1 System Identification and Linear Control

A scalar discrete-time system has the following dynamics:

\[ x(t + 1) = \lambda x[t] + g(u[t]), \]

where \( g : \mathbb{R} \rightarrow \mathbb{R} \) not necessarily linear.

a) If \( g \) is approximated to order 2 around the operating point \( u^* = 0 \), so that

\[ x(t + 1) \approx \lambda x[t] + \beta_0 + \beta_1 u[t] + \beta_2 u^2[t], \]

what should \( \beta_0, \beta_1, \) and \( \beta_2 \) be?

b) Suppose that \( x[0] = 0 \). We apply a sequence of inputs

\[ \tilde{u} = (u[0], u[1], \ldots, u(N - 1)) \]

and observe states \( x[1], x[2], \ldots, x[N] \). Derive the least-squares estimates of \( \lambda, \beta_0, \beta_1, \) and \( \beta_2. \)
2 System Identification

Let’s now look at how System Identification works in the vector case. Again you are given an unknown discrete-time system. We don’t know its specifics but we know that it takes one scalar input and as two observable states.

We would like to find a linear model of the form

\[
\tilde{x}[t + 1] = A\tilde{x}[t] + Bu[t] + \tilde{w}[t],
\]

where \(\tilde{w}[t]\) is an error term due to unseen disturbances and noise, \(u[t]\) is a scalar input, and

\[
A = \begin{bmatrix} a_0 & a_1 \\ a_2 & a_3 \end{bmatrix}, \quad B = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}, \quad \tilde{x}[t] = \begin{bmatrix} x_0[t] \\ x_1[t] \end{bmatrix}.
\]

To identify the system parameters from measured data, we need to find the unknowns: \(a_0, a_1, a_2, a_3, b_0\) and \(b_1\), however, you can only interact with the system via a blackbox model. The model allows you to view the states \(\tilde{x}[t] = [x_0[t] \quad x_1[t]]^T\) and it takes a scalar input \(u[t]\) that allows the system to move to the next state \(\tilde{x}[t + 1] = [x_0[t + 1] \quad x_1[t + 1]]^T\).

a) Write scalar equations for the new states, \(x_0[t + 1]\) and \(x_1[t + 1]\) in terms of \(a_i, b_i,\) the states \(x_0[t], x_1[t],\) and the input \(u[t]\). Here, assume that \(\tilde{w}[t] = 0\) (i.e. the model is perfect).

b) Now we want to identify the system parameters. We observe the system at the initial state \(\tilde{x}[0] = \begin{bmatrix} x_0[0] \\ x_1[0] \end{bmatrix}\), input \(u[0]\) and observe the next state \(\tilde{x}[1] = \begin{bmatrix} x_0[1] \\ x_1[1] \end{bmatrix}\). We can continue this for an \(m\) long sequence of inputs.

What is the minimum value of \(m\) you need to identify the system parameters?

c) Say we feed in a total of 4 inputs \(u[0], u[1], u[2], u[3]\) into our blackbox. This allows us to observe \(x_0[0], x_0[1], x_0[2], x_0[3], x_0[4]\) and \(x_1[0], x_1[1], x_1[2], x_1[3], x_1[4]\), which we can use to identify the system.
To identify the system we need to set up an approximate (because of potential disturbances) matrix equation

\[ D \hat{p} \approx \tilde{y} \]

using the observed values above and the unknown parameters we want to find. Suppose you are given the form of \( D \) in terms of some of the observed data:

\[
D = \begin{bmatrix}
    x_0[0] & x_1[0] & u[0] & 0 & 0 & 0 \\
    x_0[1] & x_1[1] & u[1] & 0 & 0 & 0 \\
    0 & 0 & 0 & x_0[0] & x_1[0] & u[0] \\
    0 & 0 & 0 & x_0[1] & x_1[1] & u[1] \\
\end{bmatrix}.
\]

For this \( D \), what are \( \tilde{y} \) and the unknowns \( \hat{p} \) so that \( D \hat{p} \approx \tilde{y} \) makes sense? Tell us what the components of these vectors are, written in vector form.

d) Now that we have set up \( D \hat{p} \approx \tilde{y} \), explain how you would use this approximate equation to estimate the unknown values \( a_0, a_1, a_2, a_3, b_0 \) and \( b_1 \) assuming the columns of \( D \) are linearly independent. In particular, give an expression for your estimate \( \hat{p} \) for the unknowns in terms of the \( D \) and \( \tilde{y} \).
(HINT: Don’t forget that $D$ is not a square matrix. It is taller than it is wide.)

e) What could go wrong in the previous case? What kind of inputs would make least-squares fail to give you the parameters you want?