Digital Control Exercise 4

Gao Sun sungao12@student.hh.se

Problem 1 PD-control

a)

As we know:

$$\begin{cases} u(t) = P(t) + D(t) \\ P(t) = K_p e(t) \\ \frac{T_d}{N}\frac{dD(t)}{dt} + D(t) = K_p T_d \frac{de(t)}{dt} \end{cases}$$

After replacing the derivative part $\frac{d}{dt}$ with $\frac{1-q^{-1}}{h}$, we can get:

$$\frac{T_d}{N}\frac{1-q^{-1}}{h}D(k) + D(k) = K_p T_d \frac{1-q^{-1}}{h}e(k)$$

$$D(k) = \frac{NK_pT_d(1-q^{-1})}{T_d(1-q^{-1}) + Nh}e(k)$$

$$u(k) = P(k) + D(k) = K_p e(k) + \frac{NK_pT_d(1-q^{-1})}{T_d(1-q^{-1}) + Nh}e(k)$$

$$= \frac{NK_pT_d + K_pT_d + K_pNh + \left(-NK_pT_d - K_pT_d\right)q^{-1}}{T_d + Nh - T_d q^{-1}}e(k)$$

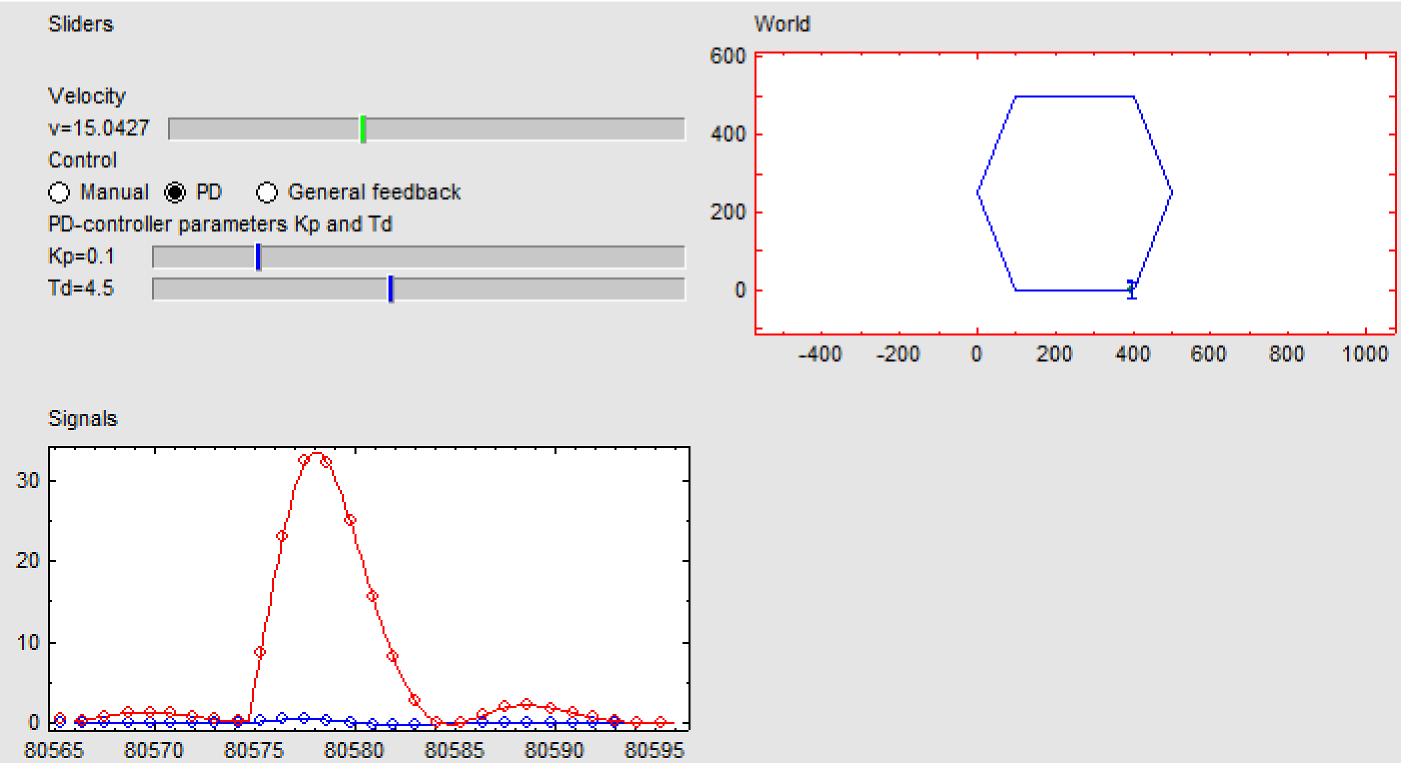
$$= \frac{\frac{NK_pT_d + K_pT_d + K_pNh}{T_d + Nh} + \left(-\frac{NK_pT_d + K_pT_d}{T_d + Nh}\right)q^{-1}}{1 + \left(-\frac{T_d}{T_d + Nh}\right)q^{-1}}e(k)$$

As $u(k) = \frac{s_0 + s_1 q^{-1}}{1 + r_1 q^{-1}}e(k)$, so the controller parameters are:

$$r_1 = -\frac{T_d}{T_d + Nh} \qquad s_0 = \frac{NK_pT_d + K_pT_d + K_pNh}{T_d + Nh} \qquad s_1 = -\frac{NK_pT_d + K_pT_d}{T_d + Nh}$$

b)

Through my trying, I found that when $K_p = 0.1$, $T_d = 4.5$, it's a good tuning of the PD-controller. The result is shown in the figure below. The robot can keep moving along the path. The signal is also quite smooth and the oscillating time is not too long.



c)

$$r_1 = -\frac{T_d}{T_d + Nh} \qquad s_0 = \frac{NK_pT_d + K_pT_d + K_pNh}{T_d + Nh} \qquad s_1 = -\frac{NK_pT_d + K_pT_d}{T_d + Nh}$$

So $p_c = -r_1 = \frac{T_d}{T_d+Nh}$, $z_c = -\frac{s_1}{s_0} = \frac{NK_pT_d+K_pT_d}{NK_pT_d+K_pT_d+K_pNh} = \frac{NT_d+T_d}{NT_d+T_d+Nh}$.

If the controller parameters K_p, T_d, N, h are all positive, then we can say that the pole and the zero of the system are both the fractions whose denominator is greater than the numerator. So the restriction is: $0 < p_c < 1, \quad 0 < z_c < 1$.

Problem 2 Discrete-time control design

a)

$A_c = AR + BS$

$= (2(1-q^{-1})^2)(1-p_c q^{-1}) + (v(q^{-1}+q^{-2}))(\frac{1}{v}K(1-z_c q^{-1}))$

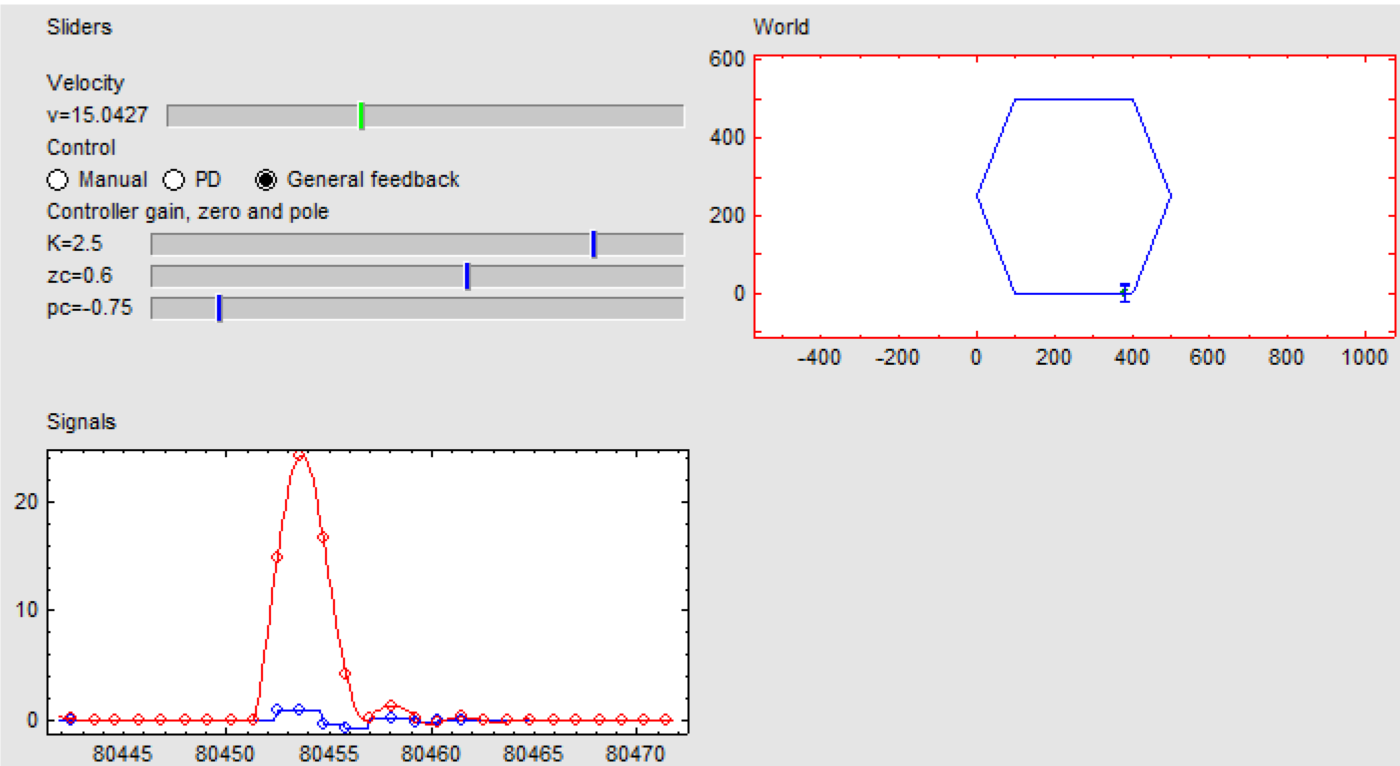
$= (-2p_c - Kz_c)q^{-3} + (4p_c + 2 - Kz_c + K)q^{-2} + (-2p_c - 4 + K)q^{-1} + 2$

In order to make the dead-beat-design, we should let A_c be a constant. So we have:

$$\begin{cases} -2p_c - Kz_c = 0 \\ 4p_c + 2 - Kz_c + K = 0 \\ -2p_c - 4 + K = 0 \end{cases}$$

So we can get the controller parameters are: $K = \frac{5}{2}, p_c = -\frac{3}{4}, z_c = \frac{3}{5}$.

b)



After tuning the 3 controller parameters, we can see that the controller's performance is improved a lot comparing with the controller we tuned in 1b). At each time point when the robot needs to make a turn, the maximum error is smaller, and the oscillating time is much shorter.

c)

As we calculated in 1c), the restriction of the PD controller is $0 < p_c < 1$, $0 < z_c < 1$. But if we use the dead-beat tuning, the p_c we get is negative. So it is impossible to achieve dead-beat design with the PD-control structure.