M.E. 530.647 Lab 3

Revision 01

Louis L. Whitcomb*
Department of Mechanical Engineering
G.W.C. Whiting School of Engineering
Johns Hopkins University
https://dscl.lcsr.jhu.edu/courses/530-647-adaptive-systems-fall-2017

1. Simulate the performance of a stable adaptive parameter estimator for the *scalar* linear dynamical system

$$\dot{x}(t) = ax(t) + bu(t); \quad a < 0 \tag{1}$$

with constant scalar parameters $a < 0$ and b. Use the adaptive parameter estimator given by

$$\begin{aligned} \dot{\hat{x}} &= a_m \Delta x(t) + \hat{a}x(t) + \hat{b}u(t) \\ \dot{\hat{a}} &= -\gamma_1 \Delta x(t) x(t) \\ \dot{\hat{b}} &= -\gamma_2 \Delta x(t) u(t) \end{aligned} \tag{2}$$

where $\Delta\theta = \hat{\theta} - \theta$. The constant scalars $a_m < 0, \gamma_1 > 0$, and $\gamma_2 > 0$ are design parameters of the adaptive estimator.

   (a) Write a function `SCALDYN1(t,x,u)` which accepts the scalar real arguments `t`, `x`, and `u`, and returns the scalar value given by equation (1) when $a = -2$ and $b = 4$.

   Check your function's results for various arguments with hand calculations (show your work) to verify that your function produces correct values.

   (b) Write a function $\texttt{EST1}(\texttt{t}, \texttt{x}, \hat{\texttt{x}}, \hat{\texttt{a}}, \hat{\texttt{b}}, \texttt{u}, \texttt{a}_\texttt{m}, \gamma_1, \gamma_2)$ which accepts NINE scalar real arguments (NOT a 9×1 vector) and returns the 3×1 vector given by (2)

$$\begin{bmatrix} \dot{\hat{x}} \\ \dot{\hat{a}} \\ \dot{\hat{b}} \end{bmatrix} = \begin{bmatrix} a_m \Delta x(t) + \hat{a}x(t) + \hat{b}u(t) \\ -\gamma_1 \Delta x(t) x(t) \\ -\gamma_2 \Delta x(t) u(t) \end{bmatrix} \tag{3}$$

   Check your function's results for various arguments with hand calculations (show your work) to verify that your function produces correct values.

   (c) Write a function $\texttt{SYS1}(\texttt{t}, \texttt{p})$ which accepts the scalar real argument `t` and the 4×1 vector `p` where

$$p = \begin{bmatrix} x \\ \hat{x} \\ \hat{a} \\ \hat{b} \end{bmatrix} \tag{4}$$

and uses the the functions `SCALDYN()` and `EST1()`, to compute the return value

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \\ \dot{\hat{a}} \\ \dot{\hat{b}} \end{bmatrix} \tag{5}$$

for the conditions

$$\begin{aligned} u(t) &= 4sin(t) \\ a_m &= -1 \\ \gamma_1 &= 1 \\ \gamma_2 &= 1. \end{aligned} \tag{6}$$

Your function should FIRST compute $u(t)$ as specified, and SECOND use `SCALDYN()` and `EST1()` to compute the return values.

Check your function's results for various arguments with hand calculations (show your work) to verify that your function produces correct values.

(d) Use `SYS1`(), `ODE45()`, and `PLOT` simulate, print, and annotate the solutions to this system for TWO different initial conditions

$$\begin{bmatrix} x(0) \\ \hat{x}(0) \\ \hat{a}(0) \\ \hat{b}(0) \end{bmatrix}. \tag{7}$$

i. Choose your initial conditions such that $x(0) \neq \hat{x}(0)$, $\hat{a}(0) \neq a$, and $\hat{b}(0) \neq b$.

ii. If necessary, use *odeset* to increase the numerical accuracy of the simulation. Run the simulation for enough time for the system to clearly reach steady-state, something line the following:

```
p0 = [10,-10,-2,5]'
opt = odeset('RelTol',1e-8);
[t,y] = ode45('sys1',[0 30],p0,opt);
```

iii. Plot the following signals. Use the "legend" command to label the individual signals. Or label by hand. The plots can be in separate plots or sub-plots.

A. $x(t)$, $\hat{x}(t)$, and $(\hat{x}(t) - x(t))$ versus time

B. $\hat{a}(t)$, and $(\hat{a}(t) - a)$ versus time.

C. $\hat{b}(t)$, and $(\hat{b}(t) - b)$ versus time.

iv. Write a Lyapunov function for this system called `LYAP1()` which can accept the state vector output of ODE45. plot the value of a Lyapunov function for this system as a function of time.

v. Comment on (a) the convergence (or lack thereof) of the parameter estimates and (b) the scalar valued Lyapunov function $V(\cdot)$.

(e) Experiment with different gains, estimator models, plants, and initial conditions. Nothing to hand in for this one.

2. Simulate the performance of a stable model reference adaptive controller for the scalar *unstable* linear dynamical system

$$\dot{x}(t) = ax(t) + bu(t); \quad a > 0 \tag{8}$$

with constant scalar parameters $a = 1$ and $b = 1$. Assume that a and $|b|$ are unknown and but $sign(b)$ is known. Use the model reference adaptive controller given by

$$\begin{aligned} \dot{x}_m &= a_m x_m(t) + b_m r(t) \\ \dot{\theta} &= -\gamma_1 sign(b) \Delta x(t) x(t) \\ \dot{k} &= -\gamma_2 sign(b) \Delta x(t) r(t) \\ u &= \theta x + kr \end{aligned} \tag{9}$$

where

$$\begin{aligned} \Delta x &= x - x_m \\ \Delta \theta &= \theta - \theta^* \\ \Delta k &= k - k^* \end{aligned} \tag{10}$$

The constant scalars $a_m = -2, b_m = 2, \gamma_1 = 1$, and $\gamma_2 = 1$ are design parameters of the adaptive estimator.

(a) Compute numerical values for θ^* and k^*.

(b) Write a function `SCALDYN2(t,x,u)` which accepts the scalar real arguments `t`, `x`, and `u`, and returns the scalar value given by equation (8).

Check your function's results for various arguments with hand calculations (show your work) to verify that your function produces correct values.

(c) Write a function $\mathtt{MRAC2}(\mathtt{t}, \mathtt{x}, \mathtt{x_m}, \theta, \mathtt{k}, \mathtt{r}, \mathtt{a_m}, \mathtt{b_m}, \gamma_1, \gamma_2)$ which accepts the 10 scalar real arguments (NOT a 10×1 vector) and returns the 3×1 vector given by (9)

$$\begin{bmatrix} \dot{x}_m \\ \dot{\theta} \\ \dot{k} \end{bmatrix} = \begin{bmatrix} a_m x_m(t) + b_m r(t) \\ -\gamma_1 \Delta x(t) x(t) \\ -\gamma_2 \Delta x(t) r(t) \end{bmatrix} \tag{11}$$

Check your function's results for various arguments with hand calculations (show your work) to verify that your function produces correct values.

(d) Write a function $\mathtt{SYS4}(\mathtt{t}, \mathtt{p})$ which accepts the scalar real argument `t` and the 4×1 vector `p` where

$$p = \begin{bmatrix} x \\ x_m \\ \theta \\ k \end{bmatrix} \tag{12}$$

and uses the the functions `SCALDYN2()` and `MRAC2()`, to compute the return value

$$\begin{bmatrix} \dot{x} \\ \dot{x}_m \\ \dot{\theta} \\ \dot{k} \end{bmatrix} \tag{13}$$

for the case

$$\begin{aligned} r(t) &= 10 * sin(t) \\ \gamma_1 &= 1 \\ \gamma_2 &= 1. \end{aligned} \tag{14}$$

Your function should FIRST compute $u(t)$ as specified, and SECOND use `SCALDYN2()` and `MRAC2()` to compute the return values.

Check your function's results for various arguments with hand calculations (show your work) to verify that your function produces correct values.

(e) Use `SYS4`(), `ODE45()`, and `PLOT` simulate, print, and annotate the solutions to this system for TWO different initial conditions

$$\begin{bmatrix} x(0) \\ x_m(0) \\ \theta(0) \\ k(0) \end{bmatrix}. \tag{15}$$

i. Choose your initial conditions such that $x(0) \neq x_m(0)$, $\theta(0) \neq \theta^*$, and $k(0) \neq k^*$.

ii. If necessary, use *odeset* to increase the numerical accuracy of the simulation. Run the simulation for enough time for the system to clearly reach steady-state, something line the following:

```
p0 = [5;-5;0;0];
opt = odeset('RelTol',1e-8);
[t,y] = ode45('sys1',[0 30],p0,opt);
```

iii. Plot the following signals. Use the "legend" command to label the individual signals. Or label by hand. The plots can be in separate plots or sub-plots.

A. $x(t)$, $x_m(t)$, and $(x(t) - x_m(t))$ versus time

B. $\theta(t)$, and $(\theta(t) - \theta^*)$ versus time.

C. $k(t)$, and $(k(t) - k^*(t))$ versus time.

iv. Write a Lyapunov function for this system called `LYAP2()` which can accept the state vector output of ODE45. plot the value of a Lyapunov function for this system as a function of time.

v. Comment on (a) the convergence (or lack thereof) of the parameter estimates and (b) the scalar valued Lyapunov function $V(\cdot)$.

(f) Experiment with different gains, reference models, plants, and initial conditions. Nothing to hand in for this one.

Hand in *ANNOTATED* printouts of your plots and printouts of your m-files. Hand in your m-file functions by emailing them to me as ZIP file attached to your email. Put "530.647 LAB #3 m-files from YOUR FULL NAME" in the subject line.

Check to verify that the files you hand in run. If your matlab functions call custom matlab m-files that you have written (for this course or otherwise) be sure to include *all* necessary files.