, 2) + " V en última
muestra\n \n ")
    end
end

```

```

Consigna 1 m/s, carga de 0 personas - Sin alinealidades:
    Salida de: 1 m/s en RP
    Acción de: 5.0163 V en RP
Consigna 1 m/s, carga de 0 personas - Con alinealidades:
    Salida de: 1 m/s en RP
    Acción de: 5.0163 V en RP
Consigna 1 m/s, carga de 0 personas - Resultado GUI:
    Salida de: 0.997 m/s en RP (media ultimas 20 muestras)
    Acción de: 4.82 V en última muestra

```

```

Consigna 1 m/s, carga de 5 personas - Sin alinealidades:
    Salida de: 0.99986 m/s en RP
    Acción de: 5.5177 V en RP
Consigna 1 m/s, carga de 10 personas - Sin alinealidades:
    Salida de: 0.99972 m/s en RP
    Acción de: 6.0192 V en RP
Consigna 1 m/s, carga de 5 personas - Resultado GUI:
    Salida de: 1 m/s en RP (media ultimas 20 muestras)
    Acción de: 5.29 V en última muestra

```

```

Consigna -1 m/s, carga de 0 personas - Sin alinealidades:
    Salida de: -1 m/s en RP
    Acción de: -5.0163 V en RP
Consigna -1 m/s, carga de 5 personas - Sin alinealidades:
    Salida de: -1.0001 m/s en RP
    Acción de: -4.5148 V en RP
Consigna 1 m/s, carga de 10 personas - Resultado GUI:
    Salida de: 1 m/s en RP (media ultimas 20 muestras)
    Acción de: 5.78 V en última muestra

```

```

Consigna 1 m/s, carga de 5 personas - Con alinealidades:
    Salida de: 0.99986 m/s en RP
    Acción de: 5.5177 V en RP
Consigna -1 m/s, carga de 10 personas - Sin alinealidades:
    Salida de: -1.0003 m/s en RP
    Acción de: -4.0134 V en RP
Consigna -1 m/s, carga de 0 personas - Resultado GUI:
    Salida de: -0.999 m/s en RP (media ultimas 20 muestras)
    Acción de: -4.81 V en última muestra

```

Consigna 1 m/s, carga de 10 personas - Con alinealidades:
Salida de: 0.99972 m/s en RP
Acción de: 6.0192 V en RP
Consigna -1 m/s, carga de 0 personas - Con alinealidades:
Salida de: -1 m/s en RP
Acción de: -5.0163 V en RP
Consigna -1 m/s, carga de 5 personas - Resultado GUI:
Salida de: -1 m/s en RP (media ultimas 20 muestras)
Acción de: -4.32 V en última muestra

Consigna -1 m/s, carga de 5 personas - Con alinealidades:
Salida de: -1.0001 m/s en RP
Acción de: -4.5148 V en RP
Consigna -1 m/s, carga de 10 personas - Con alinealidades:
Salida de: -1.0003 m/s en RP
Acción de: -4.0134 V en RP
Consigna -1 m/s, carga de 10 personas - Resultado GUI:
Salida de: -1.002 m/s en RP (media ultimas 20 muestras)
Acción de: -3.84 V en última muestra