

ARI-HW_10

Matěj Pinkas

27. April 2024

1 Úkol

- Zvolím si konstanty a a b z mého datumu narození: 10.01.2003

$$a = 1 \qquad b = 1$$

- Dopočítám konstantu c ze zadání:

$$c = 4 \left(\frac{a + b - 2}{16} + 1 \right) = 4$$

- Ze zadání a vlastních čísel získám spojitý přenos:

$$G_s(s) = \frac{s + \frac{3}{2}c}{(s + c)(s - 1)} = \frac{s + 6}{(s + 4)(s - 1)}$$

- Diskretizuji přenos $G_s(s)$ pomocí funkce $c2d(G_s, T_s)$ pro vzorkovací frekvenci $T_s = 0,05$ s:

$$G_d(z) = \frac{0,05365z - 0,03971}{z^2 - 1,87z + 0,8607}$$

- Zvolím si regulátor PI s přenosem:

$$PI(s) = K_P + \frac{1}{s}K_I$$

$$PS(z) = K_P + \frac{1}{z - 1}K_S$$

- Pomocí funkce `rltool()` provedu PID tuning přenosu:

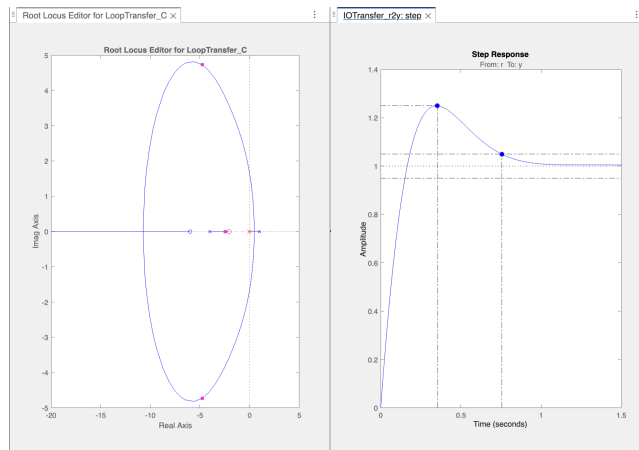


Figure 1: RLTOOL a PID tuning

při řešení se snažím splnit překmit maximálně 30% a dobu ustálení do 3 s

- Z tuningu získám konstanty:

$$K_P = 9$$

$$K_S = 2$$

- V kódu přidám regulátor a implementaci anti-windupu:

```

1  e = zeros(numel(t), 1);
2  ssum = zeros(numel(t), 1);
3
4  kp = 9;
5  ks = 2;
6
7  for k = 1:numel(t)
8      y(k) = C*x_sys(k,:)'; % output of the system
9
10     % == Your code goes here ==
11     e(k) = r(k) - y(k); % PS controller input
12     ssum(k + 1) = ssum(k) + e(k);
13
14     u(k) = kp*e(k) + ks*ssum(k+1);
15
16     if abs(u(k)) >= u_sat % anti-windup
17         u(k) = kp*e(k);
18         ssum(k+1) = 0;
19     end
20
21     % Saturation of the input
22     u(k) = max(min(u(k), u_sat), -u_sat);

```

Listing 1: Část implementace

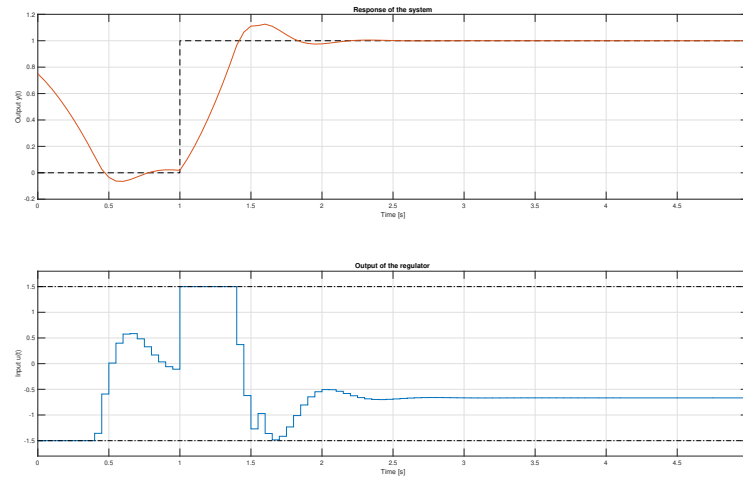


Figure 2: Odezva na jednotkový skor reference a výstup PI regulátoru