Homework 7

This homework is due on Friday, March 10, 2023, at 11:59PM. Self-grades and HW re-submissions are due on the following Friday, March 17, 2023, at 11:59PM.

1. System Identification

You are given a discrete-time system as a black box. You don't know the specifics of the system but you know that it takes one scalar input and has two states that you can observe. You assume that the system is linear and of the form

$$\vec{x}[i+1] = A\vec{x}[i] + Bu[i] + \vec{w}[i], \tag{1}$$

where $\vec{w}[i]$ is an external small unknown disturbance, $u[i]$ is a scalar input, and

$$A = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}, \quad B = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \quad x[i] = \begin{bmatrix} x_1[i] \\ x_2[i] \end{bmatrix}. \tag{2}$$

You want to identify the system parameters (a_1, a_2, a_3, a_4, b_1 and b_2) from measured data. However, you can only interact with the system via a black box model, i.e., you can see the states $\vec{x}[t]$ and set the inputs $u[i]$ that allow the system to move to the next state.

(a) You observe that the system has state $\vec{x}[i] = \begin{bmatrix} x_1[i] & x_2[i] \end{bmatrix}^\top$ at time i. You pass input $u[i]$ into the black box and observe the next state of the system: $\vec{x}[i+1] = \begin{bmatrix} x_1[i+1] & x_2[i+1] \end{bmatrix}^\top$.
Write scalar equations for the new states, $x_1[i+1]$ and $x_2[i+1]$. Write these equations in terms of the a_i, b_i, the states $x_1[i]$, $x_2[i]$ and the input $u[i]$. Here, assume that $\vec{w}[i] = \vec{0}$ (i.e., the model is perfect).

Solution:

$$x_1[i+1] = a_1x_1[i] + a_2x_2[i] + b_1u[i] \tag{3}$$
$$x_2[i+1] = a_3x_1[i] + a_4x_2[i] + b_2u[i]. \tag{4}$$

(b) Now we want to identify the system parameters. We observe the system at the start state $\vec{x}[0] = \begin{bmatrix} x_1[0] \\ x_2[0] \end{bmatrix}$. We can then input $u[0]$ and observe the next state $\vec{x}[1] = \begin{bmatrix} x_1[1] \\ x_2[1] \end{bmatrix}$. We can continue this for a sequence of ℓ inputs.

Let us define an ℓ-length trajectory to be an initial condition $\vec{x}[0]$, an input sequence $u[0], \ldots, u[\ell-1]$, and the corresponding states that are produced by the system $x[1], \ldots, x[\ell]$. **Assuming that the model is perfect ($\vec{w}[i] = \vec{0}$), what is the minimum value of ℓ you need to identify the system parameters?**

Solution: There are 6 unknowns so we need 6 equations to properly identify the system. Each additional timestep gives two new equations. To form the 6 equations we need to give the black

box $\ell = 3$ inputs. Namely, given inputs $u[0], u[1]$, and $u[2]$, we can see the state at times $t = 0, 1, 2, 3$ to give us our six equations.

Notice that the initial condition on its own gives us no equations because the unknowns we are interested in do not impact the initial condition. They govern the evolution of the system, and hence the states at times $1, 2, 3$ each give us two equations.

Note that having 6 equations is a necessary, but not sufficient, condition for us to be able to invert the system to uniquely determine the system parameters. For example, if $A = I$ and $u[0] = \cdots = u[\ell - 1] = 0$, then we would only have two independent equations.

(c) We now remove our assumption that $\vec{w} = 0$. We assume it is small, so the model is approximately correct and we have

$$\vec{x}[i+1] \approx A\vec{x}[i] + Bu[i]. \tag{5}$$

Say we feed in a total of 4 inputs $u[0], \ldots, u[3]$, and observe the states $\vec{x}[0], \ldots, \vec{x}[4]$. To identify the system we need to set up an approximate (because of potential, small, disturbances) matrix equation

$$DP \approx S \tag{6}$$

using the observed values above and the unknown parameters we want to find. Let our parameter vector be

$$P := \begin{bmatrix} \vec{p}_1 & \vec{p}_2 \end{bmatrix} = \begin{bmatrix} a_1 & a_3 \\ a_2 & a_4 \\ b_1 & b_2 \end{bmatrix} \tag{7}$$

Find the corresponding D and S to do system identification. Write both out explicitly.

Solution:

Using eq. (4), we get

$$\begin{bmatrix} x_1[0] & x_2[0] & u[0] \\ x_1[1] & x_2[1] & u[1] \\ x_1[2] & x_2[2] & u[2] \\ x_1[3] & x_2[3] & u[3] \end{bmatrix} \begin{bmatrix} a_1 & a_3 \\ a_2 & a_4 \\ b_1 & b_2 \end{bmatrix} \approx \begin{bmatrix} x_1[1] & x_2[1] \\ x_1[2] & x_2[2] \\ x_1[3] & x_2[3] \\ x_1[4] & x_2[4] \end{bmatrix} \tag{8}$$

so

$$D = \begin{bmatrix} x_1[0] & x_2[0] & u[0] \\ x_1[1] & x_2[1] & u[1] \\ x_1[2] & x_2[2] & u[2] \\ x_1[3] & x_2[3] & u[3] \end{bmatrix}, \text{ and } S = \begin{bmatrix} x_1[1] & x_2[1] \\ x_1[2] & x_2[2] \\ x_1[3] & x_2[3] \\ x_1[4] & x_2[4] \end{bmatrix}. \tag{9}$$

(d) Now that we have set up $DP \approx S$, we can estimate a_0, a_1, a_2, a_3, b_0, and b_1. **Give an expression for the estimates of $\vec{p}_1$ and $\vec{p}_2$ (which are denoted $\hat{\vec{p}}_1$ and $\hat{\vec{p}}_2$ respectively) in terms of D and S.** Denote the columns of S as $\vec{s}_1$ and $\vec{s}_2$, so we have $S = [\vec{s}_1 \ \vec{s}_2]$. Assume that the columns of D are linearly independent. *(HINT: Don't forget that D is not a square matrix. It is taller than it is wide.) (HINT: Can we split $DP = S$ into separate equations for p_1 and p_2?)*

Solution:

Notice that eq. (8) can be split into two matrix equations, one for each of p_1 and p_2:

$$D\vec{p}_1 \approx \vec{s}_1 \tag{10}$$
$$D\vec{p}_2 \approx \vec{s}_2. \tag{11}$$

Since D isn't square, it isn't invertible. However, we can still find $\vec{p}_1$ and $\vec{p}_2$ that best satisfy the equation via least-squares, which gives the solution

$$\hat{\vec{p}}_1 = (D^\top D)^{-1} D^\top \vec{s}_1 \tag{12}$$
$$\hat{\vec{p}}_2 = (D^\top D)^{-1} D^\top \vec{s}_2. \tag{13}$$

Here, $D^\top D$ is invertible (i.e. the solution is well-defined) because the columns of D are linearly independent. This was proved in 16A, but for completeness we include it here.

Assume that the columns of D are linearly independent. Let $\vec{v} \in \mathbb{R}^3$ such that $(D^\top D)\vec{v} = 0$. Then $0 = \vec{v}^\top D^\top D\vec{v} = (D\vec{v})^\top (D\vec{v}) = \|D\vec{v}\|_2^2$, so $D\vec{v} = 0$. Since D has linearly independent columns, then $\vec{v} = 0$. This means that the nullspace of $D^\top D$ is $\{0\}$, so $D^\top D$ must have full rank and is invertible.

2. Identifying systems from their responses to known inputs

In many problems, we have an unknown system, and would like to characterize it. One of the ways of doing so is to observe the system response with different initial conditions (or inputs). This problem is also called system identification. It is a prototypical example of a problem that today is called machine learning — inferring an underlying pattern from data, and doing so well enough to be able to exploit that pattern in some practical setting. Go through the attached Jupyter notebook `demo_system_id.ipynb` and answer the following questions.

(a) In Example 2, we assume that instead of measuring the state $\vec{x}$, we are instead measuring a transformation of the state $\vec{y} = T\vec{x}$ where T is a full rank matrix. Assume that we perform system ID on our observations $\vec{y}[i]$ to recover A_y, B_y such that $\vec{y}[i+1] = A_y\vec{y}[i] + B_y u[i]$. **How do the identified A_y and B_y matrices relate to the original A and B matrices in the dynamics of $\vec{x}$?** Remember that our original state dynamics are $\vec{x}[i+1] = A\vec{x}[i] + B\vec{u}[i]$.

Hint: The answer is given in the Jupyter notebook but remember to show your work.

Solution: Using our given transformation that $\vec{y} = T\vec{x}$,

$$\vec{y}[i+1] = A_y\vec{y}[i] + B_y u[i] \tag{14}$$
$$T\vec{x}[i+1] = A_y T\vec{x}[i] + B_y u[i] \tag{15}$$
$$\vec{x}[i+1] = T^{-1}A_y T\vec{x}[i] + T^{-1}B_y u[i] \tag{16}$$

Thus, $A = T^{-1}A_y T$ and $B = T^{-1}B_y$. This can be rewritten as $A_y = TAT^{-1}$ and $B_y = TB$, which is exactly what is in the Jupyter notebook.

(b) **Please share your observations on Example 2. Comment on what impact a linear transformation of the state trace has on our ability to perform system identification.**

Solution: It's nice to see that a linear transformation of the state trace does not have a tremendous effect on our ability to perform system identification. Basically, this means that we have some leeway in choosing what data to observe in practice. In a circuit, for example, we would typically choose capacitor voltages and inductor currents as our state variables. However, it's difficult to make current measurements in real time in a non-invasive way, so we would prefer for our observations to consist of only voltages. As long as we can in principle recover the inductor currents as some linear combination of voltages (this is usually possible), then we can just measure those voltages and proceed as normal.

Furthermore, even if an unwanted state transformation occurs, we know that our estimate of the system does not change drastically. As we saw, the estimated eigenvalues are still correct. Furthermore, controllability and observability are preserved in coordinate changes, so the system we identified will have almost all of the same control-theoretic properties as the true system. Believe me, that's a relief!

(c) **Prove that for any full rank transformation matrix T, the eigenvalues of A_y and A from part (a) are the same.**

Solution: Assume that the eigenvalue eigenvector pairs of A are $(\lambda_1, \vec{v}_1), (\lambda_2, \vec{v}_2), \ldots, (\lambda_n, \vec{v}_n)$. Then, $A\vec{v}_i = \lambda_i\vec{v}_i$. We claim that $T\vec{v}_i$ will be the eigenvectors of A_y. We can see this with

$$A_y T\vec{v}_i = TAT^{-1}T\vec{v}_i = TA\vec{v}_i = T(\lambda_i\vec{v}_i) \tag{17}$$

$$= \lambda_i T \vec{v}_i \tag{18}$$

Thus, A_y also has eigenvalues λ_i with its corresponding eigenvector being $T\vec{v}_i$.

(d) **Please share your observations on Example 3. Comment on the impact that changing the noise magnitude, number of samples and number of states has on the system identification performance.**

Solution: From playing around with the system and state trace parameters, you may have noticed a few trends:

- Increasing the noise magnitude σ reduced the accuracy of the identified eigenvalues (that is, the identified eigenvalues were farther away from the true ones).
- Increasing the number of samples improved the accuracy of the identification.
- Increasing the number of states has a large impact on how accurate the identification is, for a fixed noise magnitude and number of data points.

The final point is important in practice. It suggests that, if we have some control over how many states we model a system with, and if all other things are equal, then we should choose to have fewer states rather than more, so that our system identification requires less data to be accurate.

This is a point that turns out to be extremely important in machine-learning more generally — we do not necessarily always want the most complicated model.

(e) **Please share your observations on Example 4. Comment on the sample efficiency of this method, i.e. do you need more or less samples for accurate system identification when given scalar observations rather than the entire state vector?**

Solution: This example took you beyond what you have learned in lecture. It involved figuring things out without being able to observe the state and instead just seeing scalar observations of the state.

You probably noticed that identifying a system with scalar observations requires a longer sample trace to be accurate than identifying a system with state observations does. The reason why is pretty clear: you only have one scalar point of data at every time step now, instead of a vector. Nonetheless, it's a really great thing that we can still do system identification with scalar observations at all! In many applications, a scalar output is all that's available.

Having to select a value for n with no prior knowledge introduces more trial-and-error than we would like, but in the method used in this section it's a necessary evil. But is it necessary in general? I shouldn't overly anthropomorphize the data, but it must know how many states there are, since it was generated by the system. So it seems like some knowledge of the "true n" is hiding somewhere in the data, if only we know how to look...

We will return to this issue later in the course.

(f) **Please share your observations on Example 5. Comment on how important the model size is for this setting.**

Solution: This is another example that takes you a bit beyond what you have seen in lecture, but in a natural way. What happens if your model size is wrong? Do you have to get the model size right?

It's interesting to see that the effect of many eigenvalues near the origin can be effectively approximated by just a couple of eigenvalues in a smaller system. Just like example 3, this suggests

that there is such a thing as "too many states". Basically, 13 out of the 16 states of this system contribute almost nothing to the behavior of the system– we were able to throw them away and still capture the important behavior.

So what happens because we are ignoring these states and the true complexity of the underlying model? We will hopefully see in a later homework that what this does is contribute to the disturbance that our estimated model experiences. It not only has a disturbance because it didn't estimate the parameters of the model perfectly (say because of observation noises), but also because it chose a model that was simpler than reality. But as long as we can be robust to this disturbance, we are still fine.

Contributors:

- Nikhil Shinde.
- Ashwin Vangipuram.
- Sally Hui.
- Alex Devonport.
- Tanmay Gautam.