Laboratory 1

Nonparametric system identification of a single-link manipulator

System and data description. We consider the single-link manipulator depicted in Figure 1: $\tau(t)$ is the torque generated by the actuator and $q_m(t)$ is the resulting angular position of the rotor of the actuator. We have two data sets collected from this manipulator. More precisely,

- First data set:
  - inputs: $u1 = [q_m(1) \dots q_m(N)]^T$, $u1d = [\dot{q}_m(1) \dots \dot{q}_m(N)]^T$;
  - output: $y1 = [\tau(1) \dots \tau(N)]^T$.
- Second data set:
  - inputs: $u2 = [q_m(1) \dots q_m(N)]^T$, $u2d = [\dot{q}_m(1) \dots \dot{q}_m(N)]^T$;
  - output: $y2 = [\tau(1) \dots \tau(N)]^T$.

Here, $N = 201$ and the sampling time is $T_s = 0.1\,s$. These data are stored in the file `manipulator1.mat`.

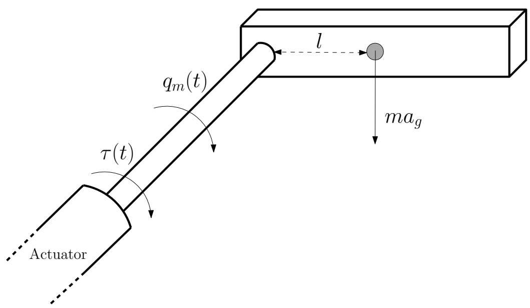

Figure 1: Schematic diagram of a single-link manipulator.

Data plotting. Plot the inputs and the output of the first and of the second data set.

Computation of the acceleration. Each data set provides only the angular position and velocity of the rotor of the actuator, however, in what follows we will need also the acceleration. The latter can be estimated using the Matlab function `ud_dot=derivative(ud,Ts)` where `ud` denotes the velocity vector, `ud_dot` its derivative vector, and `Ts` the sampling time. This function computes the derivative using the backward Euler method. In addition, to reduce the effect of the noise, the latter is filtered with a first-order low pass filter.

Inverse dynamic. We want to estimate the inverse dynamic of the manipulator, that is to find a mathematical model with input q_m and with output τ, from the first data set. From physics, we know that the nonlinear dynamic relation between q_m and τ is given by

$$\tau(t) = J\ddot{q}_m(t) - 2lma_g \sin(q_m(t)) + \tau_f(t) \tag{1}$$

where J is the moment of inertia of the link; $2l$ and m are the length and the mass, respectively, of the link; a_g is the gravitational acceleration; τ_f is the friction (composed by stiction, viscous friction and stribeck effect) acting on the manipulator. Adopting the nonparametric Bayesian viewpoint, and in view of (1), one single measurement is modeled as

$$\boldsymbol{y}_1(t) = \boldsymbol{h}(x_1(t)) + \boldsymbol{w}(t), \quad t = 1 \dots N$$

where

$$x_1(t) = \begin{bmatrix} q_m(t) \\ \dot{q}_m(t) \\ \ddot{q}_m(t) \end{bmatrix} \in \mathbb{R}^3$$

is called input location at time t and the subscript 1 stresses the fact that $q_m(t)$, $\dot{q}_m(t)$ and $\ddot{q}_m(t)$ are taken from the first data set; $\boldsymbol{h} = \{\boldsymbol{h}(u),\ u \in \mathbb{R}^3\}$ is a stochastic Gaussian process taking values in $\mathbb{R}$ which will be specified later; $\boldsymbol{y}_1(t)$ models the measured torque at time t in the first dataset. Moreover, we assume that $\boldsymbol{w}(t) \sim \mathcal{N}(0, \sigma^2)$ with $\sigma^2 = 4.2$, $\mathbb{E}[\boldsymbol{w}(t)\boldsymbol{w}(s)] = 0$ for any $t \neq s$. Stacking the N measurements we obtain

$$\boldsymbol{y}_1 = \boldsymbol{g}(x_1) + \boldsymbol{w}$$

where

$$\boldsymbol{g}(x_1) = \begin{bmatrix} \boldsymbol{h}(x_1(1)) \\ \vdots \\ \boldsymbol{h}(x_1(N)) \end{bmatrix}, \ x_1 = \begin{bmatrix} x_1^T(1) \\ \vdots \\ x_1^T(N) \end{bmatrix}, \ \boldsymbol{y}_1 = \begin{bmatrix} \boldsymbol{y}_1(1) \\ \vdots \\ \boldsymbol{y}_1(N) \end{bmatrix}, \ \boldsymbol{w} = \begin{bmatrix} \boldsymbol{w}(1) \\ \vdots \\ \boldsymbol{w}(N) \end{bmatrix}$$

and $\boldsymbol{w} \sim \mathcal{N}(0, \sigma^2 I_N)$. $\boldsymbol{g} = \{\boldsymbol{g}(x),\ x \in \mathbb{R}^{N\times 3}\}$ is a Gaussian process taking values in $\mathbb{R}^N$ with zero mean and such that

$$\mathbb{E}[\boldsymbol{g}(x_1)\boldsymbol{g}(x_2)^T] = \mathbf{K}(x_1, x_2) = \lambda[K_C(x_1(t), x_2(s))]_{t,s}$$

with $\lambda > 0$. Moreover, $\boldsymbol{w}$ is independent from $\boldsymbol{g}$. In what follows, we consider the Cauchy kernel defined

$$K_C(x_1, x_2) = \frac{1}{1 + \beta^{-1}\|x_1(t) - x_2(s)\|^2} \tag{2}$$

with $x_1(t), x_2(t) \in \mathbb{R}^3$ and $t, s = 1 \dots N$. Note that, the latter is the generalized version of the one in Table in the book (indeed, in this case $\boldsymbol{h}$ is indexed in a 3-dimensional space). Then, the MAP estimate of $\boldsymbol{g}$ is:

$$\hat{g}_{ML}(\cdot) = \mathbf{K}(\cdot, x_1)(\mathbf{K}(x_1, x_1) + \sigma^2 I_N)^{-1} y_1$$

and the resulting model is

$$\boldsymbol{y}(t) = \hat{h}_{MAP}(x(t)) + \boldsymbol{w}(t). \tag{3}$$

Validation step. We use the second data set to check whether model (3) is good or not. Let $y_2 = [\, y_2(1) \ \dots \ y_2(N)\,]^T$ be the vector containing the values of the applied torque in the second data set. The prediction of y_2 from model (3) is:

$$\begin{aligned} \hat{y}_2 &= \hat{g}_{MAP}(x_2) \\ &= \mathbf{K}(x_2, x_1)(\mathbf{K}(x_1, x_1) + \sigma^2 I_N)^{-1} y_1 \end{aligned} \tag{4}$$

where $x_2(t) = [\, q_m(t) \ \dot{q}_m(t) \ \ddot{q}_m(t)\,]^T$ is the input location at time t with angular position, velocity and acceleration of the second data set and $x_2 = [\, x_2(1) \dots x_2(N)\,]^T$. Compute the prediction (4) using the Cauchy kernel in (2) with $\lambda = 10$ and $\beta = 100$. The corresponding kernel matrices $\mathbf{K}(x_1, x_1)$ and $\mathbf{K}(x_2, x_1)$ are computed by calling the function

```
lambda*Cauchy_kernel(x,xt,beta);
```

the input arguments correspond to $\lambda, x, \tilde{x}, \beta$, respectively, and the output is $\mathbf{K}(x, \tilde{x})$.

Question 1: According to the results you found, does the estimated model describe well the manipulator system? Motivate the answer.

Validation with other values of λ and β. Do the same as in the previous point but using now the following values for λ and β:

	λ	β
Case 2	0.1	100
Case 3	10	1

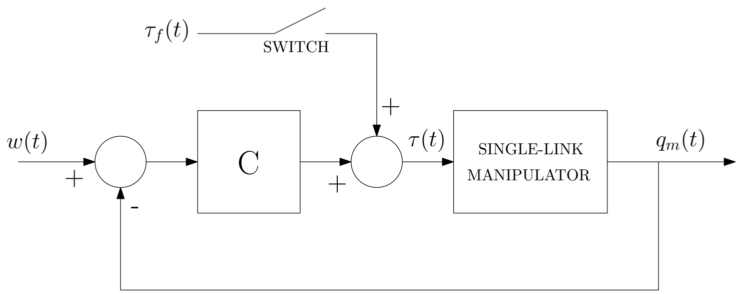

Figure 2: Position control scheme.

Question 2: According to the results you found, which is the role of λ and β?

Feedforward control. Consider the position control scheme for the previous single-link manipulator which is depicted in Figure 2. Here, $\tau(t)$ is the applied torque, $q_m(t)$ is the angular position, C is a proportional controller and $w(t)$ is the reference signal which is defined as

$$w(t) = \begin{cases} 0, & 1 \leq t \leq 10 \\ 10, & 11 \leq t \leq N. \end{cases} \tag{5}$$

$\tau_f(t)$ is the feedforward action which can be activated or not through a switch. Recalling that the sampling time is $T_s = 0.1\,s$, then it means that $w(t)$ is a step from 0 to 10 at time 1 s. Open the corresponding Simulink scheme `scheme.slx`. Notice that, the feedforward action can be designed according to the MAP estimator found before:

$$\hat{y}_f = \hat{g}_{MAP}(x_f) = \mathbf{K}(x_f, x_1)(\mathbf{K}(x_1, x_1) + \sigma^2 I_N)^{-1} y_1 \tag{6}$$

where $x_f(t) = [\, w(t)\ \dot{w}(t)\ \ddot{w}(t)\,]^T$ is the input location at time t with angular position, velocity and acceleration of the reference signal and $x_f = [\, x_f(1) \ldots x_f(N)\,]^T$. Note that, the values of the derivatives of $w(t)$ can be computed through the Matlab function `derivative`. Compute the feedforward action (6) corresponding to (5) and using the Cauchy kernel in (2) with $\lambda = 10$ and $\beta = 100$ (call the vector $\hat{y}_f$ as `yf` in Matlab). Then, run the Simulink scheme with and without feedforward action. Repeat the previous steps with: $\lambda = 0.1$, $\beta = 100$ and $\lambda = 10$, $\beta = 1$.

Question 3: According to the results you found, does the feedforward action improve the control action? Motivate the answer.