

Homework 4: Learning Dynamical Systems in Real Time

Lecturer: Marzia Cescon

Table of Contents

1. Linear regression models.....	1
1.1 Least-squares estimation.....	2
2. Recursive computations.....	2
2.1 Recursive least-squares (RLS) estimation.....	2
2.2 Time-varying parameters.....	3
References.....	7

In Homework 1 we have reviewed two common classes of mathematical models used to describe the behavior of a dynamic system from first-principles. This represented an analytical approach to the construction of a model. In many cases the processes are so complex that it is simply not possible to obtain reasonable models using only physical insights. An alternative way, then, to obtain mathematical representations of systems is offered by system identification. In this approach, experiments are performed on the system by exciting its dynamics in order to collect data. A model, then, is fitted to the generated data by assigning suitable numerical values to its parameters. Often, it happens that a model based on first-principles contains a number of unknown parameters. Identification methods can be applied to estimate the unknown parameters.

In this homework we will study a method to update the model's parameter estimates obtained with system **identification in real-time**. For more details on the methods, check [\[1\]](#), [\[2\]](#).

1. Linear regression models

Let us consider the model structure M given by the difference equation:

$$y(t) + a y(t-1) = b u(t-1) + \varepsilon(t)$$

where $t \in \mathbb{R}$ indicates time, $y(t)$ is the observed output variable, $u(t)$ is the input variable and $\varepsilon(t)$

is a disturbance term. The vector of parameters is defined as $\theta = [a \ b]^\top$. For a given dataset

$\{u(1) y(1), u(2) y(2), \dots, u(N) y(N)\}$, the term $\{\varepsilon(t)\}$ can be regarded as a function of the parameter vector. To see this, let us rewrite the difference equation as:

$$\varepsilon(t) = y(t) - (-y(t-1) u(t-1)) \cdot \theta = y(t) - \varphi^\top(t) \cdot \theta$$

where the vector $\varphi(\cdot)$ was introduced to contain the so-called *regression variables* or regressors. The parameter vector θ can be estimated in such a way that it minimizes the *sum of squared* equation errors. That is,

introducing the *least-squares loss function* $V(\theta, t) = \frac{1}{2} \sum_{i=1}^N \varepsilon^2(t) \implies \hat{\theta} = \arg \min_{\theta} V(\theta, t)$. Since the measured

variable $y(t)$ is linear in the parameters, and the least-squares criterion is quadratic, the problem admits an analytic solution. Introduce the notations:

$$Y(t) = (y(1) \ y(2) \ \dots \ y(N))^T$$

$$E(t) = (\varepsilon(1) \ \varepsilon(2) \ \dots \ \varepsilon(N))^T$$

$$\Phi(t) = \begin{bmatrix} \varphi^T(1) \\ \varphi^T(2) \\ \vdots \\ \varphi^T(N) \end{bmatrix}$$

the least-squares loss function can be written as: $V(\theta, t) = \frac{1}{2} \sum_{i=1}^N \varepsilon^2(t) = \frac{1}{2} E^T E$, where $E = Y - \Phi^T \theta$.

1.1 Least-squares estimation

The parameter vector $\hat{\theta}$ that minimizes the loss function $V(\theta, t)$ is given by the normal equation:

$$\Phi^T \Phi \hat{\theta} = \Phi^T Y.$$

If the matrix $\Phi^T \Phi$ is nonsingular, the minimum is unique and it is given by:

$$\hat{\theta} = (\Phi^T \Phi)^{-1} \Phi^T Y.$$

Remark. The condition that the matrix $\Phi^T \Phi$ is invertible, is called an excitation condition.

2. Recursive computations

In adaptive controllers the observations are obtained sequentially in real time. It is then desirable to update the parameter estimates obtained with least-squares identification recursively i.e., on-line. Computations can be arranged in such a way that if there is an estimate $\hat{\theta}(t-1)$ based on data up to a certain time $t-1$, then the new estimate $\hat{\theta}(t)$ at time t can be derived from that at time $t-1$. In contrast, off-line or batch methods use all the recorded data at once to find the parameter estimates. Recursive computations have modest primary memory requirements, as not all data need to be stored. When adopted either explicitly or implicitly in adaptive control systems, this approach ensures that a current estimated model of the process is always available.

Let us now rewrite the solution to the least-squares problem in recursive form. Let $\hat{\theta}(t-1)$ denote the least-squares estimate based on measurements up to time $t-1$.

2.1 Recursive least-squares (RLS) estimation

Assume the matrix $\Phi^T(t) \Phi(t)$ is nonsingular for all $t \geq t_0$. Given $\hat{\theta}(t_0)$ and $P(t_0) = (\Phi^T(t_0) \Phi(t_0))^{-1}$, the least-squares estimates satisfies the recursive equations:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)\varepsilon(t)$$

$$\varepsilon(t) = y(t) - \varphi^T(t)\hat{\theta}(t-1)$$

$$K(t) = P(t)\varphi(t) = P(t-1)\varphi(t)(I + \varphi^T(t)P(t-1)\varphi(t))^{-1}$$

$$P(t) = (I - K(t)\varphi^T(t))P(t-1)$$

where $y(t)$ is the output, $\varepsilon(t)$ a prediction error, $K(t)$ a weighting factor. It can be seen that the estimate $\hat{\theta}(t)$ is obtained by adding a correction to the previous estimate $\hat{\theta}(t-1)$, which is proportional to $y(t) - \varphi^T(t)\hat{\theta}(t-1)$, where the second term can be interpreted as the value of $y(t)$ predicted by the regression model $y(t) = \varphi^T(t) \cdot \theta$. The correction term is thus proportional to the difference between the measured value of $y(t)$ and the prediction of $y(t)$ based on the previous parameter estimate. The component of the vector $K(t)$ are weighting factors that tell how the correction and the previous estimate should be combined.

2.2 Time-varying parameters

In the case of slowly time-varying parameters, a time-varying weighting of the data can be introduced in the least-squares loss function, yielding: $V(\theta, t) = \frac{1}{2} \sum_{i=1}^N \lambda^{N-i} \varepsilon^2(t)$, where λ is called the *forgetting factor* and it is a parameter such that $0 < \lambda \leq 1$. This means that the most recent data is given unit weight. Using a forgetting factor, the RLS equations become:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)\varepsilon(t)$$

$$\varepsilon(t) = y(t) - \varphi^T(t)\hat{\theta}(t-1)$$

$$K(t) = P(t)\varphi(t) = P(t-1)\varphi(t)(\lambda I + \varphi^T(t)P(t-1)\varphi(t))^{-1}$$

$$P(t) = (I - K(t)\varphi^T(t))P(t-1)/\lambda$$

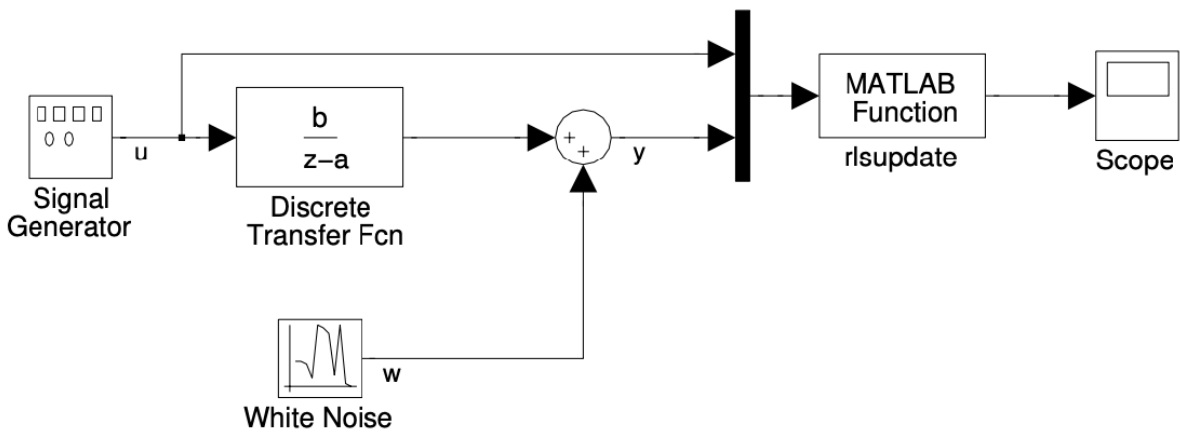
Consider collecting input-output data generated by the discrete-time system

$$\Sigma_{true} : y(t) = ay(t-1) + bu(t-1) + w(t) + cw(t-1) \quad (1)$$

where t is the discrete-time index, the parameters are $a = 0.9$, $b = 0.1$, $c = 0$ and $w(t)$ is a white noise process with mean $\mu = 0$ and variance $\sigma^2 = 1$. We will write a recursive estimation program exploiting the Simulink model below to fit the model

$$\Sigma : y(t) = \hat{a}y(t-1) + \hat{b}u(t-1) \quad (2)$$

to the generated data. This will entail the identification of $\hat{\theta} = [\hat{a} \quad \hat{b}]^T$ with regressors $\Phi = [y(t) \quad u(t)]$.



Task 1. Open the file `rlsmodel.mdl`

```
open('rlsmodel.mdl')
```

What Simulink block was chosen to implement the recursive least-squares?

```
Task1 = "Select one of the following"
```

Task 2. Now open the file `rlsupdate.m`

```
% Enter Code Here.
```

What variables have been defined as persistent? And why? Hint: Try 'help persistent'.

```
% Enter Code Here.
```

Task 3. Implement the recursive equations of the least squares in the file `rlsupdate.m`. Choose initial conditions $\hat{\theta}(0) = \theta_0 = [0 \ 0]^T$ and $P(0) = P_0 = I$.

```
open('rlsupdate.m')
```

Task 4. Persistency of excitation condition.

(a) Experiment with the choice of the input signal. Start with a square wave of unit amplitude and a period of 100 samples. Repeat the experiment with various amplitudes and periods. Provide plots of the various experiments that you carried out. Function `HW4_plots1` is used for plotting.

- Trial 1:

```

clear rlsupdate % clears persistent variables from previous runs
% White Noise Parameters:
h = 0.60;      % Sample time [s]
sigma = 3;     % Variance

% Transfer Fcn Parameters:
a = 0.9;      % Pole
b = 0.1;      % Numerator

% Input Signal (Square Wave) Parameters:
uA = 1;       % Amplitude
uP = 100;     % Period [s]

sim('rlsmodel.mdl')
HW4_plots1(a, b, data, output);
subplot(2,1,1);
title({'Inputs and Outputs', ['Square Wave Input, uA = ', num2str(uA), ', uP = ', num2str(uP)]})

```

- Trial 2:

```

clear rlsupdate

% Input Signal (Square Wave) Parameters:
uA = 1;       % Amplitude
uP = 100;     % Period [s]

sim('rlsmodel.mdl')
HW4_plots1(a, b, data, output);
subplot(2,1,1);
title({'Inputs and Outputs', ['Square Wave Input, uA = ', num2str(uA), ', uP = ', num2str(uP)]})

```

- Trial 3:

```

clear rlsupdate

% Input Signal (Square Wave) Parameters:
uA = 1;       % Amplitude
uP = 100;     % Period [s]

sim('rlsmodel.mdl')
HW4_plots1(a, b, data, output);
subplot(2,1,1);
title({'Inputs and Outputs', ['Square Wave Input, uA = ', num2str(uA), ', uP = ', num2str(uP)]})

```

(b) Try with a PRBS input signal. Provide plots. Hint: Check this [link](#) for suggestions.

```
clear rlsupdate
% Enter Code
```

(c) Use a Unit Pulse input signal at $t = 50$. Provide plots. Hint: Use the [Discrete Pulse Generator Block](#) or a user defined block.

```
clear rlsupdate
% Enter Code
```

(d) How does the choice of the input signal affect the result? Comment on the convergence properties of the estimated parameters with respect to the input signal.

Task 5. Initial conditions. Try different initial conditions. Do the initial conditions affect the result? Explain and provide plots of the various experiments that you carried out. Hint: Change the initial conditions in `rlsupdate.m`, then run the `rlsmodel.mdl` using default square wave input signal.

- Trial 1:

```
% Enter Code
```

- Trial 2:

```
% Enter Code
```

- Trial 3:

```
% Enter Code
```

Task 6. Model structure. In this task, $c = -0.5$. The equation error $w(t) + cw(t-1)$ gives a colored noise. Show that fitting of the model in (2) gives biased estimates (it is okay to show only plots). Hint: Modify `rlsmodel.mdl`, then run using default square wave input signal.

```
% Enter Code
```

Task 7. Bonus. Write a program for extended least squares estimation and show that this gives unbiased estimates of \hat{a} , \hat{b} and \hat{c} .

```
% Enter Code
```

Task 8. Influence of forgetting factor. Assume a is time varying and introduce a forgetting factor λ for its estimation. Test different values of λ . How does λ affect the result? Provide plots of the various experiments that you carried out.

- Trial 1:

% Enter Code

- Trial 2:

% Enter Code

- Trial 3:

% Enter Code

References

- [1] Ljung, L., & Söderström, T. (1983). *Theory and practice of recursive identification*. MIT press.
- [2] Åström, K. J., and Wittenmark, B. (1995). *Adaptive Control*. Addison Wesley.