COSMOS: Making Robots and Making Robots Intelligent

Lecture 5: Introduction to modeling*

Jorge Cortés and William B. Dunbar

March 7, 2006

Abstract

In this lecture, we define what a model is. We will also see how the notion of a model will help us answer questions about a system. There are various modeling techniques: differential equations, automatas, etc. In this course, we will only use difference equations (remember Lectures 3 and 4!).

Contents

1 Introduction

Given a system of interest to us, for instance, a robot, a car, or a rocket, we need a way to predict its behavior. A ***model*** of the system does precisely that: it provides us with a prediction of how the system will behave under different situations. Models help sort out what is going on with the system.

Of course, models aren't perfect. They try to describe very complex natural and engineered phenomena. Therefore, we should have in mind that a model is always an approximation of the actual

*Part of the exposition in this lecture builds upon Chapter 2 of K. J. Åström and R. M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers.* Preprint, 2005. Available online at http://www.cds.caltech.edu/~murray/amwiki behavior of the real system. This ***uncertainty*** comes from various sources: we usually do not know with total exactness, for example, the values of the parameters of the system (the mass, the friction coefficient, etc.). Another reason is that many dynamic processes are just to difficult to model exactly, and we are often forced to make approximate models.

When approximations are required in modeling, and this is almost always the case, a principle that has the power to save us is called ***robustness***. Robustness is the ability of a system to be *insensitive to measurement, parameter and environmental variations or uncertainties.* Let's consider examples of such variations, in the case of cruise control.

Example 1.1 (Sources of uncertainty in cruise control) Recall the cruise control model

$$v_{k+1} = v_k + \frac{\Delta}{m}[-bv_k + u_{eng} + u_{hill}].$$

An example of measurement uncertainty is if your measurements of v_k are not exact. For example, your speed sensor is actually giving you $v_k + \sigma$, where σ is a noisy signal that contaminates the speed signal. An example of parameter uncertainty is if we don't know the mass m exactly. This is the case as fuel is being burned causing a decrease in mass and we are not taking this into account since it is assumed that m is constant for all time. Lastly, an example of environmental uncertainty is u_{hill}. At best we might be able to assume how large or small this term gets, but in reality, we don't know the exact incline of the road at any instant in time.

Robustness is one of the most useful properties of control. Think again about the cruise controller of your parents' car. You (well, not you, the driver!) want to keep the car going at constant speed, right? The cruise controller automatically adapts its behavior to the slope of the road so that the system is insensitive to climbing uphill or going downhill, and it does so *with no exact knowledge of the true incline of the road traveled!* In short, the cruise controller makes the actual behavior of the car robust to changes in the road incline conditions.

Another perspective is that robustness is what makes it possible to design feedback systems based on strongly simplified models. For us, it is very important to have a good understanding of robustness and to have ways of expressing it quantitatively.

2 Systems modeling

To get a better idea of what a model is, consider the following example.

Example 2.1 (Predator-prey system) The predator prey problem refers to an ecological system in which we have two species (in this case, lynxes versus hares), one of which feeds on the other. This type of system has been studied for decades and it is known to exhibit very interesting dynamics. Figure 1 (right) shows a historical record taken over 90 years in the population of lynxes versus hares[1]. As can been seen from the graph, the annual records of the populations of each species are oscillatory in nature.

[1]Figures taken from "The Connected Curriculum Project" at http://www.math.duke.edu/education/ccp.

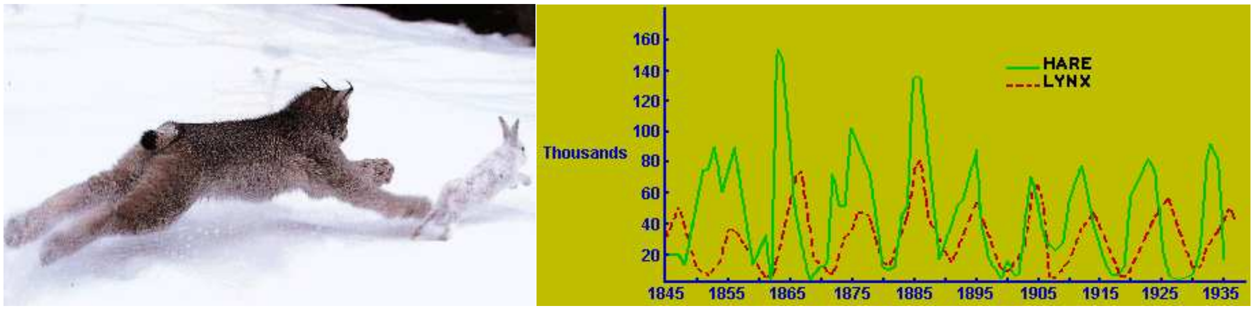

Figure 1: Canadian lynx versus hare

Let us construct a simple model for the evolution of the populations. To do so, we will keep track of the rate of births and deaths of each species.

Let H represent the population of hares and L represent the population of lynxes. Let $k = 0, 1, 2, \ldots$ denote the days at which we look at the populations, starting on January 1, 1845. For instance, H_k is the hare population at day k. Let b_r be the percentage of the current hare population that is expected to be born over a years time, which is assumed to be constant and the same for every year. By this notation, from day k to day $k+1$, $b_r H_k/365$ new hares are born (we divide between 365 because b_r is the annual rate).

On the other hand, let d_f be the percentage of the current lynx population that is expected to perish over a years time, which is also assumed to be constant and the same for every year. Therefore, from day k to day $k+1$, $d_f L_k/365$ lynxes are expected to perish. Finally, let us model the fact that the lynxes are eating the hares with a term which is proportional to the hares and lynxes populations. Let's say that at day k, $aL_kH_k/365$ hares are killed by the lynxes, and that $aL_kH_k/365$ new lynxes are "produced" by eating hares. Here, a is a real number.

Written in the form of a difference equation, the system just looks like

$$H_{k+1} = H_k + \frac{b_r}{D} H_k - \frac{a}{D} L_k H_k, \tag{1a}$$

$$L_{k+1} = L_k - \frac{d_f}{D} L_k + \frac{a}{D} L_k H_k, \tag{1b}$$

where $D = 365$. This simple model makes many simplifying assumptions – such as the fact that hares never die of old age or causes other than being eaten – but it often is sufficient to answer simple questions about the system. To illustrate the usage of this system, we can compute the number of lynxes and hares from some initial population. This is done by starting with (H_0, L_0) and then using equations (1) to compute iteratively the populations in the following day, i.e., generating the orbit of the difference equation (1)! A sample orbit for an specific choice of parameters and initial conditions is shown in Figure 2. It sure has the same form as the plot in Figure 1!

Task 2.2 Compute the equilibrium points of the discrete-time dynamical system (1).

Task 2.3 Write a program in MATLAB® which generates a plot like the one in Figure 2. Ideally, you should define it as a function of the parameters a, b_r, d_f, the initial populations H_0, L_0 and the number of iterations N, so that you can play around with different values of them.

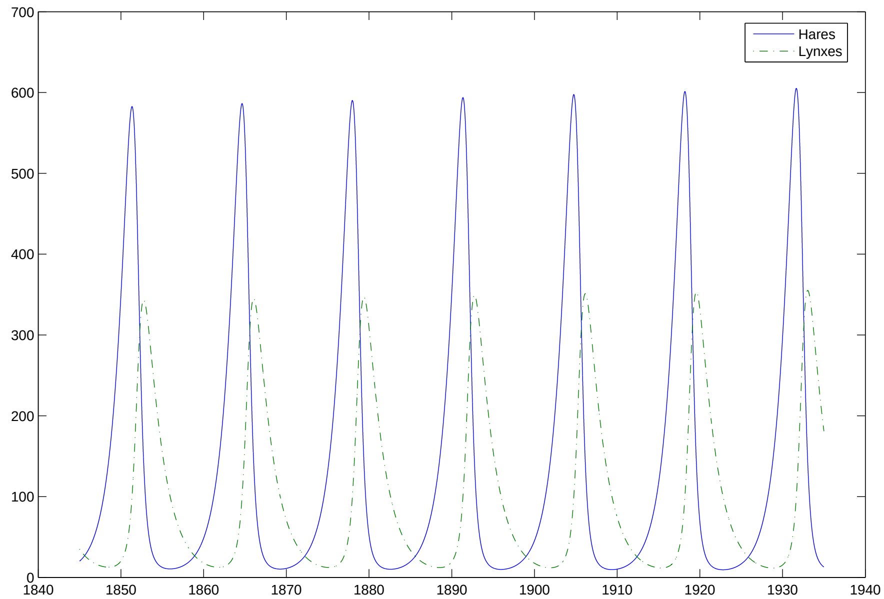

Figure 2: A simulation of the predator-prey model with $a = .007$, $b_r = .7$ and $d_f = .5$. The initial populations are 20 hares and 35 lynxes.

Task 2.4 Use the program that you created in Task 2.3 to determine the stability character of the equilibrium points that you computed in Task 2.2.

We can associate to a single system many different models. The model you use depends on the questions you want to answer. In the previous example, we know that there are various important things that we have not considered, like the fact that hares never die. In control problems, so long as feedback provides robustness, models do not have to be completely exact.

2.1 What do we want models for?

As we said earlier, to predict how the system will behave under different situations. Models will help us to find out what is going on with the system. For instance, in the predator-pray example, questions we might want to answer include

- Given the current population of hares and lynxes, what will it be next year?
- If we hunt down lots of lynxes in a given year, what will the effect on the rabbit and lynx population be?
- How do long term changes in the amount of rabbit food available affect the populations?

We can answer these questions once we have constructed a model for the evolution of the populations. Of course, the answers we get will depend on the model we use. So it is very important to be conscious about the assumptions made when constructing the model. For instance, in our case we have assumed that

- The predator species is totally dependent on the prey species as its only food supply
- The prey species has an external food supply and no threat to its growth other than the specific predator

You can imagine that the answers we get will be very different if a sudden frost freezes the food supply in a given year!

2.2 Terminology about models

The ***states*** describe the state of the system. They capture the effects of the past. They are independent physical quantities.

The ***inputs*** describe the external excitation to the system. The inputs are extrinsic to the system dynamics (externally specified).

The ***dynamics*** describes the state evolution. It consists of an update rule for the system state, and it is a function of the current state and any external inputs. The dynamics might depend on a set of ***parameters***. Different values of the parameters yield different dynamics, all with the same structure.

The ***outputs*** describe sensed or measured quantities. The outputs are a function of state and inputs (therefore, they are not independent variables). The outputs are often a subset of the state.

Consider the following example.

Example 2.5 (Spring mass system) Consider the spring mass system of Figure 3. The ***state***

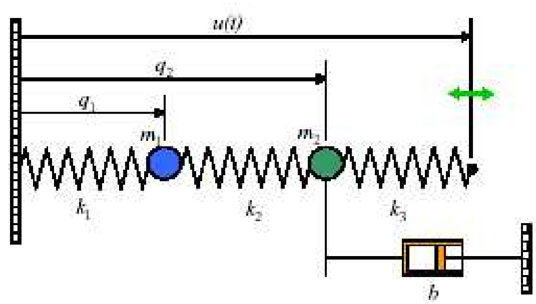

Figure 3: Spring-mass system

of the system are the position q_1, q_2 and velocities $\dot{q}_1$, $\dot{q}_2$ of each mass. The ***input*** is the position of the spring at the right end of the chain, $u(t)$. The ***dynamics*** is determined by Newton's second law (force equals mass times acceleration). The ***parameters*** are the weights m_1, m_2 of the masses and the spring coefficients k_1, k_2, k_3. Finally, the ***outputs*** are the measured positions of the masses q_1, q_2.

Task 2.6 Identify the states, dynamics, parameters, inputs and outputs in the hares and lynxes example above.

3 Robobrain model

Now, we are finally ready to introduce you to a model of the robot that we will be constructing and programming. First we need to define the variables and the parameters that will be in the model. To begin, examine the schematic image of an enlarged `Robobrain` robot sitting the middle of a room, given in Figure 4. The variables that define the position of the robot in the room are

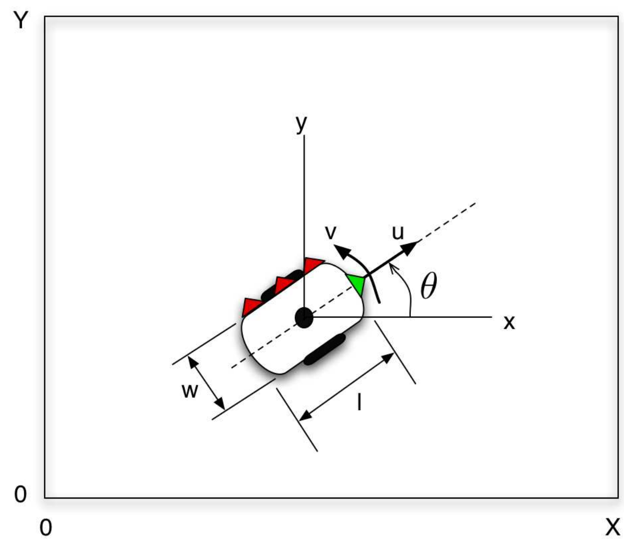

Figure 4: Schematic drawing of an enlarged Robobrain robot operating in a room. The border in the drawing represents the room walls, and the coordinate system shows that the lower left corner is denoted $(x, y) = (0, 0)$ and the upper right corner is denoted $(x, y) = (X, Y)$, with X and Y given by the dimensions of the room.

x and y, using the coordinate frame indicated in the figure. The *orientation* of the robot, that is, the direction that the robot is heading, is denoted by the variable θ, and we define this angle to be measured from the horizontal x-axis to the dashed line that runs through the middle of the robot.

Also indicated in the figure are the forward velocity u and the turning velocity v. These velocities can be computed as a function of the angular velocities of the left and right wheels, denoted ω_L and ω_R, respectively, measured in radians per second (rad/sec). We now proceed with generating these functions.

The forward velocities of the right and left wheels, v_R and v_L, respectively, measured in meters per second (m/sec), are given by

$$v_R = \omega_R r, \quad v_L = \omega_L r,$$

where r is the radius of each wheel in meters (m). Since we can command the independent angular velocities (ω_L, ω_R), we are in effect controlling the independent forward velocities (v_L, v_R).

Now, there is another way of defining the controls in terms of the forward velocities (v_L, v_R). Define the average forward velocity u (m/sec) as

$$u = \frac{1}{2}(v_R + v_L),$$

and the turning velocity v (rad/sec) as

$$v = \frac{2\pi}{l}(v_R - v_L),$$

where l is the width of `Robobrain` (the distance between the wheels). These variables are also shown in Figure 4. The velocity u is in fact the forward speed (m/sec) of the robot, and velocity v is the angular turning speed of the robot (rad/sec).

Substitution of the equations for (v_L, v_R) in terms of the commanded angular velocities (ω_L, ω_R) gives the following equations for u and v

$$u = \frac{r}{2}(\omega_R + \omega_L), \quad v = \frac{2\pi r}{l}(\omega_R - \omega_L).$$

We are now in a position to present the model of the dynamics of `Robobrain`. In other words, as with the cruise control example, we will define an update rule (that is, an iterated map) for the three variables (states) that describe a new location of the robot $(x_{k+1}, y_{k+1}, \theta_{k+1})$, in terms of the current values (x_k, y_k, θ_k), the sample period Δ, and the current values for the controls (u_k, v_k).

For previous discussions, recall that Δ is the sample period and denote the sample times as $t_k = k{*}\Delta$, $k = 0, 1, 2, \ldots$. Also, x_k is the x position of the robot at time t_k, using the same subscript notation for the other variables. The discrete-time dynamic model of the robot dynamics is given by

$$\begin{aligned} x_{k+1} &= x_k + \Delta u_k \cos(\theta_k) \\ y_{k+1} &= y_k + \Delta u_k \sin(\theta_k) \\ \theta_{k+1} &= \theta_k + \Delta v_k \end{aligned}$$

Task 3.1 Identify the states, dynamics, parameters, inputs and outputs in the model of `Robobrain`.

Task 3.2 Can anyone explain the iterated map for the orientation θ? What are the fixed points of this equation?

Task 3.3 Can anyone explain the iterated map for the position x? This is less obvious. The equation says that the next value x_{k+1} is the current value x_k plus a term that is the product of the amount of time between updates Δ, and the component of the forward velocity vector u_k that is in the x-direction, that is, $u_k \cos(\theta_k)$. What are the fixed points of this equation? You can assume that θ_k is equal to its fixed point value computed in the previous task.

The previous tasks brought to light an important point that has not been made explicitly yet in our lectures. If you have a state x_k and a control u_k and some iterated map that described the evolution of x given u, then a fixed point value for x corresponds to a particular choice for u, correct? We will examine this point more in the coming lecture.