ECE 618 - Estimation & System Identification

Lecture 2 - Motivation

A Problem Formulation and Examples

Outline

3. Features of System Identification Problems
   - Auto Regressive Moving Average Models (ARMA)
   - Two example systems (black boxes)
   - Formulation of Sys Id Problems
   - Examples
   - Factors that Affect the Solution

Auto Regressive Moving Average Models (ARMA)

We are interested here in discrete-time linear time invariant systems. In the single-input / single-output case, these are typically written, with input u, & output y, in the z-transform domain as

$$Y(z) = G(z)U(z)$$

where

$$G(z) = \frac{a_0 + a_1 z^{-1} + \cdots + a_n z^{-n}}{1 + \beta_1 z^{-1} + \beta_2 z^{-2} + \cdots + \beta_n z^{-n}}$$

and z^{-1} is the unit delay operator.

In the systems identification literature, these models are called autoregressive moving average (ARMA) models. Using the notation of Ljung's book (p. 41 ff), we have inputs $e(t)$ (where t is discrete time, or integers), outputs $v(t)$, and rational transfer function

$$R(q) = \frac{C(q)}{A(q)}$$

where q^{-1} is the unit delay operator, and

$$C(q) = 1 + c_1 q^{-1} + \cdots + c_{nc} q^{-nc}$$

$$A(q) = 1 + a_1 q^{-1} + \cdots + a_{na} q^{-na}$$

(Note that a_0 from the previous formulation would correspond to c_0 here, and is assumed to be unity.)

In the time domain, the relationship between input $e(t)$ and output $v(t)$ is

$$v(t) + a_1 v(t-1) + \cdots + a_{na} v(t-n_a)$$

$$= e(t) + c_1 e(t-1) + \cdots + c_{nc} e(t-n_c)$$

The first line is the autoregressive (meaning dependence upon delayed values of the same variable) part, and the second line is the moving average.

To connect this with the world of digital filtering, if $n_a = 0$, then

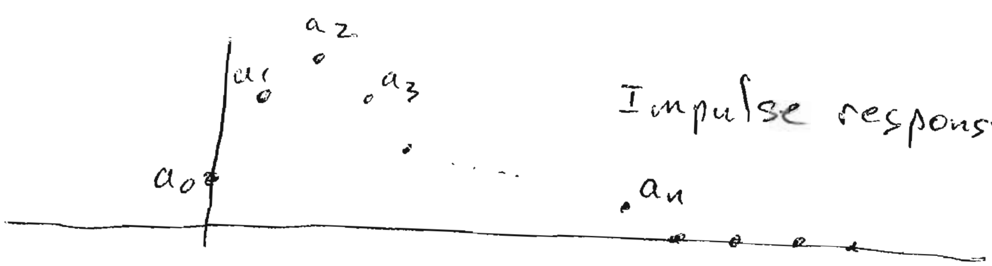
$$v(t) = e(t) + c_1 e(t-1) + \dots + c_{nc} e(t-n_c)$$

or, using notation from discrete-time linear time invariant (LTI) systems,

$$y(t) = a_0 u(t) + a_1 u(t-1) + \dots + a_n u(t-n)$$

These are moving average (MA) models because each output is a weighted average of the input data stream falling within a window to slides forward in time as t advances.

In ~~did~~ digital filtering, this is a finite impulse response, or FIR, filter because the impulse response is simply the sequence of coefficients $\{a_0, a_1, \dots, a_n\}$, which is zero for all time instants after n:



Impulse response of FIR filter.

As an aside, because this observation is not immediately useful but simply builds connections, note that the transfer function of the MA model only has zeros (no poles).

If, instead of n_a, $n_c = 0$, then

$$v(t) + a_1 v(t-1) + \dots + a_{n_a} v(t-n_a) = e(t)$$

or, using the notation from discrete-time LTI systems,

$$y(t) + \beta_1 y(t-1) + \dots + \beta_n y(t-n) = a_0 u(t)$$

This is an autoregressive (AR) model with no moving average component. Its transfer function $G(z)$ has poles, but no zeros.

Actually, we can infer one important fact from examination of the poles and zeros of $G(z)$: Moving average, or FIR, models are always asymptotically stable, while autoregressive models are not. Their stability depends upon the wise choice of the coefficients.

This has an immediate implication for recursive identification: A recursive identification procedure for AR models can be unstable, while a similar procedure for FIR models can only be unstable if it causes the coefficients to "blow up".

The AR model is one example of an infinite impulse response filter (IIR filter): A non-zero input can cause an infinitely long non-zero transient output response. This can not happen with an FIR model: An input that is zero after time k produces an output that is zero after time $k + n_c$. As we shall see in examples, this has implications in system identification.

The autoregressive moving average model (ARMA) is also in general an IIR filter. In the time domain the input/output relationship is

$$v(t) + a_1 v(t-1) + \dots + a_{na} v(t-na)$$
$$= e(t) + c_1 e(t-1) + \dots + c_{nc} e(t-n_c)$$

or, in the notation of discrete-event LTI systems,

$$y(t) + \beta_1 y(t-1) + \dots + \beta_n y(t-n)$$
$$= a_0 u(t) + a_1 u(t-1) + \dots + a_n u(t-n)$$

(Noting again that the first equation assumes $a_0 = 1$, and the second assumes $na = n_c$.

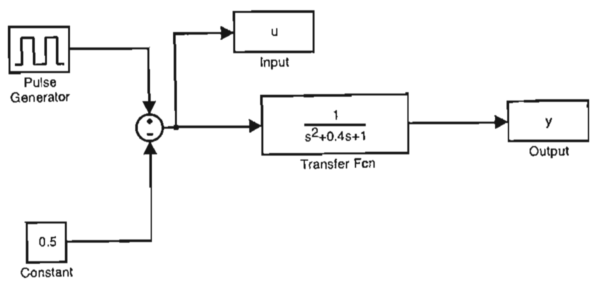
In this lecture, I will illustrate how a MA or ARMA (FIR or IIR) model can be identified from collected input/output data. To do this, I ~~need~~ need a "black box" — or a substitute for a physical system. I used Simulink to do this, and chose two LTI (continuous-time) models — a 2nd order one, and a 6th order one. I set up the Simulink model to excite the system with a signal generator (pulses or sign wave — the pulse generator is shown), and to capture sampled values of the input and output with a sampling period of 0.1 second. The constant in the model removes the bias ~~term~~ term from the pulse generator's output, which is 1 or 0 with a 50% duty cycle.

The transfer function used to generate the second order data is

$$T(s) = \frac{1}{s^2+0.4s+1}$$

which corresponds to a natural frequency of 1 and damping factor of 0.2 (lightly damped).

secondorder



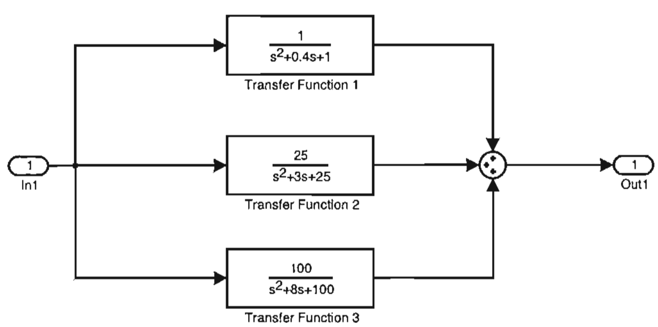
/home/birdwell/classes/ece618f2004/secondorder.mdl

printed 23-Aug-2004 16:26

page 1/1

2nd order model embedded in Simulink system used to capture the data.

six



The 6th order model.

/home/birdwell/classes/ece618f2004/six.mdl

printed 23-Aug-2004 17:11

page 1/1

The 6^{th} order model is a parallel combination of three unity gain 2^{nd} order models with natural frequencies and damping factors given in the table.

Subsystem	ω_n	ξ
1	1	0.2
2	5	0.3
3	10	0.4

Formulation of the System Identification Problem

Let's look at the moving average, or FIR model, first. I will use the notation from discrete-time LTI systems. Rewriting the equation

$$y(t) = a_0 u(t) + a_1 u(t-1) + \cdots + a_n u(t-n)$$

Note that, in the identification problem, we have $\{(u(t), y(t))\}_{t=1}^{N}$, and $\{a_i\}_{i=0}^{n}$ are the unknown system model parameters. (I start at $t=1$ because Matlab begins indices at 1 rather than $\emptyset$.)

We can rewrite the system dynamics as:

$$y(t) = \begin{bmatrix} u(t) & u(t-1) & \cdots & u(t-n) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix}$$

And, this can be done for all values of t for which we have sufficient data (i.e., the delayed values of the input). In matrix notation,

$$\underbrace{\begin{bmatrix} y(n+1) \\ y(n+2) \\ \vdots \\ y(N) \end{bmatrix}}_{\underline{Y}} = \underbrace{\begin{bmatrix} u(n+1) & u(n) & \cdots & u(1) \\ u(n+2) & u(n+1) & \cdots & u(2) \\ \vdots & \vdots & & \vdots \\ u(N) & u(N-1) & \cdots & u(N-n) \end{bmatrix}}_{\underline{U}} \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix}}_{\underline{A}}$$

I can now form a classic least-squares problem: choose the parameter vector $\underline{A}$ to minimize the l_2-norm

$$\| \underline{Y} - \underline{U}\underline{A} \|_2$$

Using $\underline{U}^{\#}$ to denote the pseudo-inverse of $\underline{U}$ (but actually performing the computation using SVD - preferred but slow - or QR),

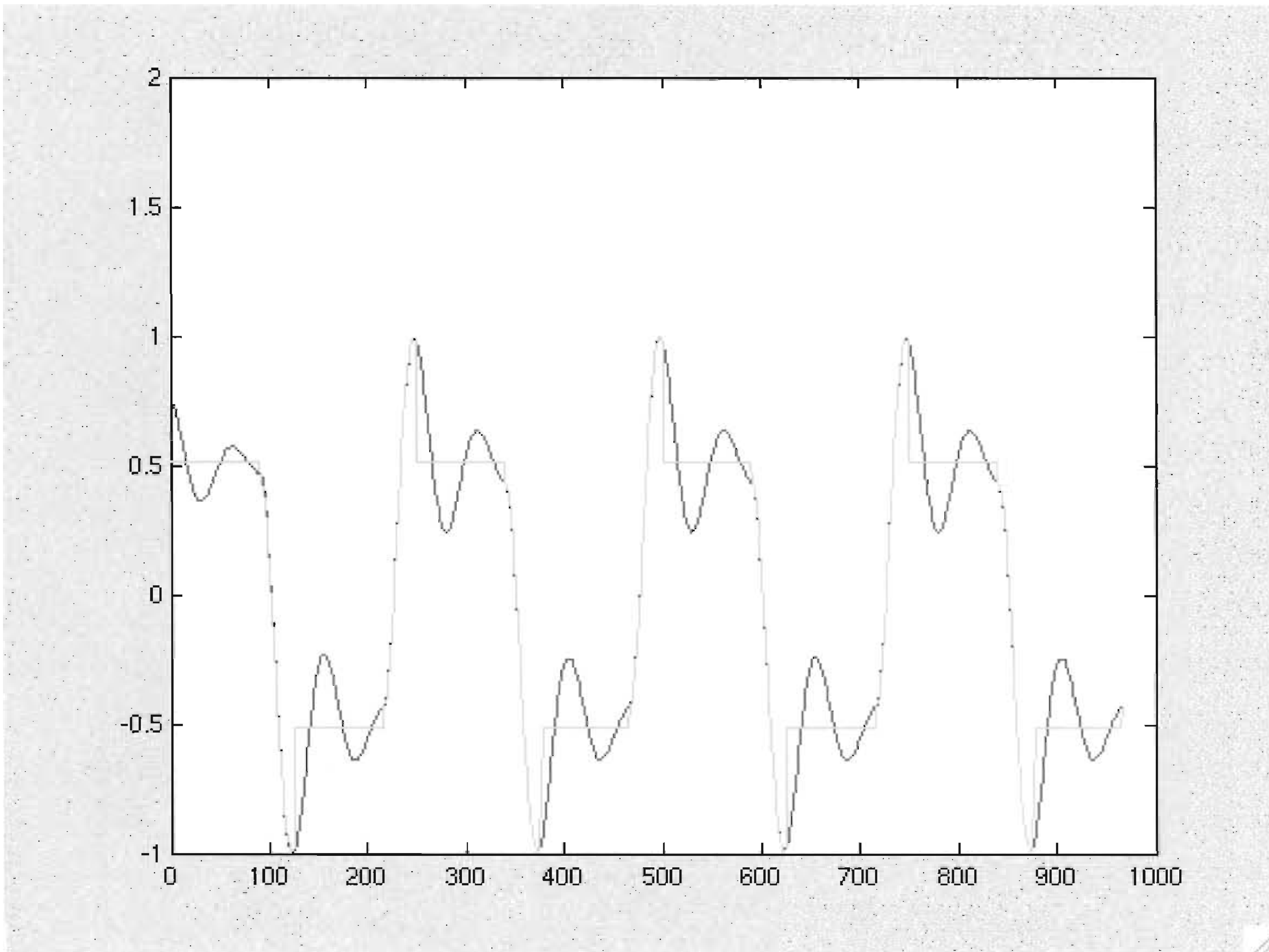
$$\underline{A}^{*} = \underline{U}^{\#}\underline{Y}$$

where $\underline{A}^{*}$ is the vector of least-squares-optimal parameters of the FIR model.

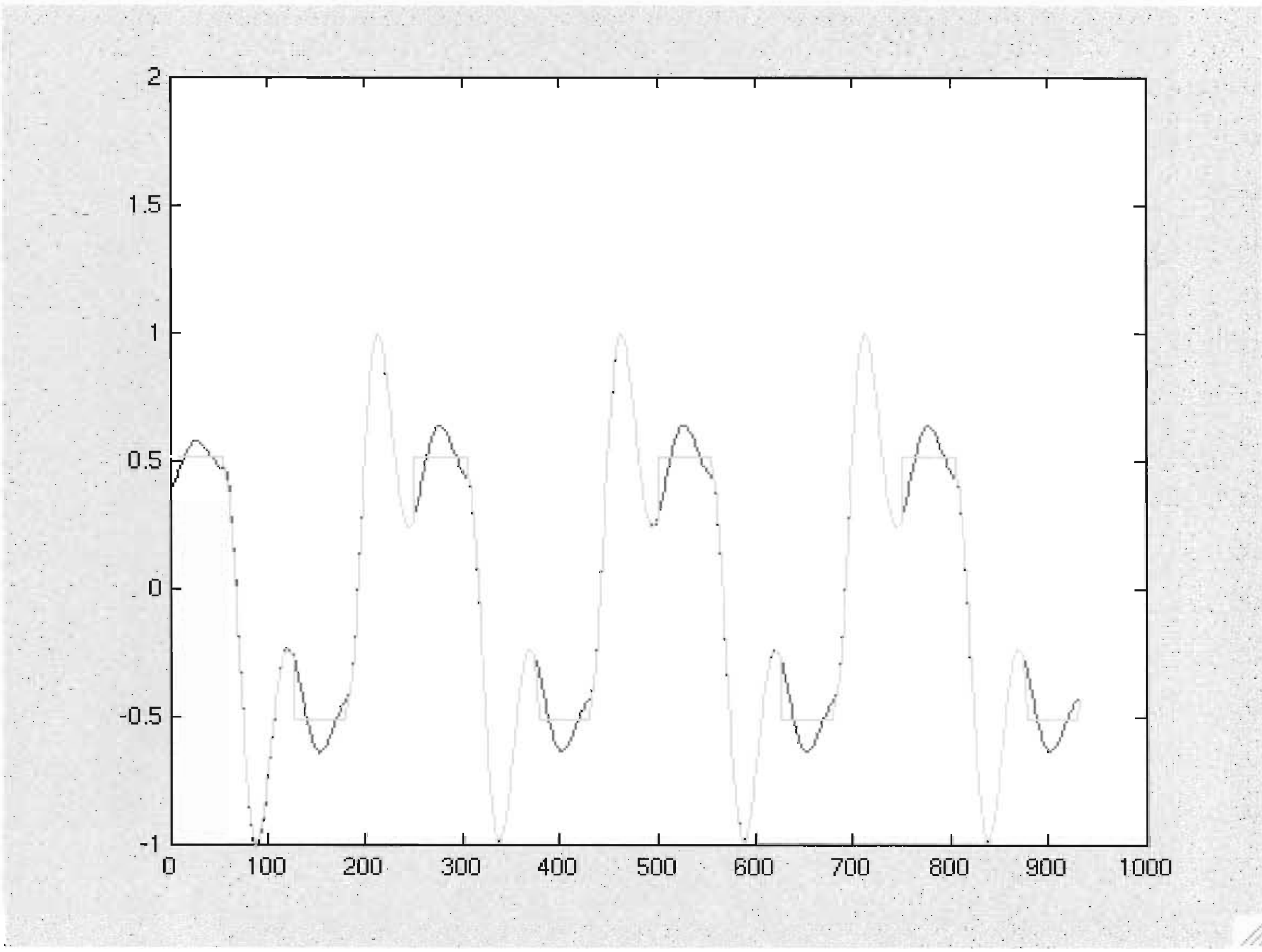
To demonstrate that this works, but also to demonstrate a limitation of FIR models, I wrote a program in Matlab that creates a movie (an AVI file) showing the measured output (in green) and the output predicted by the identified model (in magenta) as a function of n (the number of delay terms used in the moving average model).

As we might have expected, the response of the model does track the measured response, but the response is truncated after n samples.

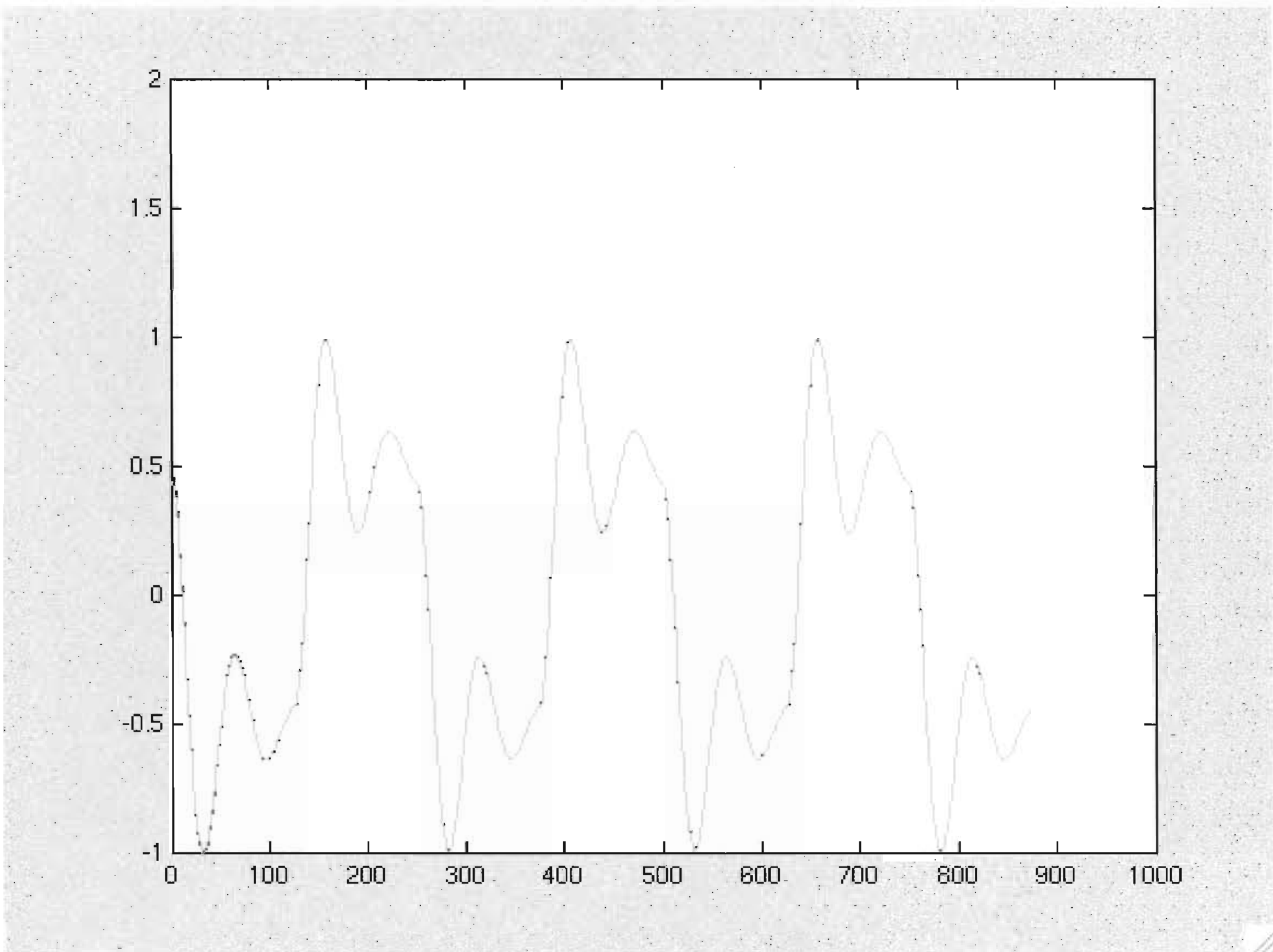
Here are three frames from the movie showing the truncation effect.



Small n



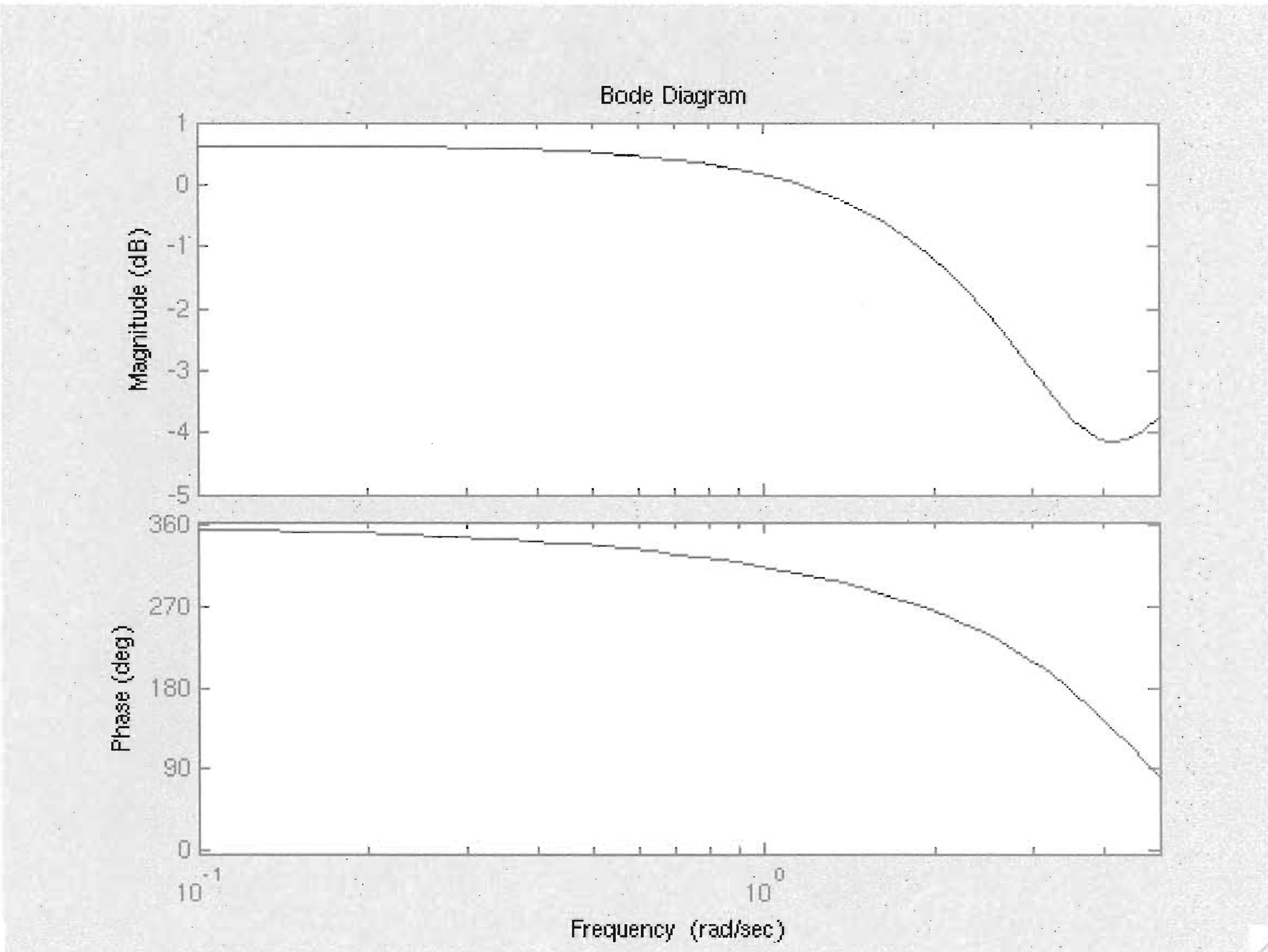
Medsum n



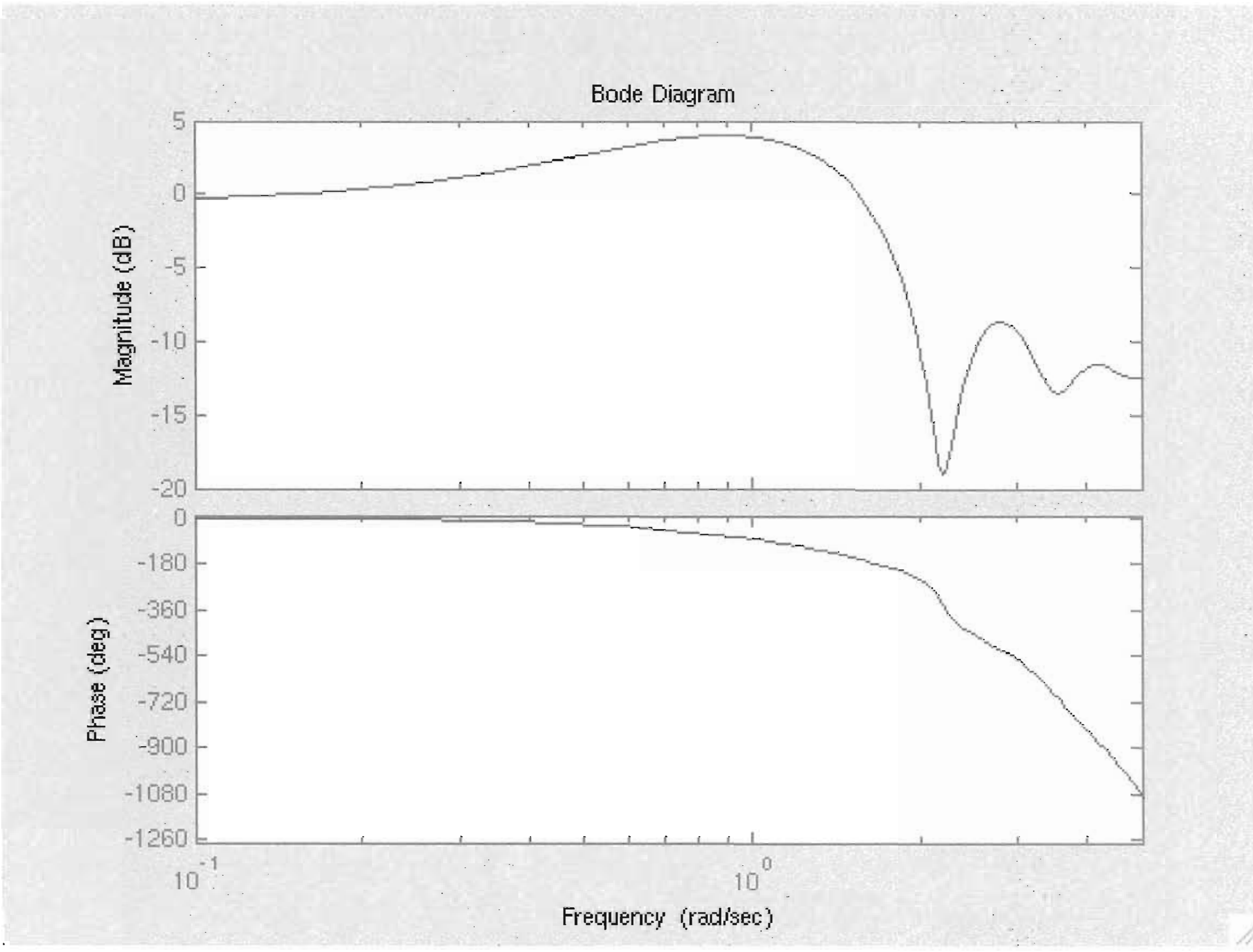
Large n

The truncation effect can create a problem when the sampling time is small relative to the duration of transients (sampling rate is large relative to system bandwidth — or, more explicitly, the frequencies at which interesting things occur.

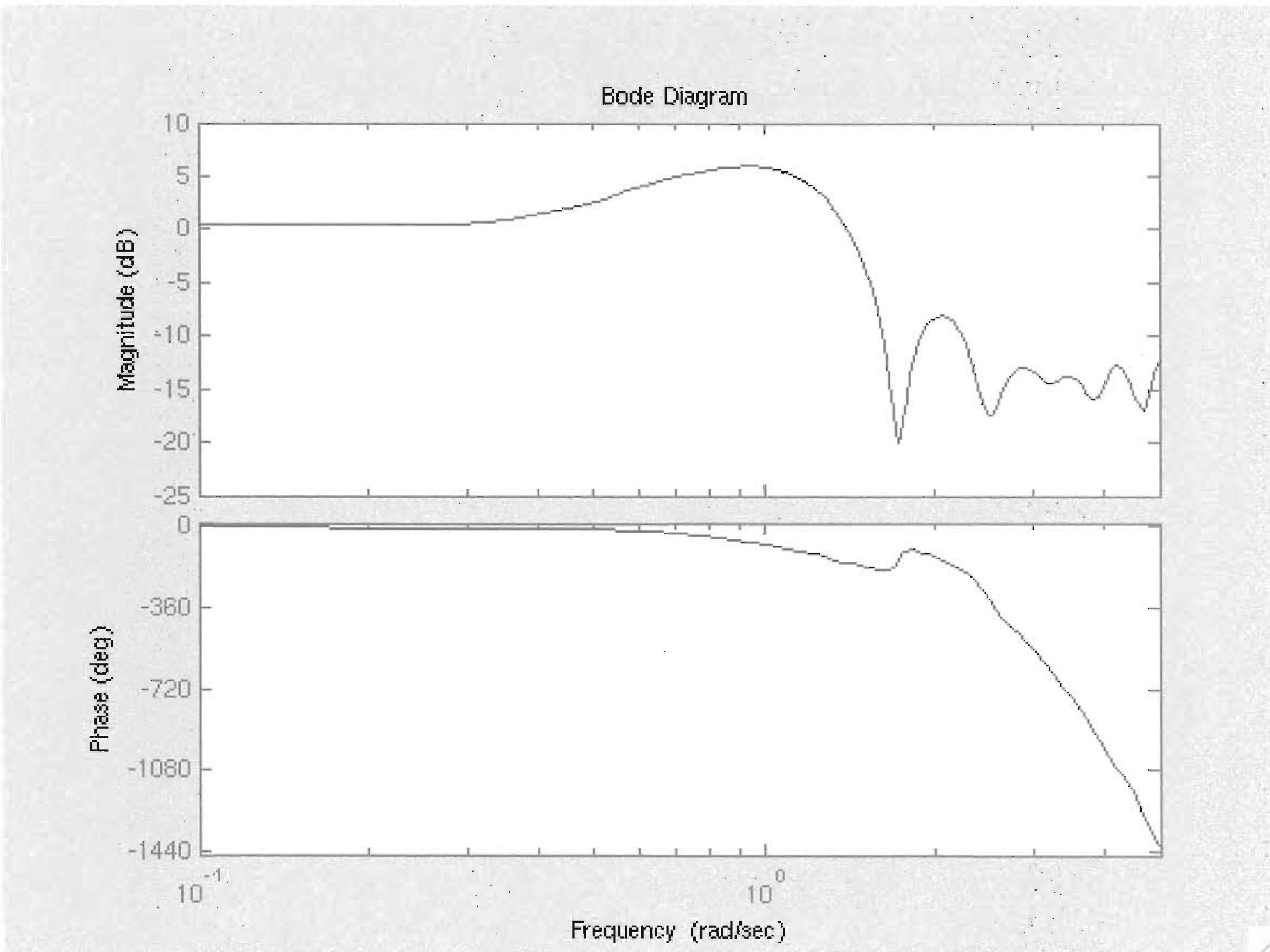
It is also informative to look at what happens to the frequency response — or Bode plot — of the identified model as n increases. We know from the theory of 2nd order LTI systems that the magnitude should be flat at low frequencies, exhibit a resonance peak due to the low damping factor, and decay at -40dB/decade at high frequencies. Plots are shown for 4 values of n from the AVI file created by Matlab, in order of increasing n. Note that the resonance peak does not exist in the first graph, and that a much larger value of n is required before the high frequency portion of the response is reasonably accurate. This is some-what counter-intuitive, and is due to the abrupt truncation of each transient after n samples.



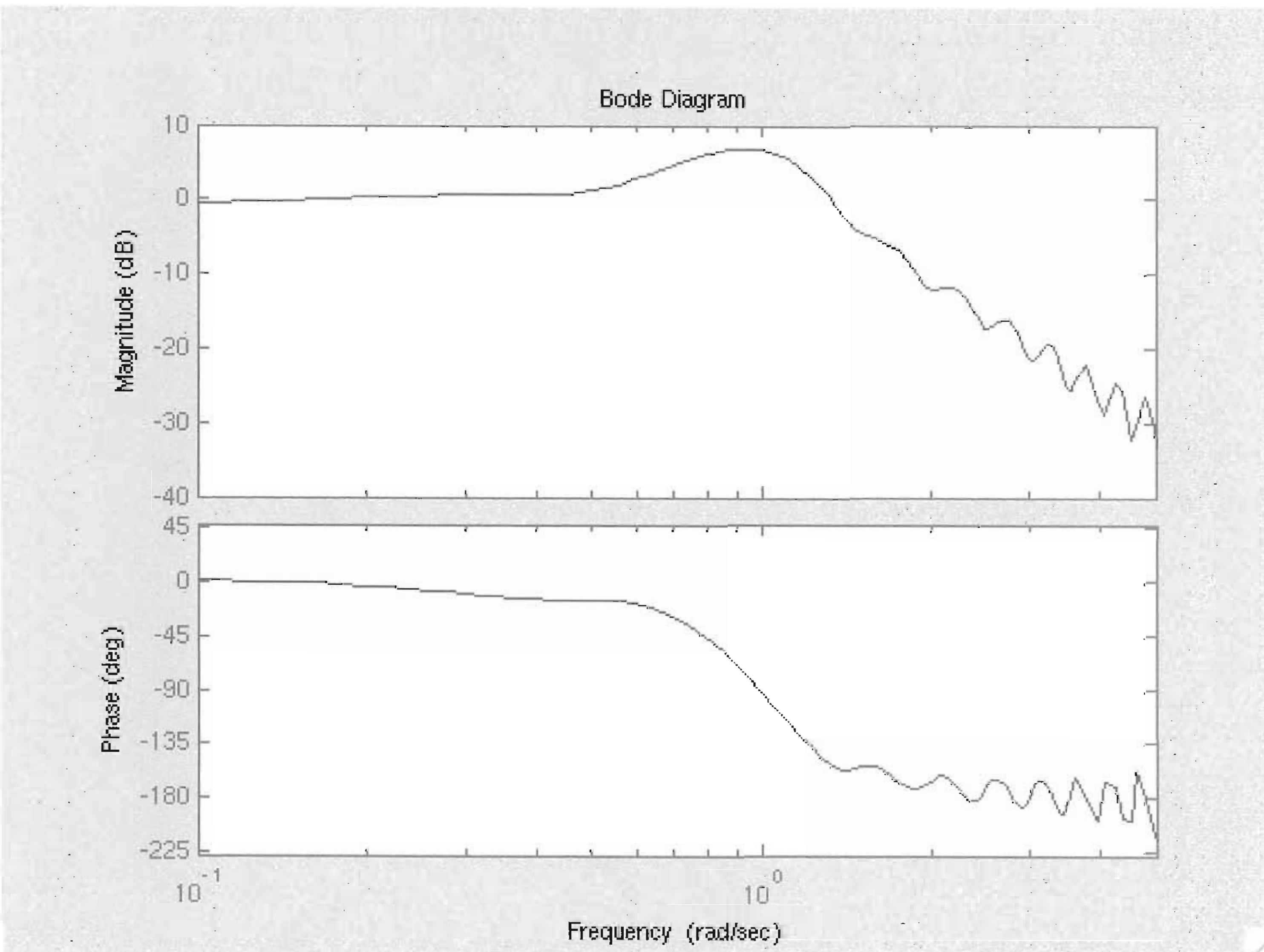
$n = n_1$ (low)



$n = n_2$ (medium-low)



$n = n_3$ (medium - high)



Bode Diagram
Magnitude (dB)
10
0
-10
-20
-30
-40
Phase (deg)
45
0
-45
-90
-135
-180
-225
10^{-1}
10^{0}
Frequency (rad/sec)

$n = n_4$ (high)

Accurately capturing the response characteristic of the 2nd order system required an FIR model with on the order of 100 delay terms (corresponding to 10 seconds), for a system with an $\omega_n = 1$ and $\zeta = 0.2$, or a 2% settling time $T_s = \frac{4}{\zeta \omega_n} = 20$ seconds.

This passed a quick test of reasonableness – the transient captured by the identified FIR model must cover roughly the time required for the measured transient to settle.

Given the number of coefficients that must be identified, and the resulting complexity of an implementation, a reasonable question is whether FIR models are worth the effort. The answer is a resounding Yes! The reasons are somewhat complex, but they boil down to (a) FIR models are very easy to implement, and efficient, and (b) the optimization problems that lead to their identification or design are often convex, which guarantees that a unique optimal solution can be found.

Let's look at a related system identification problem using an ARMA, or IIR, model.

Proceeding as before, the ARMA model equation can be rewritten as

$$y(t) = -\beta_1 y(t-1) - \cdots - \beta_n y(t-n)$$
$$+ a_0 u(t) + a_1 u(t-1) + \cdots + a_n u(t-n)$$

In vector notation, noting that the coefficients are unknown, and the measured inputs and outputs are known,

$$y(t) = \begin{bmatrix} -y(t-1) & \cdots & -y(t-n) & u(t) & \cdots & u(t-n) \end{bmatrix} \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_n \\ a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix}$$

Doing this for all values of t for which we have sufficient data, and writing it in matrix notation,

$$\underbrace{\begin{bmatrix} y(n+1) \\ y(n+2) \\ \vdots \\ y(N) \end{bmatrix}}_{\underline{Y}} = \underbrace{\begin{bmatrix} -y(n) & \cdots & -y(1) & u(n+1) & \cdots & u(1) \\ -y(n+1) & \cdots & -y(2) & u(n+2) & \cdots & u(2) \\ \vdots & & \vdots & \vdots & & \vdots \\ -y(N-1) & \cdots & -y(N-n) & u(N) & \cdots & u(N-n) \end{bmatrix}}_{\underline{\underline{Z}}} \underbrace{\begin{bmatrix} \beta_1 \\ \vdots \\ \beta_n \\ a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix}}_{\underline{c}}$$

Again, we can formulate a classic least-squares problem: Choose the parameter vector $\underline{c}$ to minimize the ℓ_2-norm

$$\| \underline{Y} - \underline{\underline{Z}}\,\underline{c} \|_2$$

The solution is

$$\underline{c}^* = \underline{\underline{Z}}^{\#} \underline{Y}$$

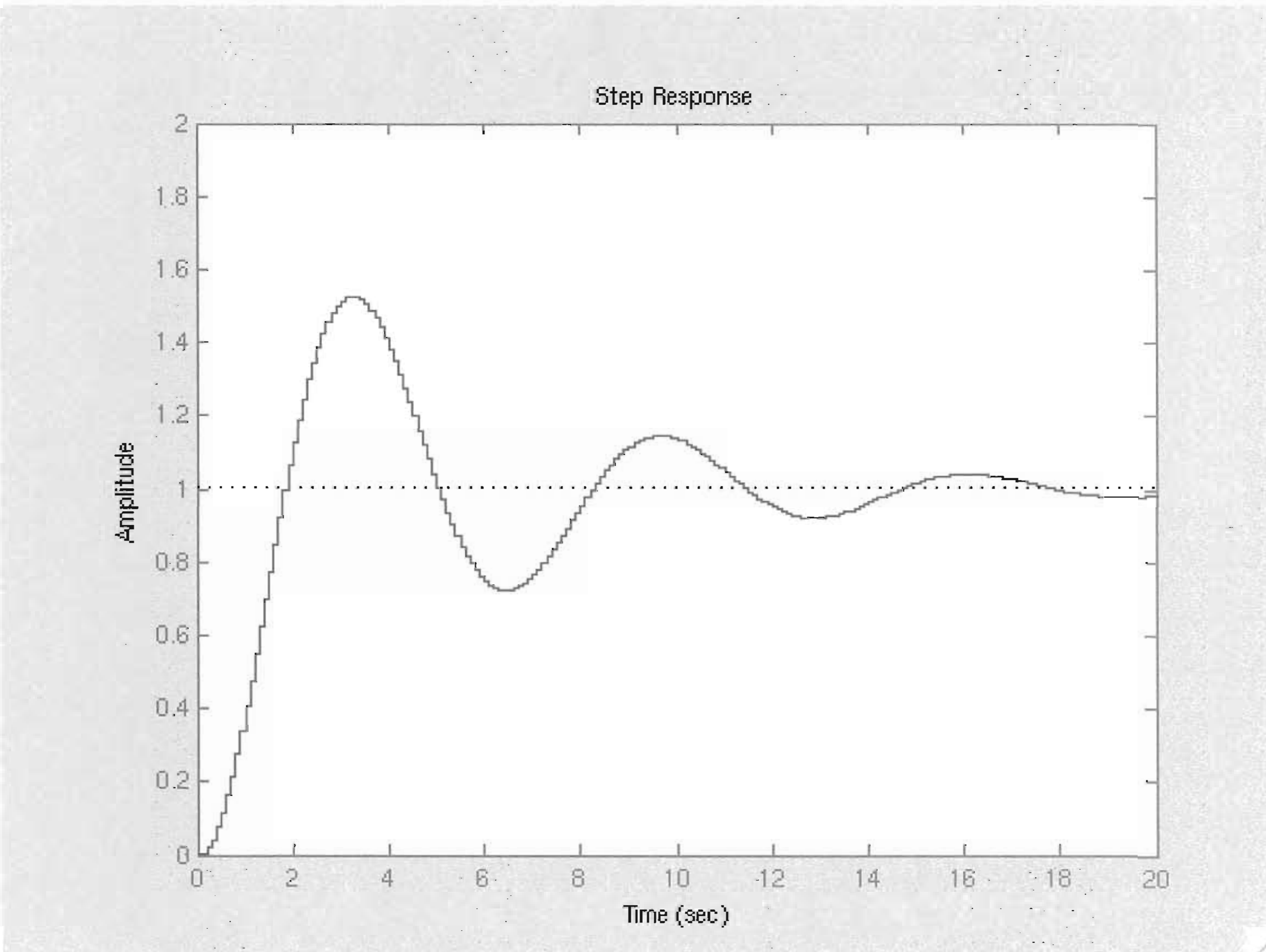
(where, again, SVD or QR is used rather than the pseudo-inverse matrix).

I wrote a similar program for the IIR case, but this time, rather than vary the order of the model (which I fixed at 2), I varied the frequency of the excitation signal. At most frequencies, the identification procedure worked well, as can be seen from the step response (as expected) of the identified model on the next page. However, for some high frequencies, the identified model was not accurate, as shown in the second plot (and as is more visible in the AVI files).

What happened? — The math did not change.

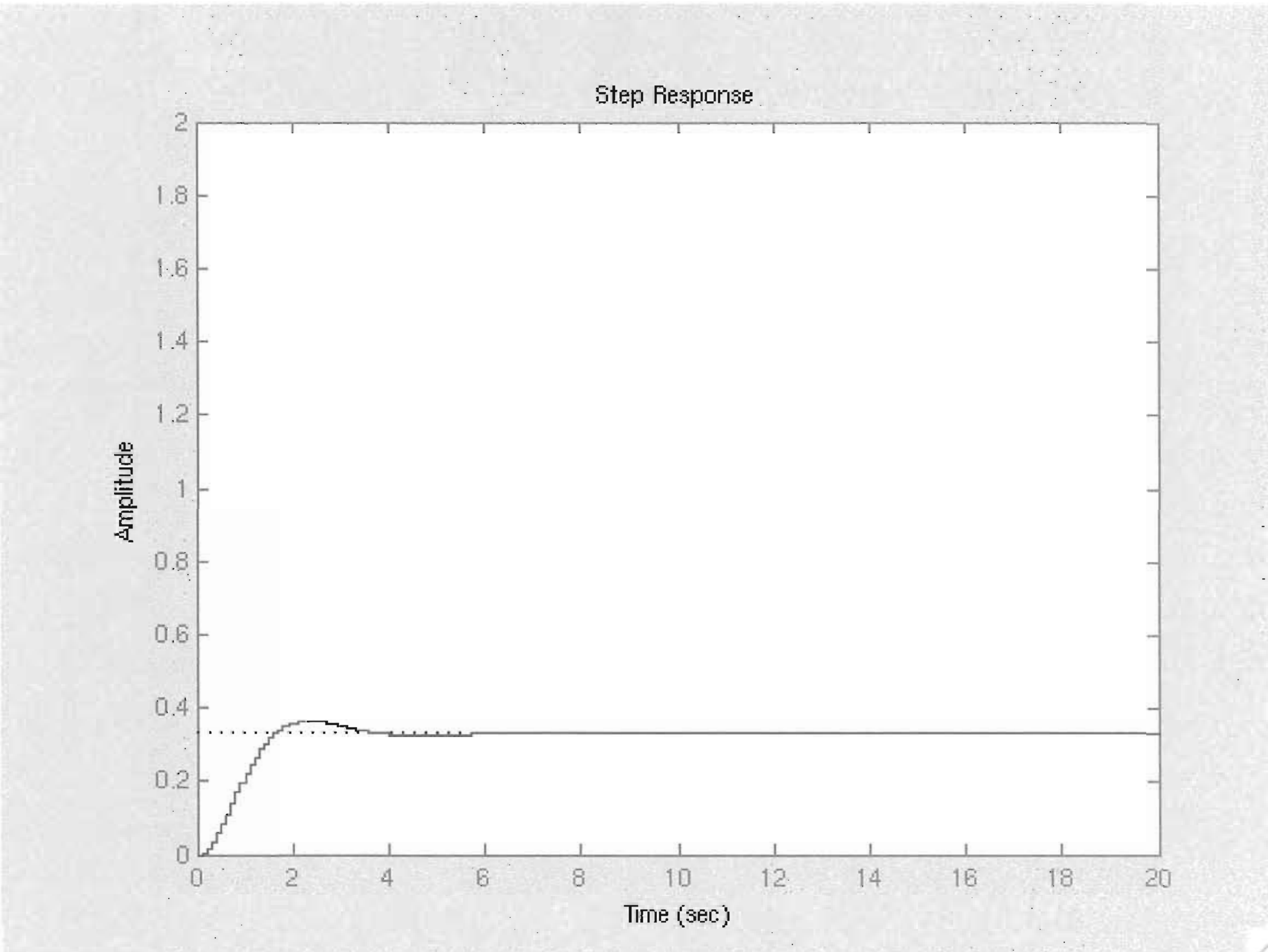
The answer lies in the information available to the identification procedure — it can not identify what it can not see. Worse — sampling effects (remember Shannon) can cause mis-identification.

Step response of identified 2nd order model



Moderate excitation frequency.
Identification works as expected.

Second step response (IIR model)



High excitation frequency.
Identified model is incorrect.

There are (at least) three possible effects that can cause unexpected results:

(1) Lack of information due to poorly chosen excitation waveforms.

(2) Errors caused by sampling (aliasing effects).

(3) Improperly chosen set of candidate models.

Note that in the examples studied so far, the models are of the form $\{M(\underline{\Theta})\}_{\underline{\Theta} \in \Theta}$

In the FIR (MA) case, ~~3~~ n parameters are used (n ~~betw~~ between 10 and 150), and in the IIR (ARMA) case, 5 parameters are used.

To illustrate points (1) and (3), I replaced my 2nd order simulation model with the 6th order model presented earlier.

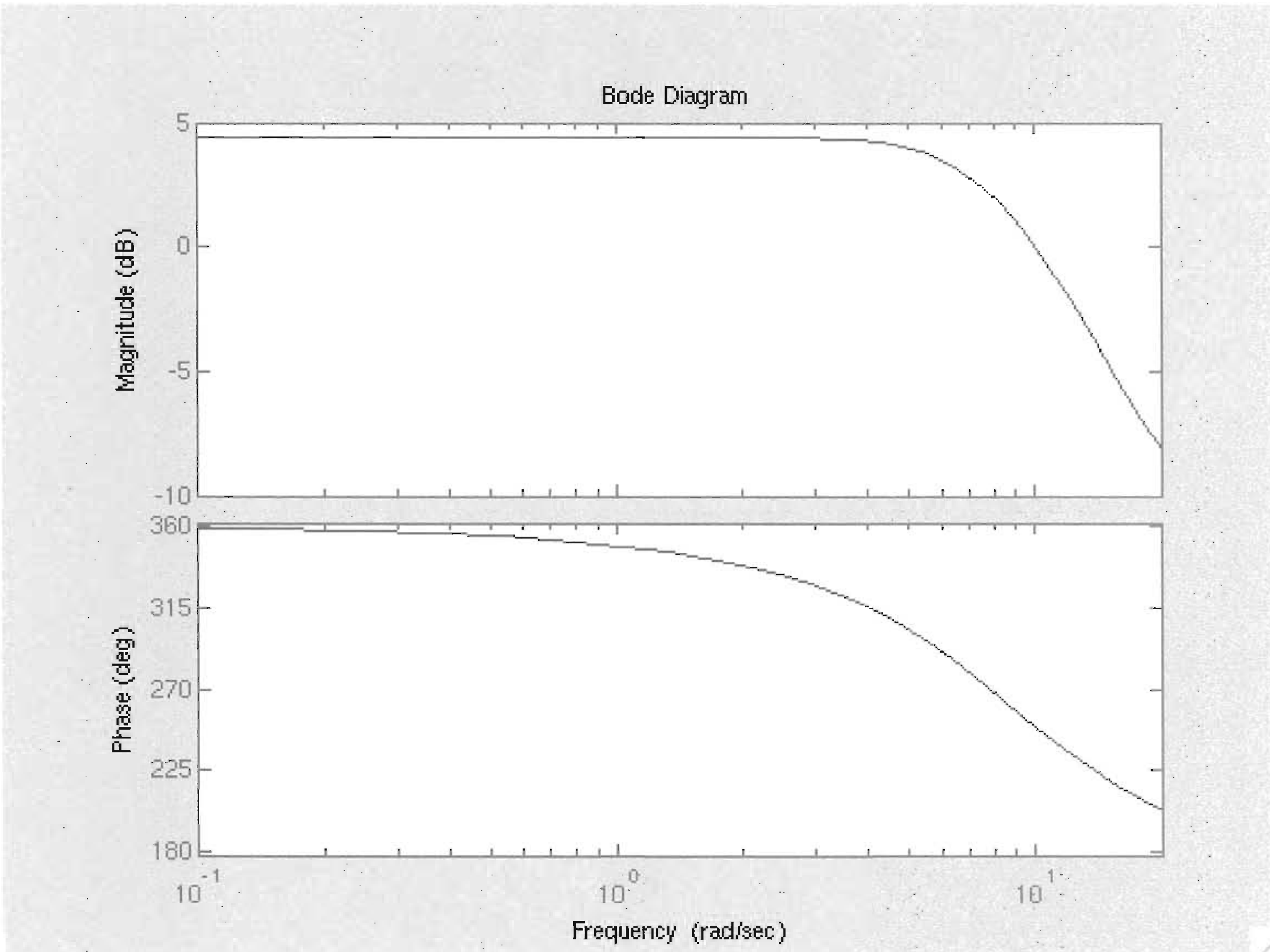
However, I still used a 2nd order model for the identification step.
The results strongly depend upon excitation frequency.

To clearly illustrate the problem, I used a sinusoidal excitation waveform and varied the frequency. This excitation is narrow-band (except for an initial transient), which is not a good idea.

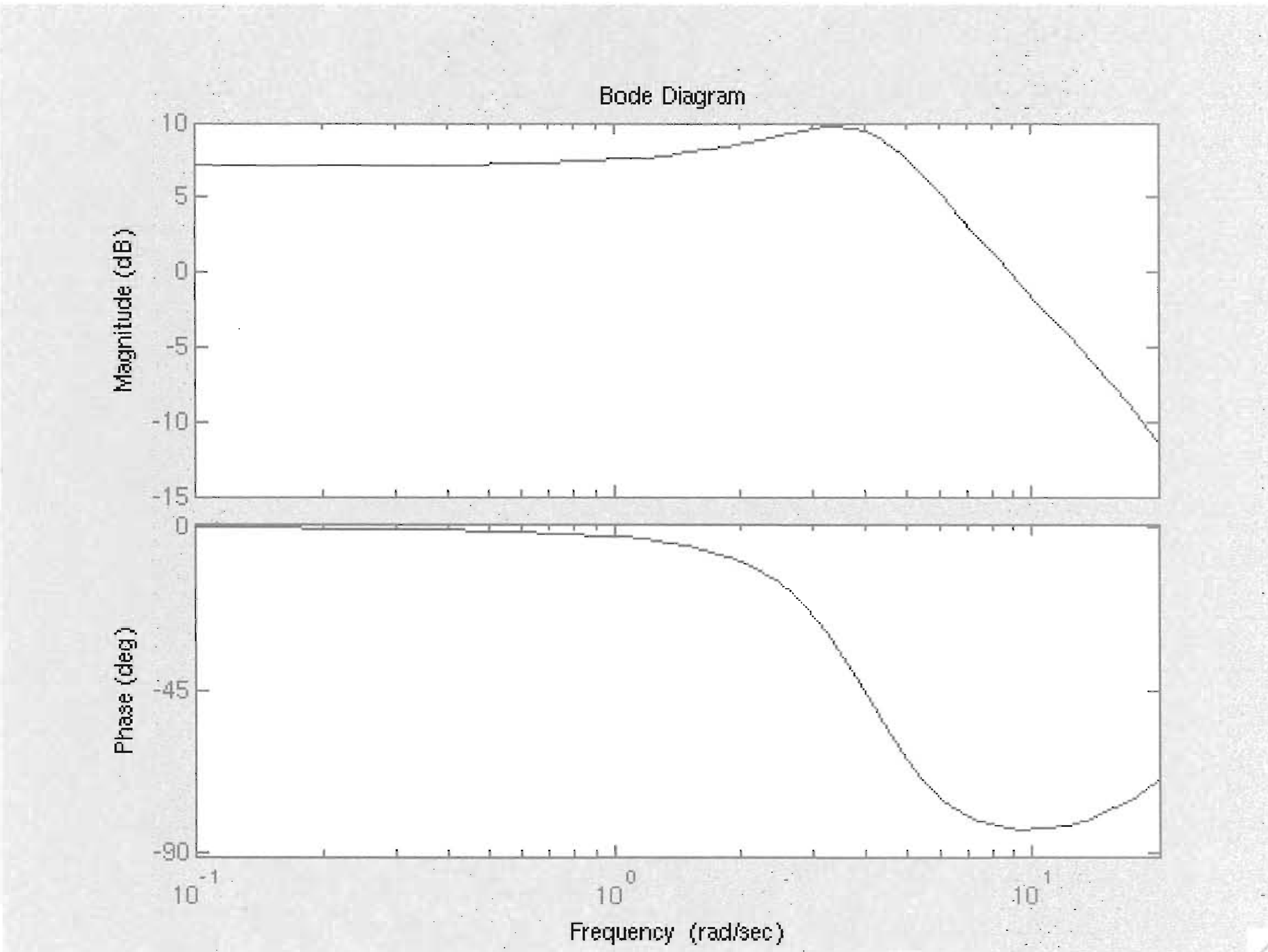
Roughly, the identification procedure identifies each of the 2^{nd} order subsystems in turn as the frequency is changed, starting with the highest frequency subsystem.

Sample Bode plots from the AVI file are shown on the following pages.

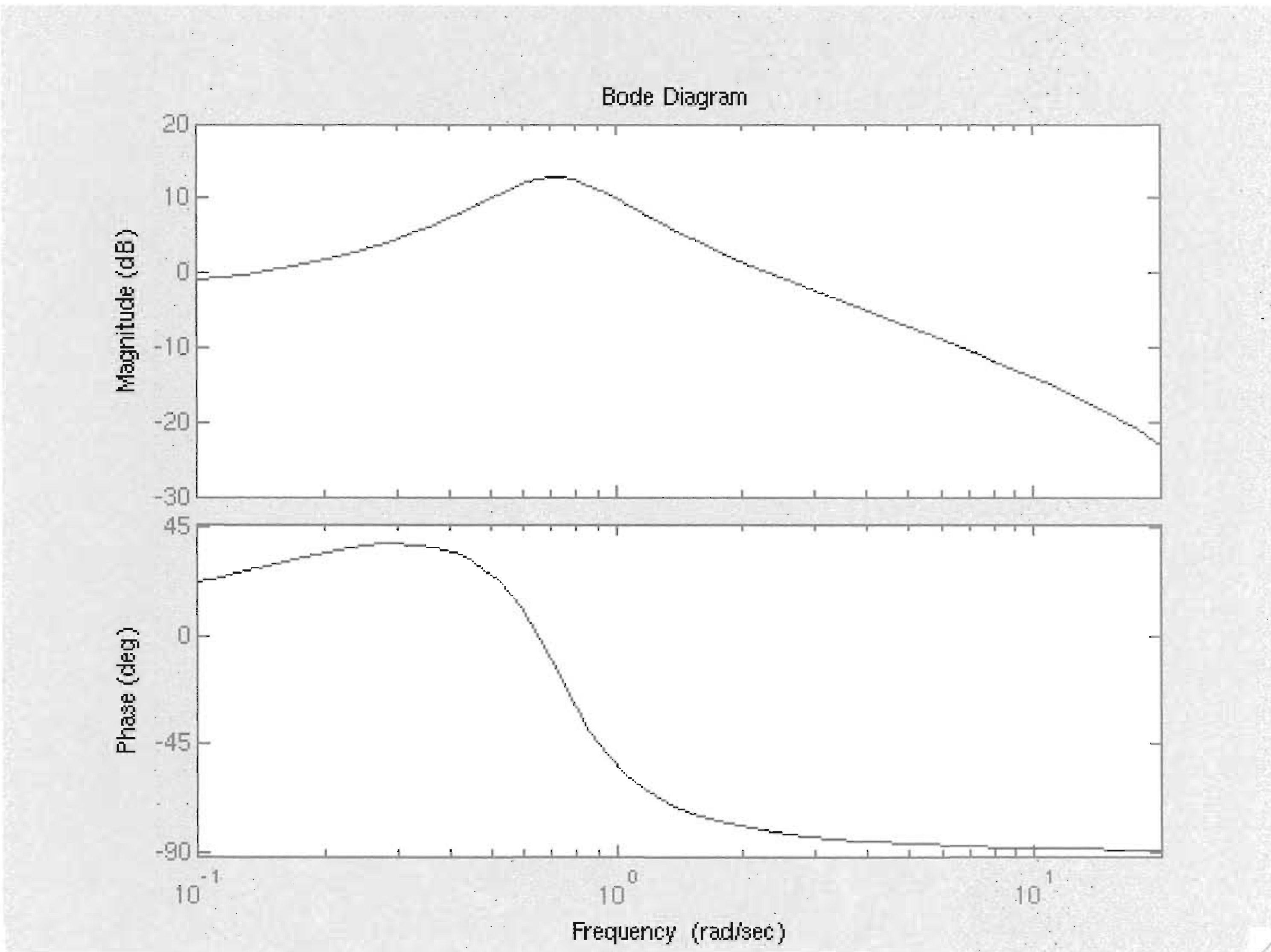
At high excitation frequency, the highest frequency subsystem is identified.



At a moderate frequency of excitation, the medium frequency subsystem is identified.



At a low excitation frequency, the low frequency subsystem is identified.



The matlab code also produced AVI files showing step responses and the ability of the identified models to track the measured responses. In all cases, the models can correctly predict the measured responses — they just can't predict behavior at other excitation frequencies. The step responses also show the characteristics of the identified subsystems — and some interesting "mixing" effects at intermediate excitation frequencies.

Overall, the example illustrates (1) the need for broad-band input signals (or test conditions that cover the full range of expected operating conditions), and (2) the need for careful choice of model families in identification problems.

Here is a quick summary of the concepts we have covered:

(1) Autoregressive (AR), moving average (MA), and autoregressive moving average (ARMA) models.

(2) Differences in notation in different communities.

(3) MA models are finite impulse response filters (FIR models). AR and ARMA in general have infinite impulse responses (IIR models).

(4) A MA models has no poles. An AR model has no zeros.

(5) MA (FIR) models are always asymptotically stable. AR/ARMA models may not be.

(6) Identification of MA, AR, and ARMA models can be posed as least squares problems.

(7) MA (FIR) models may truncate transient responses, and if they do, the high frequency characteristics of the models are probably wrong.

(8) Look at the sampling rate and the settling time of transients to determine appropriate orders for MA / FIR models.

(9) FIR / MA models require lots of coefficients, but have advantages over IIR models (guaranteed stability, ease of implementation, convex identification and design problems).

(10) Properties of physical systems that can not be observed (indirectly) through their effect on input / output behaviors can not be identified. (They can, however, be inferred with "physically-based" models.)

(11) Design experiments to capture all the information needed for identification. (Choice of input signals, sampling rate, other test conditions.)

(12) Choose a model family that has sufficient complexity or "richness" to accurately model the physical system's behavior (but not overly complex!).

(13) Don't expect magic!