System Identification

Overview

Until now, we have simulated the "**direct problem**" in which the input, such as initial conditions and forcing functions, are known, and the system is known, but the output is not. In this lesson, we learn about the inverse problem, in which the outcome is known, but the reason for the output is not. Specifically, we consider the situation where the input and output are known, but the underlying system that converts one to the other is not, and we are trying to determine what it is.

Parameter estimation is the process of identifying parameters for a systems model that cause it to fit the data as well as possible.

System identification is the process of identifying a systems model that can explain the experimental observations. That is, system identification is finding a set of equations that can be fit to the data. System identification thus includes both finding the best system and finding the best parameters for that system.

We do system identification for a variety of reasons, including the following examples:

Describe data. We may want to fit models to data to identify a small number of parameters that represent the data more efficiently that does the full data set, and allows for a better comparison. For example, we have mentioned SPR experiments in this class already. SPR data are usually fit to models to obtain to rate constants (kon and koff) and the ratio of these, the KD = koff/kon. These values are a much more convenient way to represent the data than viewing the raw data curves, and more meaningful than choosing any single data point to represent the data. If we are determining parameters that have physical meaning, a mechanistic model is needed, but in other cases an empiric model may be sufficient to describe or compare data sets.

Prediction. We may want to build an accurate model that will allow us to predict the behavior of the system in different conditions that those used to collect the data. The prediction can be used to better understand the system, or for design to figure out how to use the system to obtain specific behavior or to incorporate the system into a larger system. An empiric model is generally sufficient for such prediction.

Hypothesis testing. Parameter estimation can be used to determine whether a systems model can quantitatively explain the data. Successful fitting shows that the model or hypothesis is quantitatively consistent with the data, or can disprove a hypothesis by showing it is not consistent. If the hypothesis is disproved, this can lead experimentalists to ask what modification to the known information could provide a quantitative fit. If the model is consistent, this does not prove the hypothesis, but does support it. If one model is consistent and all other proposed models are not, then this provides very strong support for the consistent

model. Hypothesis testing requires a mechanistic model, although it is fine for one or more components of the model to described empirically.

Design. Models are often used for quantitative design, in which one designs a system to create a specific behavior or outcome. The engineer can usually vary a subset of parameters within certain bounds. So in this case parameter estimation will involve finding which parameters within those restrictions provide behavior that is closest to specifications. Design requires a mechanistic model.

Almost all modeling of interest to experimentalists involves the inverse problem.

To do parameter estimation or system identification, we need to address two challenges:

- How do we define best fit?
- How do we find the parameters and model to get the best fit?

The methods for doing this is described below.

Performing system identification involves the following steps:

1. Propose a model structure
2. Identify the data and how to it relates to the model.
3. Propose an error model for noise in that data.
4. Use this information to write a scalar objective function to be minimized in order to minimize the error.
5. Optimize the parameters to minimize the objective function (e.g. use MATLAB 'fminsearch')
6. Evaluate the residuals between model and data
7. Compare alternative models if needed.

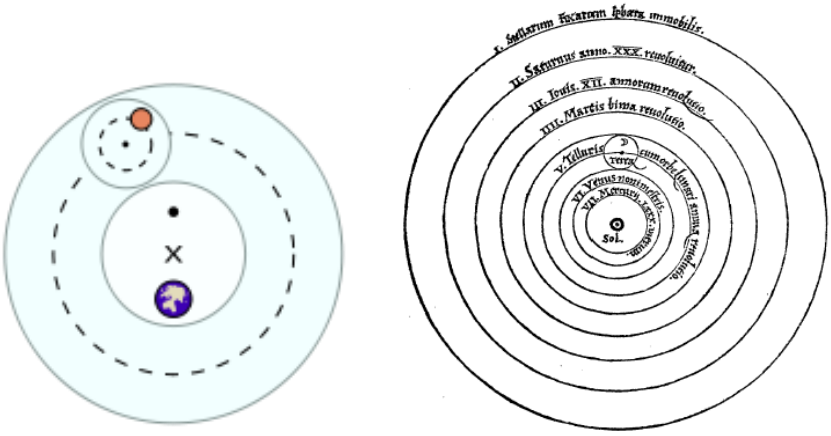
Determining Model Structure and identifying data

Model structure. Even if we are testing a specific hypothesis that determines some assumptions of the model, we need to determine how complex the model should be. There may be some processes that we know occur, but that may or may not contribute to the observed behavior. For each process that is included, there is also a question as to how much detail should be used to model it. Thus, there is always a question as to the appropriate complexity of the model.

Model complexity refers to the number of variables and/or parameters. A **parsimonious** model is one with the minimum complexity that addresses the question.

Data complexity refers to the number of measurements that indicate different types of behavior. This is not as simple as the number of measurements, since some techniques involve multiple measurements so close in time that most of the data points do not give new information. That is, data complexity can be thought of as the simplest number of parameters that could be used to fit the data given an appropriate model.

For meaningful system identification, the model must be no more complex than the data. A model that is more complex than the data can nearly always fit the data, and indeed do so equally well with many sets of parameters. If we want an empiric (black box) model, then clearly the one with the least parameters is most effective. A model with more parameters will often be '**overfit**' meaning it will fit the measurements perfectly but will swing so wildly in other locations that it is highly unlikely to be predictive of new measurements. A classic example of an overfit model is the geocentric model for astronomy, which assumed that all planets and the sun revolved around the earth. To get this model to fit measurments about the positions of the sun and other planets, it was necessary to add complexity to the model in the form of 'epicycles in which these objects went on orbits within orbits. The complexity allowed the models to fit existing data, but the models were not predictive, and as soon as new data was collected, this data contradicted the model, so new models were developed with yet more complexity. When Copernicus introduced the heliocentric theory that all planets revolved around the sun, and the moon around the earth, and that all orbits were elipses, this created a simpler model with fewer parameters that not only fit the data but was predicted, so that it still fit when new data was collected.



Model validity refers to whether the model is robust for the question being asked. While we have described above the motivation to simplify the model, the model won't be valid if we make simplifying assumptions that contradict the essential aspects of the question. For a simplistic example, if we seek to predict the effect of ligand density on a targeted drug delivery particle, the model must include some process that responds to ligand density, although this process may be represented as a empiric or mechanistic model within the larger model.

If the question being asked requires a complex model, then it is necessary to obtain data with more complexity than the model. One way to do this is to test components of the model, to fit individual parameters from an in vitro study, or to fit a small number of parameters describing a process within the larger model. When a complex mechanism is to be tested, a common practice is to use previously published data to characterize these model components. That is, the structure and many of the parameters of a model may be known from prior knowledge, so that just a small number of parameters are fit to determine if the entire mechanism can explain some observed behavior. One of the issues that arises in such a case is that some of the experiments

used to test components of the model may be slightly different from the conditions of the experiments used to study the entire system. This can involve studies in different species, or in different physical conditions such as pH and temperature. We will return at the end of this lesson to address mechanisms to combine new data with previously published data to fit a single model.

Identifying data. Let $\vec{z}$ be the set of data we are want to compare to the model. First we need to identify outcomes of the model that correspond to this data. We will refer to these outcomes as the vector $\vec{y}$, where $y(i)$ corresponds to $z(i)$. The outcomes are some function of the vector trajectory of system states, $\vec{X}(t)$. That is: $\vec{y} = \vec{f}\left(\vec{X}(t)\right)$. The outcomes may be simply the values of one or more variables at specified times, or could be equilibrium values, combinations of variables at a specified time such as $X_i(t) + X_j(t)$ or even things like the rate of change of a variable, the area under a curve (integer) of a variable, the frequency of oscillations, or the time and value of a maximum of a variable.

Theory and Computations of Parameter Estimation.

To estimate parameters, we start with the data $\vec{z}$ and a model structure that results in model outcomes $\vec{y}$. Now our job is to identify an optimal set of parameters $\vec{p}$ that provide the best fit of the model outcomes to the data. To do this, we assume that when the model is correct, the difference between the model and the data is noise due to measurement error. We refer to the difference between the measurement z(i) and the value $y(i, \vec{p})$ predicted by the model for that measurement as the **residual** res(i).

$$res(i) = z(i) - y(i, \vec{p})$$

We will try to jointly minimize all these residuals, but we cannot minimize all at once. There will always be a trade-off in which different parameter values minimize some at the expense of others. To determine which of these parameter values is best, we need a model for error. That is, we need a theory that says how large the error is expected to be for each measurement, so that we tolerate errors of this size, but not errors that are significantly larger. We assume that each measurement error has a zero mean (so there is no systematic error), is normal distributed, and is independent from other errors. This means that the model for each measurement error is completely determined by a single value, the standard deviation, that can be unique to that measurement. We thus refer to the error model as the set of $\sigma(i)$, or the vector $\vec{\sigma}$.

The two most common error models are as follows. The **constant coefficient of variation** (CV) assumes the error is a constant percent of the data. That is, $\sigma(i) = CV * z(i)$, or $\sigma(i) = CV * y(i, \vec{p})$. The **constant standard deviation** (SD) model assumes that the error is a constant that is independent of the data and model value: $\sigma(i) = SD$. It should be noted that there may be different types of measurements in the data set, in which case there may be a different error model for each of these subsets of the data. If we understand the source of error in the experiments, or have multiple measurements for each z(i), we can estimate $\sigma(i)$. However, it should be noted that we should not use the standard deviation measured from multiple measurements for our error model $\sigma(i)$. This is because the standard deviation of several

measurements is only an estimate of the true error expected if we took an infinite number of measurements. We don't want to overly constrain a model to fit a point z(i) in which duplicate measurements happened to lie close together but could both be quite far off the true value. At any rate, we assume an error model to the best of our ability, but if it turns out we are wrong, we will notice this after we fit when we evaluate the parameter estimates.

We use this error model to weight the residuals, and we square the weighted residuals to penalize large differences. Thus, we define an objective function J to be the sum of the square of the weighted residuals:

$$J(\vec{p}) = \sum_i \left(\frac{res(i)}{\sigma(i)} \right)^2 = \sum_i \left(\frac{z(i) - y(i, \vec{p})}{\sigma(i)} \right)^2$$

This function is a scalar, and so we alter all the parameter values to minimize it.

Because we do not in general have an analytic expression for the outcomes y, we also do not have an analytic expression for the objective function, so we cannot set derivatives to zero or similar approaches to minimize J. Instead, we use a numerical minimization function such as fminsearch in MATLAB. These algorithms only find local minima, so we need to use many initial guesses and compare the local minima thus found to determine which is the global minimum. The set of parameters that give the lowest value for J is the best set of parameters that we could find, but it remains possible that we have not found the global minimum.

If we find many quite different sets of parameters that provide reasonable or even identical fits, this means that the model is **overfit**. That is, the model is too complex for the data and we have learned that the data can explain the model, but we have no confidence on the parameters we have identified. In this case, it is advised to use a simpler model or to find more data that can be used to constrain the parameters further.

If we do identify a single set of parameters that fits the data the best, we are ready to do two things to evaluate the fit of the model. First we plot the data and the model together to use our visual processing abilities to determine if the fit is reasonable at all. If the fit deviates unacceptably from the data, we already know that we have not found a fit of the model to the data. If the fit looks reasonable, we need to evaluate the residuals. They should have zero mean. In addition, they should be independent, so a plot of the residuals or weighted residuals over time should not show any time-dependent pattern. Finally, the residuals should match the model for the error. If we assumed CV, then plotting res(i) vs z(i) should show a linear trending 'envelope' in which the res(i) fluctuate, but the deviation from zero increases with z(i). If we assumed SD, then there should be no trend in this plot. Alternatively, we can plot the weighted residuals, which should always show no trend with time or z(i).

System Identification Summary

- System identification is the process of identifying a model that best describes the system that fits, or describes, a set of measurements. Depending on the question to be asked, the

best model may be mechanistic or empirical, simple or complex, but the complexity of the model should not exceed the complexity of the data to be fit.
    - When identifying a model, we think about model structure, model complexity, data complexity, and model validity.
    - Model complexity refers to the number of variables and/or parameters included in the model.
    - Data complexity, however, refers to the number of measurements that indicate different types of behavior. This is separate from the number of data points collected for a single experimental condition.
    - Model validity refers to whether the model is robust for the question being asked.
- Mathematically, the process is as follows.
    - Let $\vec{z}$ be the vector of n measurements ().
    - Assume a model for the data.
    - Let $\vec{p}$ be the vector of m unknown parameters in the data model.
    - Let $\vec{y}(\vec{p})$ be the vector of n values that are any combination of variables and parameters in the model, which correspond to the n data points.
    - Assume a model for the error, that defines each element of $\vec{\sigma}$ as a function of the corresponding element of $\vec{y}$ or $\vec{z}$. The most common models are a constant standard deviation model: $\sigma_i = SD$, and a constant coefficient of variation model: $\sigma_i = CV * z_i$ or $\sigma_i = CV * y_i$.
    - The weighted residual for the i^{th} data point is the difference between model and data divided by the expected error: $wres_i = \frac{y_i(\vec{p}) - z_i}{\sigma_i}$
    - We then minimize the objective function: $J = \sum_{i=1}^{n} wres_i^2$, or $J = \sum_{i=1}^{n} \left(\frac{y_i(\vec{p}) - z_i}{\sigma_i}\right)^2$
- To find the 'best' fit, we need to compare different models of the data, models for the error, and different values of the parameters given those models.
    - To find the best parameter values for a given model, we run fminsearch with multiple initial guesses for the parameters, and pick the parameters that give the lowest value of J.
    - To find the best model for the error, we hope to use prior knowledge about the source of the error, or fit an error model to the measurement error if available. We then run the minimization with a given data model, and plot the weighted residuals versus time or versus the data values. The best error model gives the least bias, or correlation between weighted error and data or time.
    - To find the best model for the data, we compare J *if* we used the same error model, and regardless, plot the data versus the model and the weighted residuals versus data or time. The best data model has the lowest J and least bias with the fewest parameters. This is a clear question if two models have the same number of parameters, if the model with more parameters has a similar or higher J or bias, or if the model with fewer parameters has such a high J or bias that it clearly fails to fit the data acceptably. When the model with more parameters provides a moderately better fit, the best of the two models may depend on the purpose.

- An overfit model is one in which there are too many parameters to describe the complexity of the data. Overfit models can be spotted when multiple, different parameter sets provide similar or identical fits. We have very little confidence the in the parameter estimates of overfit models.