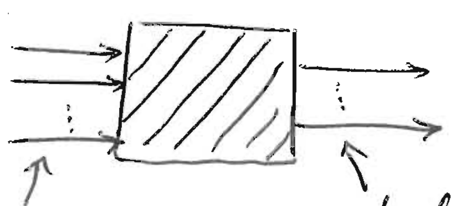
# ECE 618 – Estimation & System Identification
## Lecture 1 – Motivation

Outline
1. What are Estimation & System Identification?
2. In what context are they used?
3. Features of System Identification Problems (Lecture 2)
   - Auto Regressive Moving Average Models (ARMA)
   - Formulation of Sys Id Problems
   - Examples
   - Factors that Affect the Solution

1. What are Estimation & System Identification?

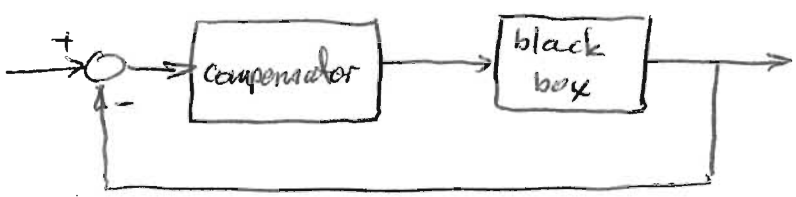
We begin with a "black box"



It has inputs, outputs – and, inside, it processes the inputs (assume in real time) in some manner that generates the outputs. Initially, we don't know how (and we may never know the details).

As a control engineer, we want to "do something" useful with the black box. It may be a robotic arm, or an induction motor, or the active truss structure of a telescope's mirror assembly. Perhaps we have an idealized model of how we would like the black box to behave.

We try to force it to behave the way we wish — we embed it in a control system. The classic diagram:




But, from undergraduate controls courses, we know we can't choose (or "design") a compensator without knowing something about the dynamic behavior of the black box.

For more advanced controllers, we may need more — we need information about the state of the model of the black box.

An important distinction:

The black box is a physical system (at least sometimes it is...). For example, it has dynamic properties because it obeys the laws of physics.

Think of an airplane. Its flight path responds to the action of moving air on its wings and body (lift and drag, but also flexure, and a gradually changing center of mass as fuel is consumed and as passengers move), and to the influence of gravity.

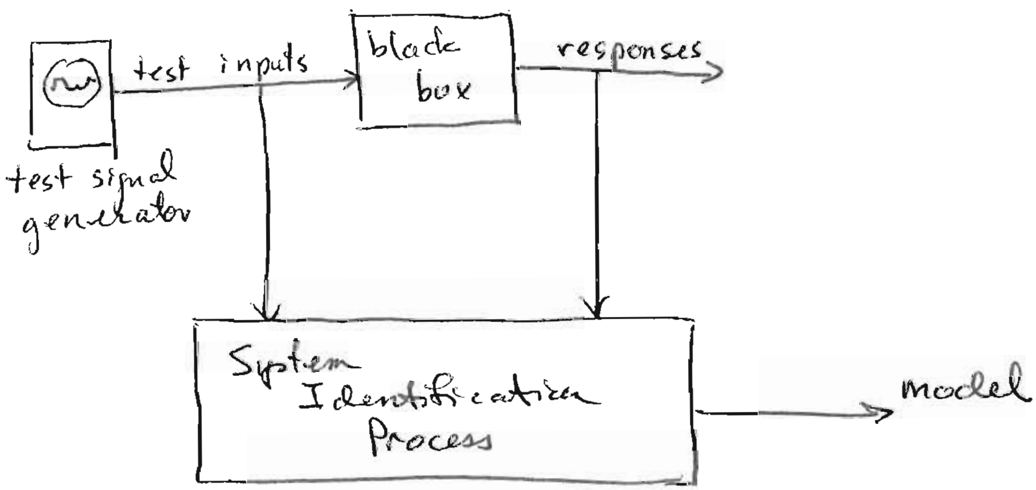
Given enough computing power, we can model the airplane to any desired level of detail — a finite element model, perhaps, that incorporates flexure and heating effects.

But, no matter how detailed the model, it is still a model and does not substitute for the physical system.

In addition, as we increase the complexity of the model, we usually increase the required knowledge about the physical system — represented by model parameters.

In contrast to a <u>physical</u> <u>system</u>, a <u>system</u>, or <u>dynamical</u> <u>system</u>, is just a mathematical abstraction that approximates ~~the~~ some behaviors of the physical system. Often, we will just call it a <u>model</u>.

This course introduces the methods that can be used to find appropriate (e.g., accurate) models of physical systems through experiment.




This is System Identification: Given inputs and outputs (as a function of time), find an acceptable model.

Note that I said "acceptable" and used the words "appropriate" and "accurate" — not "best" or "optimal". Models are only "best" or "optimal" w.r.t a specific set of possible models, of many sets.

Furthermore, excess complexity can yield:

(1) Implementation problems (more expensive h/w and more s/w development and more bugs).

(2) Difficulty in finding suitable values for all the parameters.

(3) Difficulties in testing and validation of the model (or a design based upon the model) against the physical system.

(4) Fragility, or non-robustness, of the resulting design.

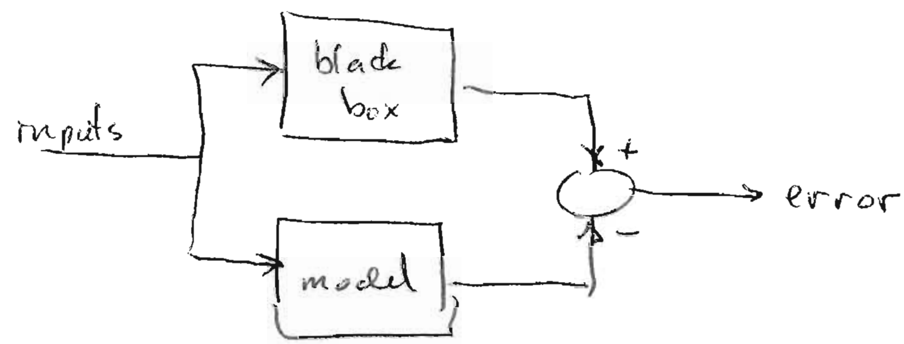
Simply put: Simple is usually better. (KISS).

However, a model must be able to capture features of the physical system that are relevant to the application.

For example, a 1st order linear model can not predict overshoot.

There is a trade-off between complexity and accuracy — one that requires engineering judgment (meaning an automatic model design method is not available, and, for the general case, is not likely).

Suppose you have an accurate (even exact) model, and you want to compare its behavior to the physical system's.




At time zero, you "flip the switch" to start both the black box and the model.

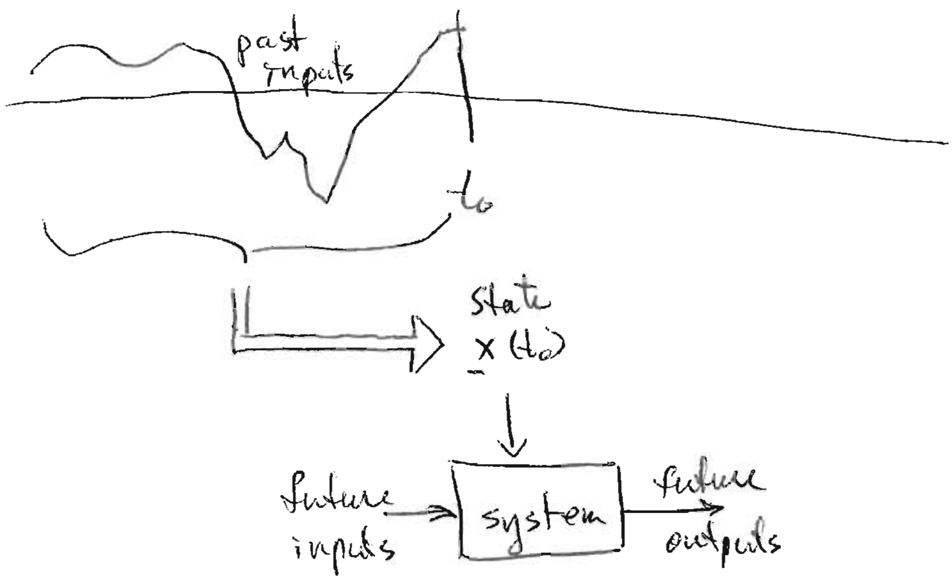
Under what conditions will the error be small?

We know that if the model and the black box are identical and if their <u>initial</u> conditions are the same, then the error will be zero.

What are these initial conditions? In 511, we introduced the concept of <u>state</u>:

> <u>Def</u>: The state of a system at time $t_0$ is the information necessary to uniquely determine, for all $t > t_0$, the system's outputs for all possible known inputs.

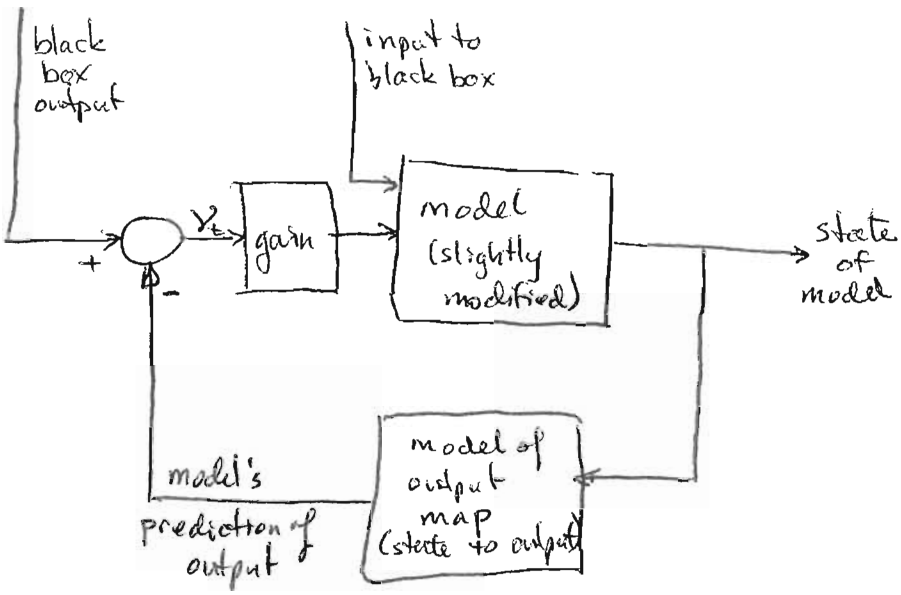
Another way of saying this is the state captures, at time $t_0$, all of the effects that past inputs can have on future outputs.




That is fine if the system and the black box have identical dynamics – but they won't – ever.
Now what?

If the model is "sufficiently accurate" it should be possible to find a value for the model's state, $\underline{x}(t_0)$, at time $t_0$ that makes the error between the model's output and the black box's output small.

In fact, you can design a control system to make the model's state $\underline{x}(t)$ track this minimizing trajectory in real time, as new input and output data arrive. This is state estimation.




$\gamma_t$ = the <u>innovations process</u> = the difference between the actual observed output of the black box and the output predicted by the model.

The innovations process is new information to the state estimator — it is a record of the black box's behavior that was not predicted by the model.

The state estimator's gain adjusts the state of the state estimator (called the <u>state estimate</u>) to (we hope) cause the model to more accurately predict future outputs of the black box.

In the linear case, this state estimator is called the Kalman (or Kalman-Bucy) filter, or the Wiener filter in steady-state.

There's a nice connection between state estimation and system identification. (You might have guessed this from the title of the course!)

What if the model is not a single fixed model, but is a parameterized family of models?

$$\{M(\underline{\theta})\}_{\underline{\theta} \in \Theta}$$

The parameter vector $\underline{\theta}$ takes values in a set $\Theta$ (capital $\theta$) and uniquely determines a model $M(\underline{\theta})$.

In this case, why should we not consider the parameter $\underline{\theta}$ a part of the model's state?

In fact, we can, in which case the gain of the last figure modifies not only the (traditional) state of the model, but also the model's parameter vector.

If the gain is chosen correctly and "everything else" (a lot of math) works out, then the state estimator not only adjusts the state, but also dynamically tunes the model to minimize error!

The "Holy Grail" of systems theory and control design — often unattainable — is to be able to control a system that is initially unknown or poorly known.

The approach — Adaptive Control — combines system identification, state estimation, and controller design.

If you have a model of a physical process that is reasonably accurate, you have a decent chance of controlling the process (making it behave "better").

So, why not identify the model and control the process at the same time? There are many approaches (see, for example, Model Reference Adaptive Control, or MRAC), a few of which are guaranteed stable if their conditions are satisfied.

As you might ~~sp~~ suspect, however, there are dangers. For example, if the initial model $M(\theta_0)$ is awful, chances are good the controller will do something stupid.

These dynamically tuned models are the heart of <u>recursive identification</u> (in system identification books and articles) and the <u>Extended Kalman (Bucy) Filter</u> (in books and articles on filtering).
We will cover them both ways — and, ~~but~~ by the way, the notation is different. Each field is established (entrenched), so live with it and learn both notations (no choice if you want to understand the literature).

So, where does this go? (What can be done with this mathematical stuff?)

The buck can stop here. Examples: GPS, enhancement of images (such as removal of blur due to the atmosphere), inertial navigation (the second major application), and target tracking (the first, during WW II). But, there's more.

Stability results in adaptive control therefore tend to come with some fine print. An example: The closed-loop system may be guaranteed asymptotically stable, after an arbitrarily large excursion in its response (such as tracking error). The problem being that physical things tend to break when they experience these excussions! The risk of this can be reduced by trying to ensure that initial values of parameter estimates are near reality.

So, enough generalities. Up to this point, some valuable concepts have been introduced, which you will see more of as the course unfolds. Among these:

1. The difference between a "physical system" and a "system", "dynamical system", or "model".
2. Controllers need models, and the models have to accurately capture behaviors of the physical system in order for the closed loop system (controller + physical system) to work well.
3. System Identification procedures capture test input/output data from a physical system and (we hope) find an acceptable model.

4. There is a trade-off between model complexity and model accuracy, and between complexity and fragility, robustness, and difficulty of use and implementation.

5. A system's state uniquely determines future outputs given any known inputs. It also captures all the effects past inputs can have on future outputs.

6. A state estimator is a control system applied to a model, to force its state to track a value (or trajectory) that minimizes error between the model's prediction of a physical system's output and the actual measured outputs.

7. This difference between actual and predicted outputs is called the innovations process, which captures information about the physical system that is new to the model.

8. The linear state estimator is called the Kalman-Bucy Filter, or, in steady state, the Wiener Filter (after the people who invented it).

9. System identification problems can be cast as state estimation problems when the model is one of a parameterized family of models. This is done by augmenting the state of the estimator with the parameter vector and allowing the estimator to "tune" the parameter vector as it updates its state estimate.

10. This is called "recursive identification" in the systems identification community, and "Extended Kalman-Bucy Filtering" in estimation theory and practice. (Note: The techniques are <u>not</u> identical – just similar. The notations are significantly different.)

11. If recursive identification and controller design are combined, the result is the field of "Adaptive Control".

12. Adaptive controllers can sometimes be proven asymptotically stable, but there is often a catch – such as the possibility of arbitrarily large excursions in, for example, tracking error before the response "settles down". Carefully chosen initial conditions are important (perhaps by using Sys Id before "closing the loop").