# Lecture 1: Introduction to Dynamic System Learning and Estimation

The process of developing models of dynamic systems from experimental data is called *system identification* or *system learning*, and using measurements of a system's output to infer its internal state is called *state estimation*. The need for system identification/learning and state estimation is ubiquitous in science and engineering, and this course will explore these topics in the context of dynamic systems modeled as ordinary differential equations. We will survey a broad range of topics including classical techniques and selected machine-learning-based methods. The class will include both theory and practical implementation. Please refer to the syllabus for more information on the topics that will be covered and relevant textbooks.

## What is a Dynamic System?

A *dynamic system* is a collection of components that act together to perform a specific task. Systems have inputs that produce a response observed in the output. As engineers, we decide how to define our system — we typically choose the the inputs and outputs to be the most pertinent variables that are relevant to our analysis or design. Then, using physics and other modeling principles, we develop the governing equations to relate them. Consider the following examples:

- e.g., a car suspension is a mechanical system.
  Input: bump in the road. Output: oscillation of the car frame.
- e.g., a servomotor is an electromechanical system.
  Input: Pulse-width-modulated (PWM) signal. Output: Servo arm position.
- e.g., a water boiler is a thermal system.
  Input: Voltage to heating element. Output: Water temperature.

The components of a system are the individual functioning units of a system

- e.g., springs, gears, motors, resistors, and pumps may be a partial list of components found in a car suspension, a servomotor, or a water boiler system

Dynamic systems are not limited to mechanical engineering systems and dynamics arise in diverse fields ranging from chemistry and biology to epidemiology and astromechanics.

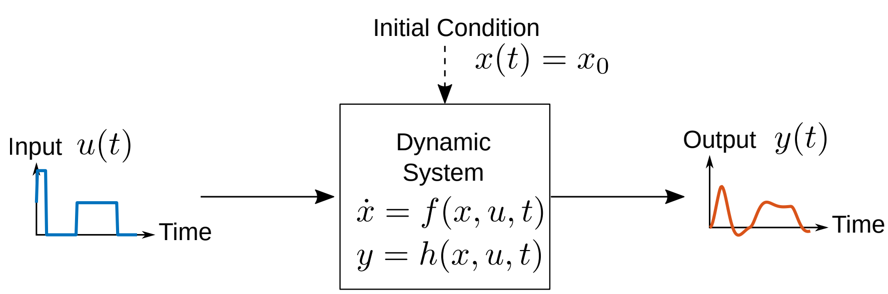


Figure 1: Example dynamic system input/output

It is useful to think of a dynamic system according to a *block diagram* that graphically depicts the system (or a component of a system) as a block with input and output arrows, as shown above. The left-hand side of the above diagram depicts the time-history of an input signal and the right-hand side shows the output signal. Input signals can have different shapes, such as ramps, sinusoids, step inputs, or impulses, that determine the output signal (system response) for a given initial condition. The outputs from one block can form the inputs to another block to create more complex systems.

A *static system*'s output depends only on the current input. For example, if the input is $u(t)$ then for a static input-output system model the output state $y(t)$ would be expressed at any given time $t$ as some function $y = h(u, t)$. On the other hand, a *dynamic system*'s output depends on both the input signal, time, and an *internal state* that evolves according to an ordinary differential equations (ODEs) of the form

$$\dot{x} = f(x, u, t) \tag{1}$$

where $x(t)$ is the state variable, $\dot{x}(t)$ is the state derivative (also called *state-rate*), $u(t)$ is an input, and $t$ is time. Even without an external input (i.e., ignoring $u(t) = 0$) the system state changes over time depending on the initial conditions, $x(t_0) = x_0$. In this course we will primarily concern ourselves with dynamic systems modeled in *state-space* form (as above).

The *output* of a system $y(t)$ is the time-varying signal of measuremenets as the system responds. In practice, the sensors used to measure the system state may alter the signal in some way (e.g., they might add noise or scale/bias the signal). If our sensors directly measure all states then $y(t) = x(t)$. If the output is a linear combination of the states then $y(t) = Cx(t)$ (here we're assuming $x(t) = [x_1(t)\ x_2(t)\ \cdots\ x_n(t)]$ is a vector of $n$ state variables and $C$ is a matrix). Or, if the output is a nonlinear transformation of the states then $y(t) = h(x(t))$. Other descriptions of the output may also include additional terms to describe noise or dependency on control input or time.

## What is State Estimation?

The goal of state estimate is to infer the state $x(t)$ of a dynamic system using only knowledge of the inputs $u(t)$ and (potentially noisy) outputs $y(t)$ of the system which correspond to the right and left sides of Fig. 1. State estimation is interesting to engineers for at least two reasons:

- *Engineers need to estimate system states to implement state-feedback controllers.* Control systems are most easily designed assuming that the internal state of a dynamic system is available for feedback and, when this state is not available (or is of poor quality), an estimate can be used instead. The entire system state may not be available as an output for many reasons, e.g., the sensors available can only measure a subset of the state or perhaps the sensor reports a quantity that is related to the state by a linear or nonlinear transformation. As an example, consider a state consisting of the $(x, y)$ planar position of an object. A range or bearing sensor produces an output $y = \sqrt{x^2 + y^2}$ or $y = \text{atan}(y/x)$ that is related by a nonlinear transformation to the state.

- *Often an engineer needs to estimate the systems states because they are interesting in their own right.* Even if the system state is not needed directly by a control system it may be useful for other real-time systems or post-processing. For example, a health monitoring system may rely on the internal state of a mechanical system to determine its status. Or, in robotics, a common use of state estimation is for mapping—to infer the geometry of obstacles in its surroundings from noisy measurements.

It is desirable to design state estimators that are optimal and that take advantage of information encoded by the known dynamics of the system and statistics about the measurements and other sources of noise and uncertainty. By defining an error metric we can formulate state estimation algorithms that optimize (i.e., minimize) this error. Generally speaking a state estimator takes on a sequence of output measurements $y_1, y_2, y_3, \ldots, y_k$ at time instants $t_1, t_2, \ldots, t_k$ and produces an estimate of the current state $\hat{x}_k$ (the " ˆ " (hat) symbol denotes an estimated quantity). State estimation is also called *filtering* when the estimate is needed in *real-time* (e.g., to be acted on). A related forward-looking concept is *prediction* (i.e., estimate the state at a time in the future relative to the currently available measurements) and *smoothing* (i.e., estimating the state at a time in the past relative to the currently available measurements). The relationship between smoothing, filtering (state estimation), and prediction is illustrated below.

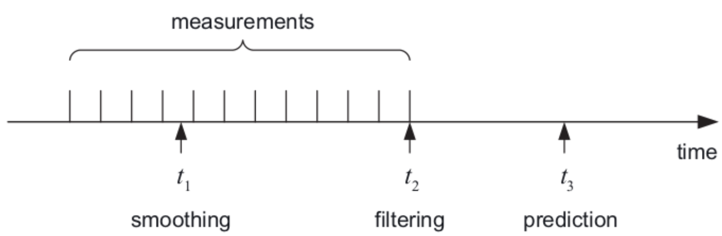


Image source: [Gibbs, 2011, Chapter 1]

## What is System Identification / Learning?

State estimation generally assumes the model of the system is already known. However, this is often not the case and system identification is the process of determining a model of the system's dynamics, $\dot{x} = f(x, u, t)$ from observed input/output data. In this course we use the terms *system identification* and *system learning* interchangeably. The former term has a longer history and refers to a body of well established techniques on deriving dynamical models from experimental data. The latter term has connotations of more modern, data-driven and machine-learning-based approaches. System identification/learning is interesting to engineers for the main reason that:

- *System models are needed to use model-based tools for simulation, control-design, and analysis.* As alluded to earlier, state estimation and control design often assume that an accurate system model is available. However, such models are difficult to obtain and significant effort is often needed to establish accurate models of dynamic systems. For example, an aircraft will undergo extensive fluid dynamic simulation studies and wind tunnel tests to establish a model of the aerodynamic forces before the aircraft is even constructed. Once the aircraft is built a series of carefully planned flight tests are needed to confirm and adjust the system model. Even after all of this effort the model is often valid only in the region of the aircraft's operating space (e.g., flight speeds, maneuver conditions) that was explored experimentally and modeled. The model outside of this parameter space can exhibit much different behavior and would require further system identification. However, it is important to note that we only need a model that is "good enough" for the task at hand. Control system are responsible for handling any discrepancies caused by modeling errors — but these should be sufficiently small to ensure good performance and stability.

When we think about system identification/learning we can distinguish between two cases:

- *Parameter identification*: In this case the engineer has a trusted model of the system (e.g., derived from first-principles or based on prior experience) and it remains to determine the parameters of the system that best match the data. For example, consider a precisely fabricated pendulum apparatus with a very thin rod and heavy mass at one end. The equations of motion modeling this system are accurately described by a first principles model that includes the parameters: length $L$, gravity $g$, and rotational damping coefficient $b$:

  $$\ddot{\theta} + b\dot{\theta} + \frac{g}{L}\sin\theta = u(t)$$

  While $L$ is easily determined (and $g$ is known), the damping coefficient of the bearing can be challenging to measure directly. System identification can be used to determine this unknown parameter by comparing experimentally derived data sets of the pendulum swinging under different initial conditions or control inputs.

  The field of *optimal experimental design* concerns itself with designing the experiments to be as informative as possible such that parameters can be estimated accurately with the minimum effort for experimental runs.

- *Model determination:* Sometimes the engineer does not have a good model of the system to begin with (e.g., the system might be difficult to model from first-principles or it exhibits some complexities that are not captured by simpler models). However, it is likely that they have some intuition behind what are the key variables in the model from which they can select a set of candidate models that might be appropriate. Model determination requires simultaneously identifying the parameters of a particular model that best fit the data while also selecting the model from a set of candidate models. This can be achieved by defining some criterion that quantifies how well the model fits the data and validating it on a new data set. If the model needs to be revised the process is repeated. Otherwise, the model is accepted. This procedure is summarized in the flowchart below. Often the simplest model is chosen that explains most of the data — in line with the principle of *Occam's razor*.

## References

[Gibbs, 2011] Gibbs, B. P. (2011). *Advanced Kalman Filtering, Least-squares and Modeling*. John Wiley & Sons.

[Ljung, 1995] Ljung, L. (1995). *System Identification Toolbox: User's Guide*. Citeseer.

[Simon, 2006] Simon, D. (2006). *Optimal State Estimation: Kalman, H infinity, and Nonlinear Approaches*. John Wiley & Sons.

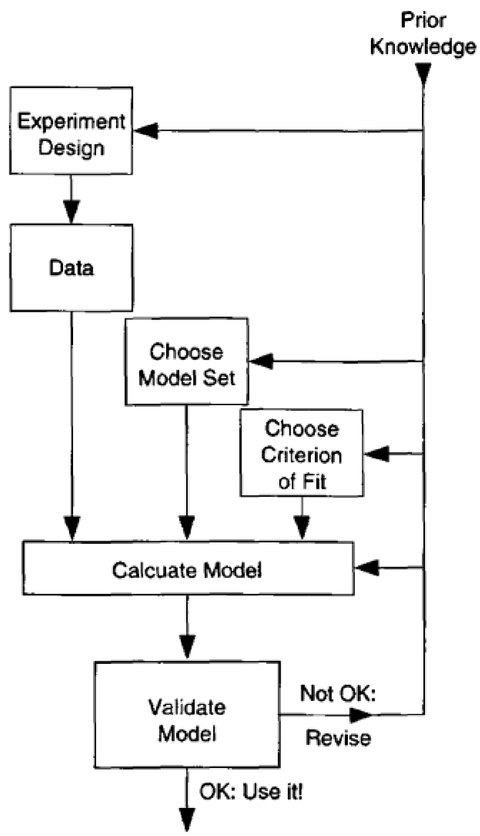


Image source: [Ljung, 1995].