Laboratory 3

Adaptive position control of a DC servomotor

Description of the system. We consider the system of Figure 9.3 where

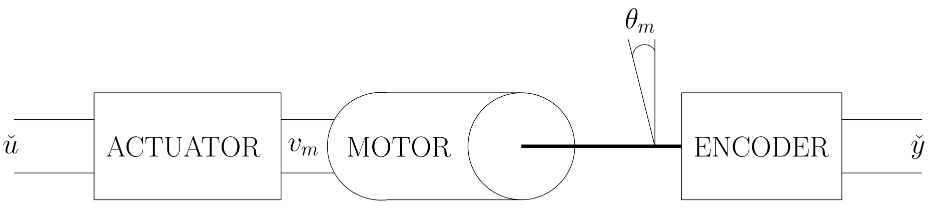

Figure 9.3: DC servomotor scheme.

- $\breve{u}$ $[V]$ is the (discrete-time) input applied to the actuator; the output of this block is a quantized signal in the interval $\pm 3\,V$;
- v_m $[V]$ is the voltage applied to the motor;
- θ_m $[rad]$ is the angular position of the rotor of the motor;
- $\breve{y}$ $[rad]$ is the (discrete-time) output of the encoder measuring θ_m; this measurement is affected by quantization errors;
- the sampling time is $T_s = 0.001$ s.

Adaptive control scheme. Let $\breve{\mathcal{M}}$ be the unknown model describing the system above: $\breve{u}(t)$ is the input, $\breve{y}(t)$ is the output and $\breve{e}(t)$ describes the errors. We consider the adaptive position control scheme of Figure 9.4 where $y(t)$ and $u(t)$ are the filtered versions of $\breve{y}(t)$ and $\breve{u}(t)$. Finally, $w(t)$ is the reference signal which is a rectangular wave taking values $0° - 100°$ of period 5 s, and duty cycle 50 %. The filtering is performed to remove the known dynamic. Indeed, let $\mathcal{F}_c(s)$ be a transfer function describing the relation between $v_m(\tilde{t})$ and $\dot{\theta}_m(\tilde{t})$, then we have

$$\dot{\theta}_m(\tilde{t}) = \mathcal{F}_c(s)v_m(\tilde{t}), \quad \tilde{t} \in \mathbb{R}.$$

Since $\dot{\theta}_m(\tilde{t}) = s\theta_m(\tilde{t})$, we have

$$\theta_m(\tilde{t}) = \frac{1}{s}\mathcal{F}_c(s)v_m(\tilde{t}), \quad \tilde{t} \in \mathbb{R}. \tag{9.10}$$

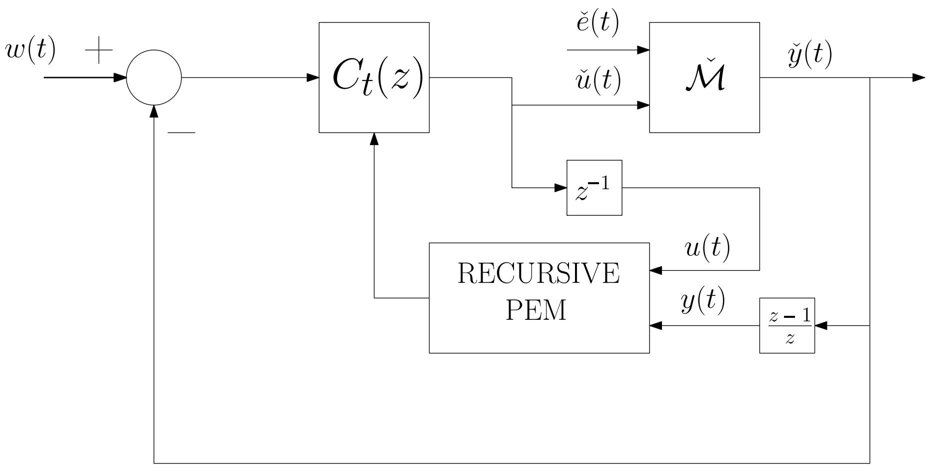

Figure 9.4: Adaptive control scheme.

Discretizing (9.10), we obtain

$$\check{y}(t) = \frac{1}{z-1}\mathcal{F}(z)\check{u}(t), \quad t \in \mathbb{Z}$$

where $\check{y}(t) = \theta_m(tT_s)$, $\check{u}(t) = v_m(tT_s)$ and $\mathcal{F}(z)$ is the discretized version (up to a constant factor) of $\mathcal{F}_c(s)$. In the above model we have a known dynamic. Accordingly, we perform the following filtering operation

$$\underbrace{\frac{z-1}{z}\check{y}(t)}_{:=y(t)} = \mathcal{F}(z)\underbrace{\frac{1}{z}\check{u}(t)}_{:=u(t)}$$

and thus

$$y(t) = \mathcal{F}(z)u(t).$$

Open the corresponding Simulink file `scheme.slx`.

Recursive PEM identification. We estimate $\mathcal{F}(z)$ recursively by using the ARX model structure

$$\mathcal{M} \,:\, A(z)\boldsymbol{y}(t) = B(z)\boldsymbol{u}(t) + \boldsymbol{e}(t)$$

with $n_A = 1$, $n_B = 2$ and $n_k = 0$. Accordingly, the estimate will be

$$\hat{\mathcal{F}}_t(z) = \frac{\hat{B}_t(z)}{\hat{A}_t(z)} = \frac{\hat{b}_{0,t}z + \hat{b}_{1,t}}{z + \hat{a}_{1,t}}$$ [1]

[1]Here, $\hat{A}_t(z)$ and $B_t(z)$ are understood as polynomials in z. This will be useful later on.

and $\hat{\theta}_t := [\,\hat{a}_{1,t}\, \hat{b}_{0,t}\, \hat{b}_{1,t}\,]^T$ is the RPEM estimate using $\mathcal{M}$. Open the Matlab function `recursiveid.m` (used in the block "Recursive identification" in the Simulink scheme). The input parameters are $\hat{\theta}_{t-1}$, S_{t-1}, $y(t)$, $y(t-1)$, $u(t)$ and $u(t-1)$; the output ones are $\hat{\theta}_t$, S_t, $y(t)$ and $u(t)$. Implement in this function the recursive PEM method using $\lambda = 0.999$. The initial conditions of the algorithm are stored in the block "Memory buffer register": $\hat{\theta}_0 = [\,-0.2\; 0.02\; 0.001\,]^T$, $S_0 = 0.01 I_3$, $y(0) = 0$ and $u(0) = 0$.

Compensator design. We consider the following compensator with integral action

$$C_t(z) = \frac{N_t(z)}{(z-1)D_t(z)}$$

where the polynomials $N_t(z)$ and $D_t(z)$ need to be designed. At time t, we consider the feedback control problem of Figure 9.5 . The closed loop

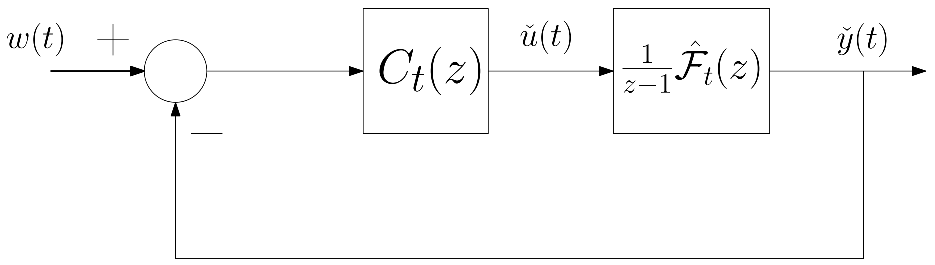

Figure 9.5: Feedback control scheme.

transfer function is

$$T_t(z) = \frac{(z-1)^{-1}C_t(z)\hat{\mathcal{F}}_t}{1+(z-1)^{-1}C_t(z)\hat{\mathcal{F}}_t} = \frac{\hat{B}_t(z)N_t(z)}{(z-1)^2\hat{A}_t(z)D_t(z) + \hat{B}_t(z)N_t(z)}.$$

Note that, $\hat{\mathcal{F}}_t(z) = \hat{B}_t(z)/\hat{A}_t(z)$ is known. We require that the denominator of $T_t(z)$ has the following structure

$$D_\star(z) = z^2(z-p)(z-\bar{p})$$

where $p = \exp((-\xi\omega_n + j\omega_n\sqrt{1-\xi^2})T_s)$ with

$$\xi = \frac{|\ln O|}{\sqrt{\pi^2 + (\ln O)^2}}, \quad \omega_n = \frac{1.8}{t_R}.$$

In plain words, $T_t(z)$ is well approximated by a second order model and $p, \bar{p}$ are the dominant poles. The quantities t_R and O are the rise time and the overshoot. Recall that:

- t_R is the time that the step response of the system takes from 0 to reach 90% of the steady state value;
- $O = 1 - \tilde{O}$ where $\tilde{O}$ is the ratio between the maximum value of the step response and the steady state value.

See also Figure 8.2. Next, we take $t_R = 0.1$ s and $O = 0.3$. Then, N_t and D_t are given by solving the following Diophantine equation (see `diophantine.m`):

$$(z-1)^2 \hat{A}_t(z) D_t(z) + \hat{B}_t(z) N_t(z) = D_\star(z).$$

Open the Matlab function `Cparameters.m` (used in the block "Controlller" in the Simulink scheme). The input parameter is $\hat{\theta}_t$ and the output ones are the numerator $N_t(z)$ and the denominator $(z-1)D_t(z)$ of $C_t(z)$. Implement this function. Notice that Matlab represents polynomials with numerical vectors containing the polynomial coefficients ordered by descending power. Finally, the product of two polynomials is obtained by using the Matlab function `conv`.

Simulation. Run the Simulink scheme.

Question 1: Is the control law appropriate? What conclusions can you draw about the RPEM estimate? Motivate the answer.

Changing λ. Run the Simulink scheme with $\lambda = 1$ and $\lambda = 0.95$.

Question 2: What differences can you recognize with respect to before? Motivate the answer.