

Identificación y Control Adaptativo

Trabajo Práctico 4

Truls Carlson

21 de abril de 2025

Índice

1. Enunciado del Problema	1
2. Ejercicios	2
2.1. No adaptativo genérico	2
2.1.1. Resultados	4
2.2. Separación de dinámicas	6
2.2.1. Resultados	6
2.3. Efecto integral	7
2.3.1. Resultados	7
2.4. Adaptativo indirecto	8
2.4.1. Resultados	9
2.5. Adaptativo directo	10
2.5.1. Resultados	11
3. Control Adaptativo Incremental	12
Referencias	14

1. Enunciado del Problema

Continuamos con el tanque dado del Trabajo Práctico 1. Ahora vamos a analizar diferentes variantes del algoritmo de control STR.

2. Ejercicios

2.1. No adaptativo genérico

Para implementar un algoritmo de control STR no adaptativo genérico, usamos parámetros fijos. Es decir, en lugar de adaptar los parámetros se usan parámetros ya calculados.

El proceso se expresa como se da en [1],

$$A(q)y_k = B(q)(u_k + v_k) \quad (1)$$

Con la siguiente ley de control lineal STR,

$$R(q)u_k = T(q)r_k - S(q)y_k \quad (2)$$

Se pueden encontrar dos de los polinomios a través de la ecuación *Diofantina*,

$$A_{lc} = A(q)R(q) + B(q)S(q) \quad (3)$$

donde A_{lc} es el polinomio característico en lazo cerrado deseado y A y B son los polinomios desde la discretización que hicimos en TP1 [2],

$$G(z) = \frac{B}{A} = \frac{b_1 q^{-1}}{1 + a_1 q^{-1}} \quad (4)$$

También suponemos,

$$R(q) = r_0 + r_1 q^{-1}, \quad S(q) = s_0 \quad (5)$$

para que el grado de S satisfaga $\deg(S) \leq \deg(R)$. Con $A_{lc} = 1 + \alpha q^{-1}$ se puede escribir la ecuación 3 como,

$$\begin{aligned} 1 + \alpha q^{-1} &= (1 + a_1 q^{-1})(r_0 + r_1 q^{-1}) + (b_1 q^{-1})s_0 \\ 1 + \alpha q^{-1} &= r_0 + r_1 q^{-1} + a_1 r_0 q^{-1} + a_1 r_1 q^{-2} + b_1 s_0 q^{-1} \\ 1 + \alpha q^{-1} &= r_0 + (r_1 + a_1 r_0 + b_1 s_0)q^{-1} + a_1 r_1 q^{-2} \end{aligned} \quad (6)$$

que da,

$$r_0 = 1, \quad r_1 + a_1 r_0 + b_1 s_0 = \alpha, \quad a_1 r_1 = 0 \quad (7)$$

Inmediatamente podemos ver que r_1 debe ser 0. Entonces obtenemos la expresión por s_0 como,

$$s_0 = \frac{\alpha - a_1}{b_1} \quad (8)$$

Desde *Caso 1* en [1] tenemos que

$$t_0 = \frac{1 + \alpha}{b_1} \quad (9)$$

y que la ley de control se puede expresar como

$$u_k = t_0 r_k - s_0 y_k \quad (10)$$

Con el siguiente código implementé el control STR no adaptativo genérico,

```

1 % Parametros del tanque
2 r_0 = 0.05;
3 r_o = 0.0254;
4 g = 9.81;
5 theta = 60;
6 a = pi*r_o^2;
7 F_in = 0.0035;
8
9 % Punto de trabajo
10 h_bar = 0.2;
11 k_bar = F_in / (a*sqrt(2*9.81*h_bar));
12
13 % Linearizacion desde TP1
14 syms h k
15
16 F_out_lin = k*a*sqrt(2*g);
17
18 f = (F_in - F_out_lin*sqrt(h)) / (pi * ((h^2)/3 + 2*r_o*h/sqrt(3) + r_o^2));
19
20 dfdh = diff(f, h);
21 dfdh_eval = double(vpa(subs(dfdh, [h, k], [h_bar, k_bar])));
22
23 dfdk = diff(f, k);
24 dfdk_eval = double(vpa(subs(dfdk, [h, k], [h_bar, k_bar])));
25
26 % Discretizacion ZOH
27 Ts = 1; % Tiempo del muestreo
28
29 G = tf([0 dfdk_eval], [1 -dfdh_eval]);
30 SC = ss(G);
31 Gz = c2d(G, Ts);
32 [B,A] = tfdata(Gz, 'v');
33 a1 = A(2);
34 b1 = B(2);
35
36 n = 300;
37 plc = .86; % Polo deseado
38 Alc = poly(plc);
39 alpha = Alc(2);
40 s0 = (alpha-a1) / b1;
41 t0 = (1+alpha) / b1;
42 prec = Ts/10;
43 t = 0:prec:Ts;
44 yc = zeros(size(t));
45 uu = [];
46 yy = [];

```

```

47 vv = [];
48 x = zeros(1,length(SC.a));
49 y = zeros(n,1);
50 u = zeros(n,1);
51
52 r = ones(n,1)*0.6; % Referencia
53 vc = zeros(size(t))'; % Perturbacion
54
55 for i = 5:n
56     if i>n*2/3; r(i) = 0.2; end % Se cambia la referencia
57     y(i) = yc(end); % Muestreo de la salida
58
59     u(i) = t0*r(i)-s0*y(i); % Accion de control
60
61     uc = u(i)*ones(size(t))'; % Bloqueador actuacion
62
63     if i > n/3 % Se introduzca perturbacion
64         vc = -.2*ones(size(t))';
65     end
66
67     % Simular el sistema
68     [yc,ts,x] = lsim(SC,uc+vc,t,x(length(x),:));
69     uu = [uu; uc(2:length(uc))];
70     yy = [yy; yc(2:length(yc))];
71     vv = [vv; vc(2:length(yc))];
72 end

```

Listing 1: Código que implementa un control STR no adaptativo genérico.

2.1.1. Resultados

En la figura 1 se pueden ver las respuestas de la planta continua (G) y de la planta discretizada con el método ZOH ($G(z)$).

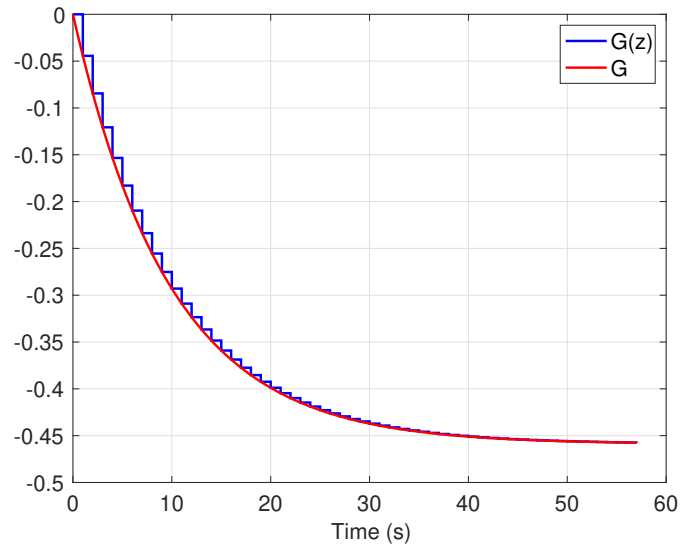


Figura 1: Respuesta al Escalón

Además, en la figura 2 se puede ver la acción de control generada a través del algoritmo STR, y la correspondiente salida. Como se puede ver, la salida se desvía de la referencia luego de introducirse la perturbación.

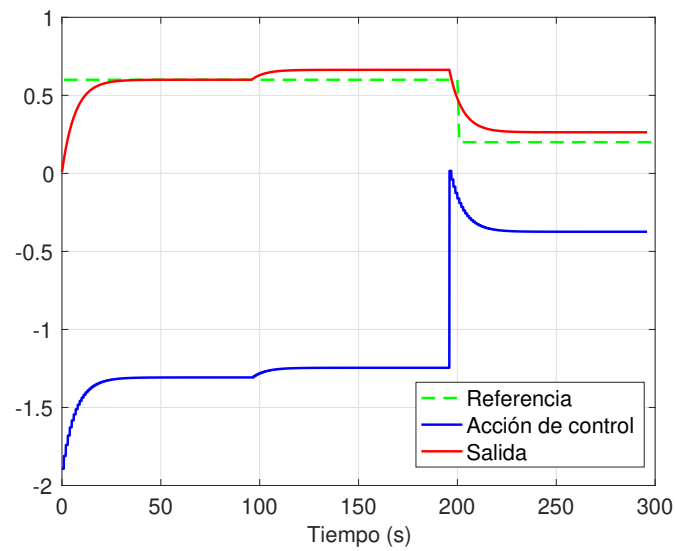


Figura 2: Referencia, u y y desde control STR no adaptativo genérico.

2.2. Separación de dinámicas

Implementamos el algoritmo de control STR con separación de dinámicas, es decir, tener una dinámica para seguir cambios de la referencia y otra dinámica para corregir perturbaciones. Resultando en una ley de control,

$$u_k = u_k^r + u_k^p \quad (11)$$

donde u^r es la acción de control para seguir la referencia, y u^p es la acción de control para corregir perturbaciones. La ponemos en la ecuación 1 y resolvemos por u_p ,

$$\begin{aligned} A(q)y_k &= B(q)(u_k^r + u_k^p + v_k) \\ B(q)u_k^p &= -B(q)v_k \\ u_k^p &= -v_k \end{aligned} \quad (12)$$

Suponemos que $A(q)y_k = B(q)u_k^r$.

La expresión de la nueva ley de control implementé como se muestra en el siguiente código,

```
1 u_ref(i) = t0*r(i) - s0*y(i); % Seguir la referencia
2 u_dist(i) = -(vc(end)); % Corregir perturbaciones
3
4 uc = (u_ref(i)+u_dist(i))*ones(size(t))';
```

Listing 2: El cambio en el código 1 que implementa separación de dinámicas.

2.2.1. Resultados

Como se puede ver en la figura 3, la nueva ley de control corrige las perturbaciones. En la realidad, se tendrían que medir las perturbaciones y luego estimar una respuesta para eliminarlas.

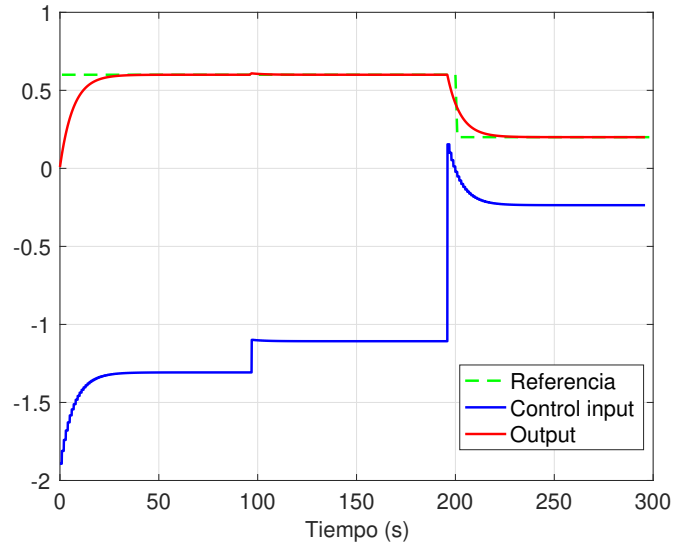


Figura 3: Referencia, u y y con separación de dinámicas.

2.3. Efecto integral

Desde *Caso 3* en [1] tenemos que la ley de control agregado de efecto integral se puede expresar como:

$$u_k = u_{k-1} + t_0 r_k - s_0 y_k - s_1 y_{k-1} \quad (13)$$

A diferencia de la ley de control 10, ahora se depende de las entradas y salidas anteriores. Esto da un efecto integral. Los parámetros se pueden calcular como se da en *Caso 3* en [1].

Lo implementé como se muestra en el siguiente código,

```

1 % Nuevos parametros
2 s0 = 1/b1*(alc1-a1+1);
3 s1 = 1/b1*(alc2+a1);
4 t0 = sum(A1c)/b1;
5 % ---
6 u(i) = u(i-1) + t0*r(i) - s0*y(i) - s1*y(i-1);

```

Listing 3: El cambio en el código 1 que implementa efecto integral.

2.3.1. Resultados

Como se puede ver en la figura 4, el agregado de efecto integral hace que la salida pueda seguir la referencia también después de que se introduzcan perturbaciones.

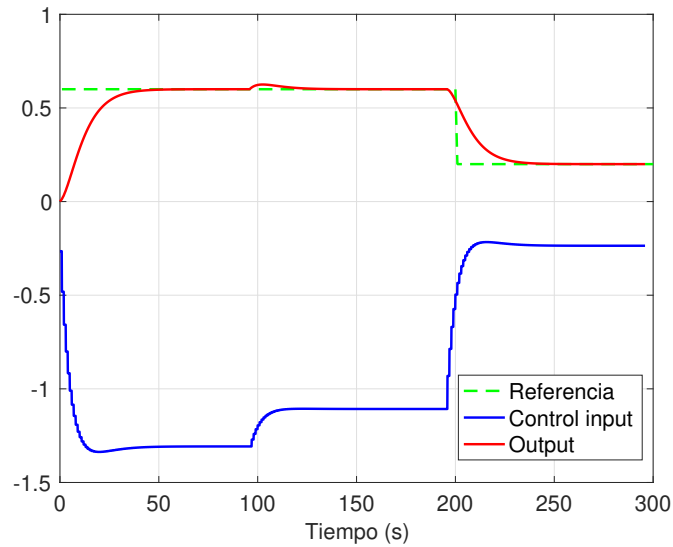


Figura 4: Referencia, u y y con efecto integral

2.4. Adaptativo indirecto

Con el método STR adaptativo indirecto, se estiman los parámetros de la planta en tiempo real. En vez de calcular los parámetros a_1 y b_1 a partir de la planta discretizada $G(z)$, ahora se calculan con el método de mínimos cuadrados recursivo, como hicimos en el TP2 [3]. Sin embargo, ahora se añade un factor de olvido para que el algoritmo dé menos peso a las entradas antiguas [4].

Lo implementé como se muestra en el siguiente código,

```

1 % Inicializar RLS
2 P = 10000*eye(2);
3 theta_hat = -0.3 * ones(2,1);
4 lambda = 0.99; % Factor de olvido
5
6 theta_hist = zeros(2, n);
7
8 for i = 5:n
9     if i>n*2/3; r(i) = 0.2; end
10    y(i) = yc(end); % Muestreo de la salida
11
12    % Metodo de minimos cuadrados
13    x_k = [-y(i-1); u(i-1)];
14    y_hat = x_k' * theta_hat;
15    K = (P * x_k) / (lambda + x_k' * P * x_k);
16    theta_hat = theta_hat + K * (y(i) - y_hat);
17    P = (P - K * x_k' * P) / lambda;
18
19    theta_hist(:, i) = theta_hat;

```

```

20
21 % Parametros estimados
22 a1 = theta_hat(1);
23 b1 = theta_hat(2);
24
25 s0 = 1/b1*(alc1-a1+1);
26 s1 = 1/b1*(alc2+a1);
27 t0 = sum(Alc)/b1;
28
29 u(i) = u(i-1)+t0*r(i)-s0*y(i)-s1*y(i-1); % Accion de
control

```

Listing 4: El cambio en el código 1 que implementa STR adaptativo indirecto.

2.4.1. Resultados

Como se puede ver en la figura 5, la entrada y la salida se comportan de manera similar a las que obtuvimos en la sección anterior 2.4. Los parámetros a_1 y b_1 se adaptan en tiempo real, como se muestra en la figura 6. Los comparé con los obtenidos a partir de la planta discretizada, y son similares hasta que se introduzcan perturbaciones.

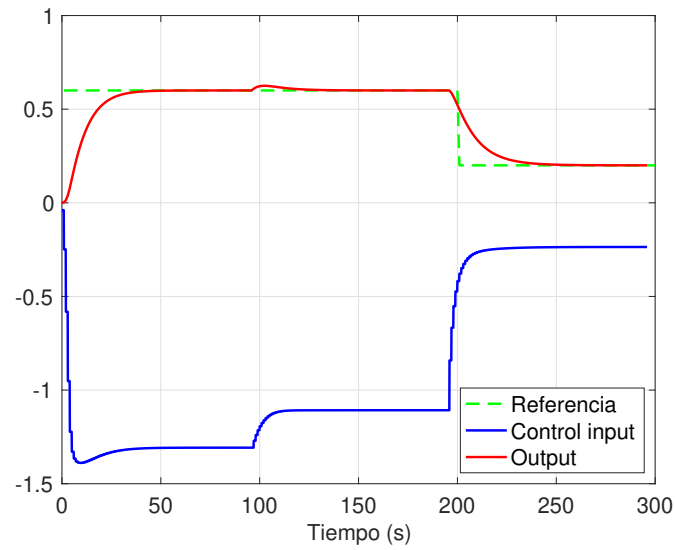


Figura 5: Referencia, u y y con STR adaptativo indirecto.

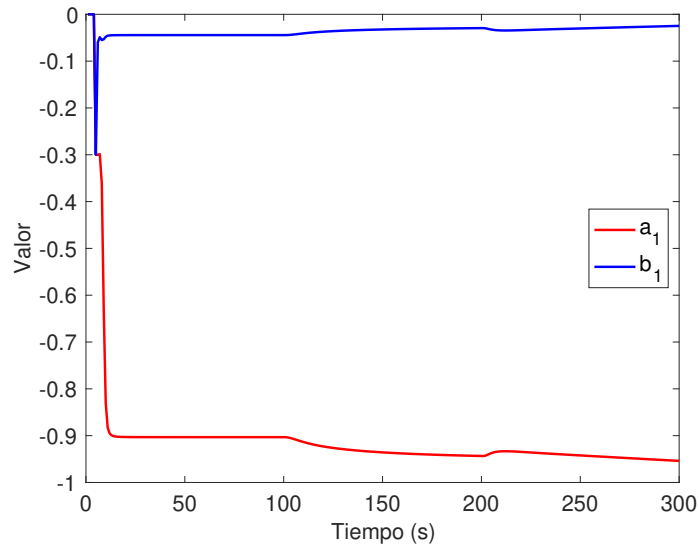


Figura 6: Los parámetros estimados con mínimos cuadrados.

2.5. Adaptativo directo

A diferencia del método adaptativo indirecto, en el método STR adaptativo directo no se estiman los parámetros del modelo de la planta, sino que se estiman directamente los parámetros de la ley de control. Es decir, el algoritmo adapta los coeficientes del controlador en tiempo real, con base en el error entre la salida del sistema y el modelo y_m deseado.

Lo implementé como se muestra en el siguiente código,

```

1 % Inicializar RLS
2 P = 10*eye(2);
3 theta_hat = [-0.3; -3];
4 lambda = 0.90; % Factor de olvido
5
6 theta_hist = zeros(2, n);
7
8 alpha = 0.86; % Polo deseado
9 y_m = zeros(n, 1); % Modelo
10
11 for i = 5:n
12     if i > 200; r(i) = 0.2; end
13
14     y_m(i) = alpha * y_m(i-1) + (1 - alpha) * r(i-1);
15
16     y(i) = yc(end); % Muestreo de la salida
17
18     % Metodo de minimos cuadrados
19     x_k = [-y(i-1); r(i-1)];
20     K = (P * x_k) / (lambda + x_k' * P * x_k);
21     theta_hat = theta_hat + K * (y(i) - y_m(i));

```

```

22     P = (P - K * x_k' * P) / lambda;
23
24     theta_hist(:, i) = theta_hat;
25
26     % Parametros de la ley de control
27     s0 = theta_hat(1);
28     t0 = theta_hat(2);
29
30     u(i) = t0 * r(i) - s0 * y(i);

```

Listing 5: El cambio en el código 1 que implementa STR adaptativo directo.

2.5.1. Resultados

Como se puede ver en la figura 7, la salida sigue la referencia, aunque presenta un overshoot. Además, los parámetros no convergen tan bien, como se muestra en la figura 8. Para mejorar este modelo, es necesario ajustar el tiempo de muestreo T_s , el polo deseado α , la matriz P , el factor de olvido λ y la estimación inicial de θ .

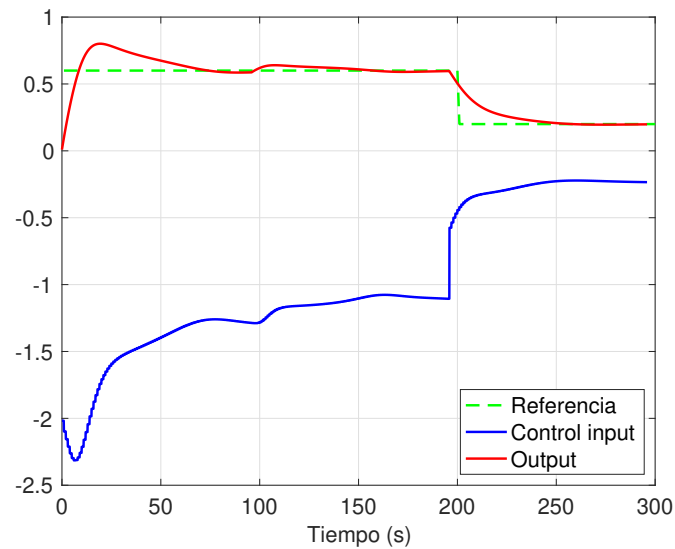


Figura 7: Referencia, u y y con STR adaptativo directo.

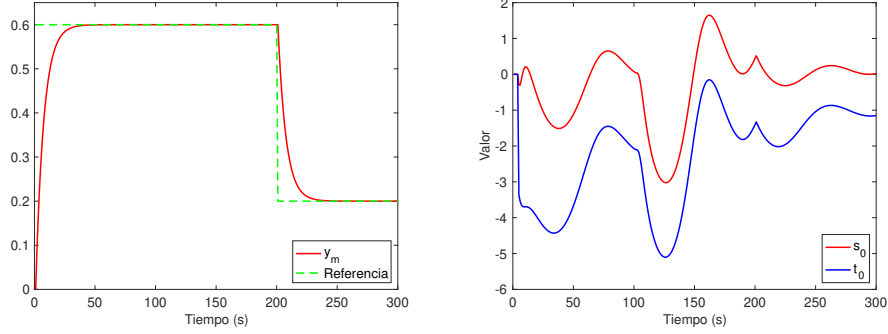


Figura 8: Modelo (izquierda) y los parámetros estimados s_0 y t_0 .

Debido a que el tanque es un sistema lento, un tiempo de muestreo de 1s puede ser demasiado pequeño. Esto también podría explicar por qué los parámetros oscilan, como se muestra en la figura 8. Por eso, aumenté el tiempo de muestreo a 5s y fijé el polo deseado en 0,70. Como se puede ver en la figura 9, esto claramente mejoró el desempeño del sistema. Además, un mayor tiempo de muestreo actúa como un filtro pasabajos, atenuando parte del ruido presente en las señales.

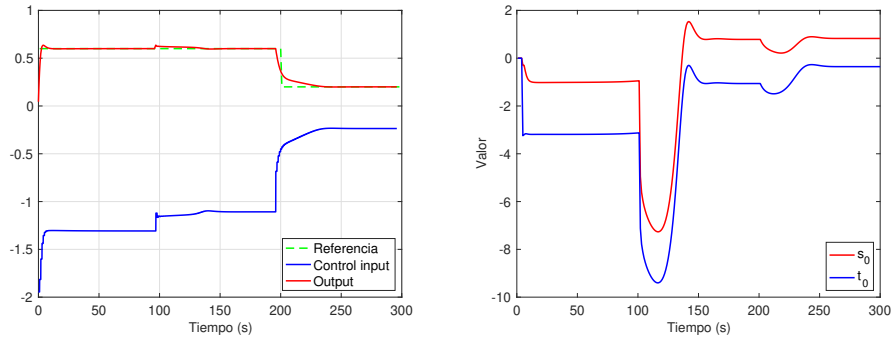


Figura 9: Referencia, u y y con T_s más grande (izquierda) y los parámetros correspondientes s_0 y t_0 .

3. Control Adaptativo Incremental

La ley de control adaptativo incremental se puede expresar como:

$$u_k = u_{k-1} + \Delta u_k \quad (14)$$

Sin embargo, esto ya fue implementado en la ecuación 13. Si definimos:

$$\Delta u_k = t_0 r_k - s_0 y_k - s_1 y_{k-1} \quad (15)$$

ambas expresiones resultan equivalentes.

En la sección 2.4, se utiliza precisamente una ley de control de esta forma, en combinación con un método STR adaptativo indirecto.

Tal vez no entendí.

Referencias

- [1] Universidad de Buenos Aires Facultad de Ingeniería. *STR Determinístico*. 2024.
- [2] Truls Carlson. *Trabajo Práctico 1*. 2024.
- [3] Truls Carlson. *Trabajo Práctico 2*. 2024.
- [4] Recursive least squares filter. https://en.wikipedia.org/wiki/Recursive_least_squares_filter. Accedido: 16-03-2025.