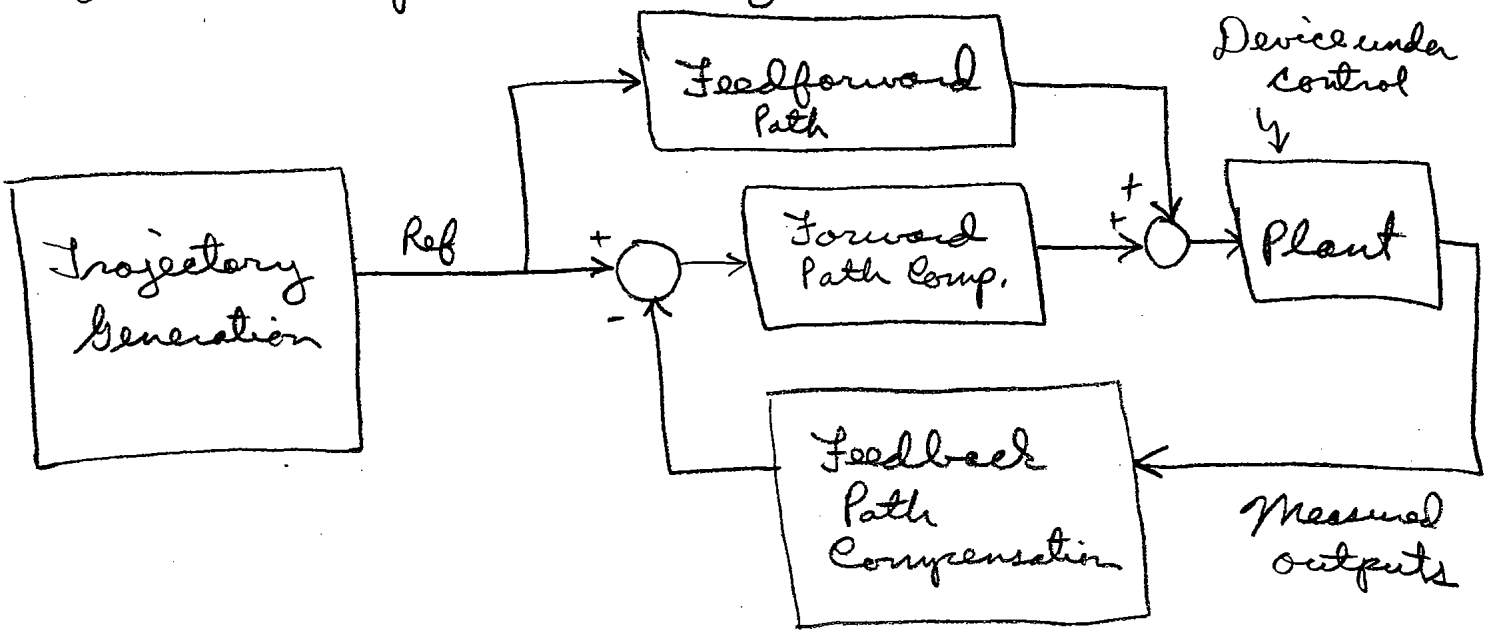
• Control Issues

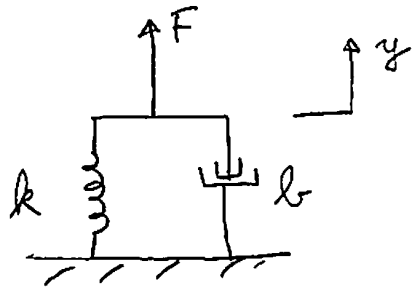
- A thorough understanding of feedback theory is <u>essential</u> for achieving the highest performance from precision machines.
- You need to have a solid understanding of
  - linear system theory
  - Time-domain and frequency-domain concepts and performance metrics
  - PID controllers and tuning
  - Root locus
  - Bode plots; frequency domain stability metrics:
    - phase margin
    - gain margin
  - Nyquist stability criterion

- Computational tools such as Matlab/Simulink are essential
- Dynamic analyzer is very helpful for tuning
- For rapid controller prototyping, consider products such as:
  - dSPACE
  - PMAC (Delta Tau)
  - MEI

- Overview of control system



First-order system example



$$\sum F = m\ddot{y}^{0} a = 0$$

$$\Rightarrow F - ky - b\dot{y} = 0$$

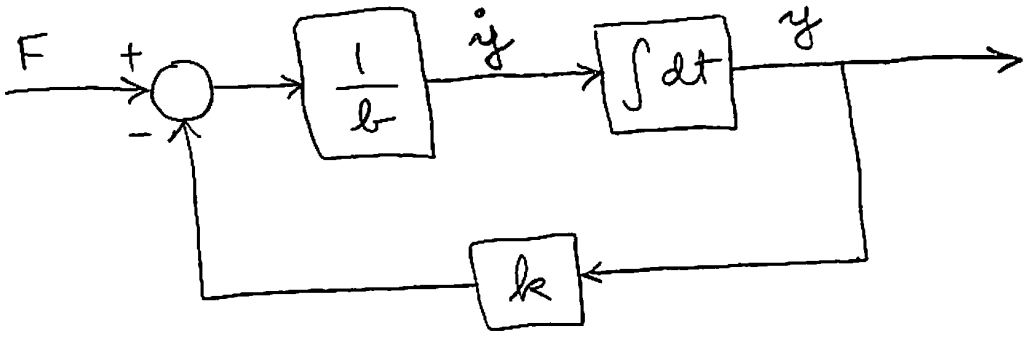
$$\Rightarrow b\dot{y} + ky = F$$

Transfer function:

$$bsY(s) + kY(s) = F(s)$$

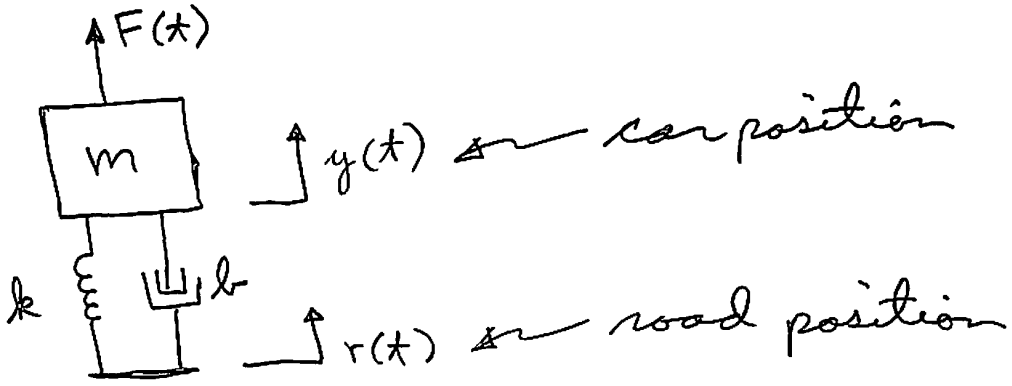
$$\Rightarrow \frac{Y(s)}{F(s)} = \frac{1}{bs+k}$$

Block diagram: $\dot{y} = \frac{1}{b}(F - ky)$



y is "state variable"

Car suspension example

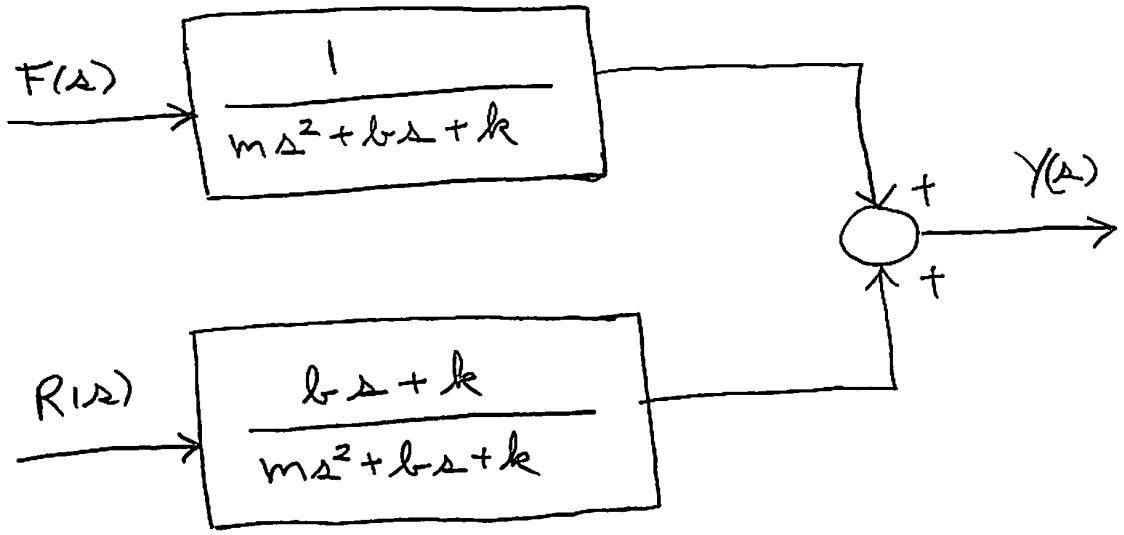


$$m\ddot{y} = \Sigma F = F - k(y-r) - b(\dot{y}-\dot{r})$$

$$\Rightarrow m\ddot{y} + b\dot{y} + ky = \underbrace{F + b\dot{r} + kr}_{\text{inputs}}$$

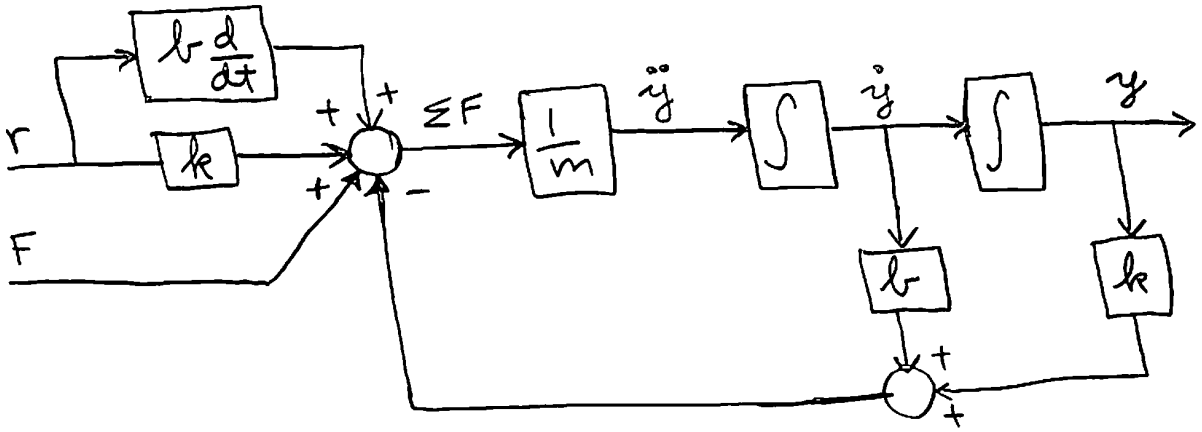
Taking Laplace transform of both sides:

$$(ms^2 + bs + k)\,Y(s) = F(s) + (bs+k)\,R(s)$$

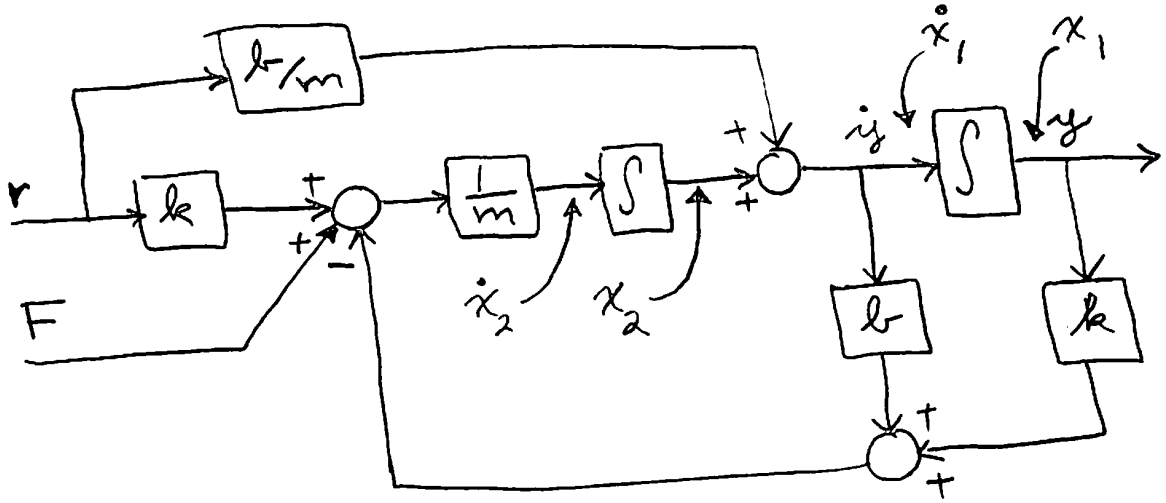


Internal (State) Model:

$$\ddot{y} = \frac{1}{m}\left[F + b\dot{r} + kr - b\dot{y} - ky\right]$$



Rearranging gives:



Define state variables x_1, x_2 as outputs of integrators.

$$\bar{x} \triangleq \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$ state vector

Input vector:

$$\bar{u} = \begin{bmatrix} F \\ r \end{bmatrix}$$

Output:

$$y = x_1 \qquad \text{(a scalar in this case)}$$

Put in standard state variable form

$$\underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}}_{\dot{\bar{x}}} = \underbrace{\begin{bmatrix} 0 & 1 \\ -\frac{b}{m} & -\frac{k}{m} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{\bar{x}} + \underbrace{\begin{bmatrix} 0 & b/m \\ \frac{1}{m} & \frac{k}{m} \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} F \\ r \end{bmatrix}}_{\bar{u}}$$

$$\dot{\bar{x}} = A\bar{x} + B\bar{u}$$

Output is

$$y = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$y = C\bar{x}$$

Standard state-variable form:

$$\dot{\bar{x}} = A\bar{x} + B\bar{u}$$

$$\bar{y} = C\bar{x} + D\bar{u}$$

($D\bar{u}$: feedforward term)

State-space form:

- Compact notation
- Naturally handles multi-input, multi-output (MIMO) case
- Well-suited to machine computation
- Standard for large body of controls literature and algorithms

Especially note: For high-order systems must use state space form for computations. Transfer function representation loses accuracy in more than textbook problems. (See Matlab Control Toolbox tutorial for some examples.)

- Computational tools such as Matlab work most reliably with state-space form.

  e.g. poles: eig(A)

Transfer function from state-space:

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

Take Laplace transforms: (assume zero I.C.)

$$sX(s) = AX(s) + BU(s)$$

$$Y(s) = CX(s) + DU(s)$$

$$X(s) = (sI - A)^{-1} B\, U(s)$$

and so

$$Y(s) = \underbrace{\left[C(sI - A)^{-1}B + D\right]}_{\text{Transfer function}} U(s)$$

(matrix in MIMO case)

How to go from a transfer function to a state-space representation?

matlab command tf2ss provides this function.

For example, if (see F, P, E for more details)

$$H(s) = \frac{b_1 s^{n-1} + b_2 s^{n-2} + \cdots + b_n}{s^n + a_1 s^{n-1} + \cdots + a_n}$$

Then, tf2ss yields

$$A = \begin{bmatrix} -a_1 & -a_2 & \cdots & -a_n \\ 1 & 0 & & 0 \\ 0 & 1 & & \\ \vdots & & & \\ 0 & & 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 1 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} b_1 & b_2 & \cdots & b_n \end{bmatrix}$$

- Control canonical form
- An infinite number of possible state-variable models for a given transfer fcn.

This short note is intended to show you how to convert from a transfer function to a state space representation. For a given transfer function:

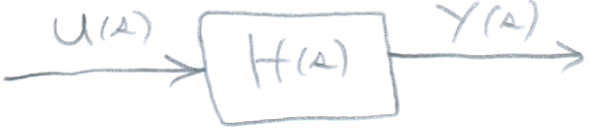
$$H(s) = \frac{Y(s)}{U(s)} = \frac{b_2 s^2 + b_1 s + b_0}{a_2 s^2 + a_1 s + a_0}$$

(This is second-order, but the extension to nth order will be apparent)

Define $b(s) = b_2 s^2 + b_1 s + b_0$

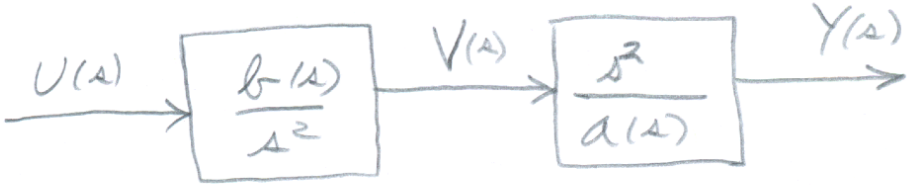
$a(s) = a_2 s^2 + a_1 s + a_0$

With these definitions the block diagram



is equivalent to

is equivalent to

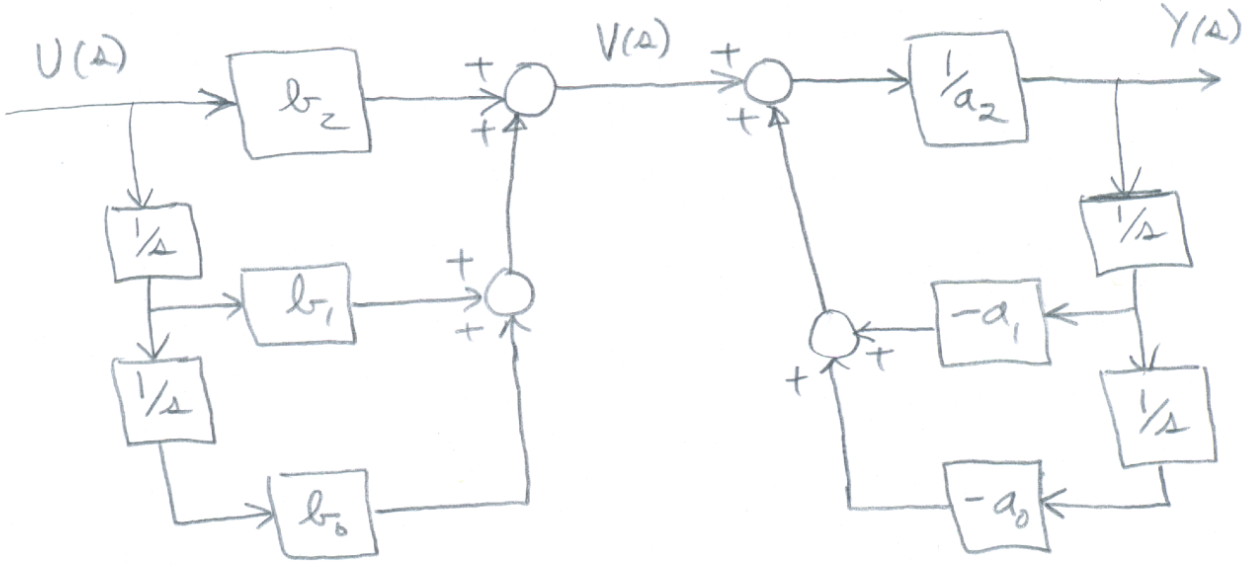


The last step may seem unmotivated, but it will allow us to express the system equations in terms of integration blocks.

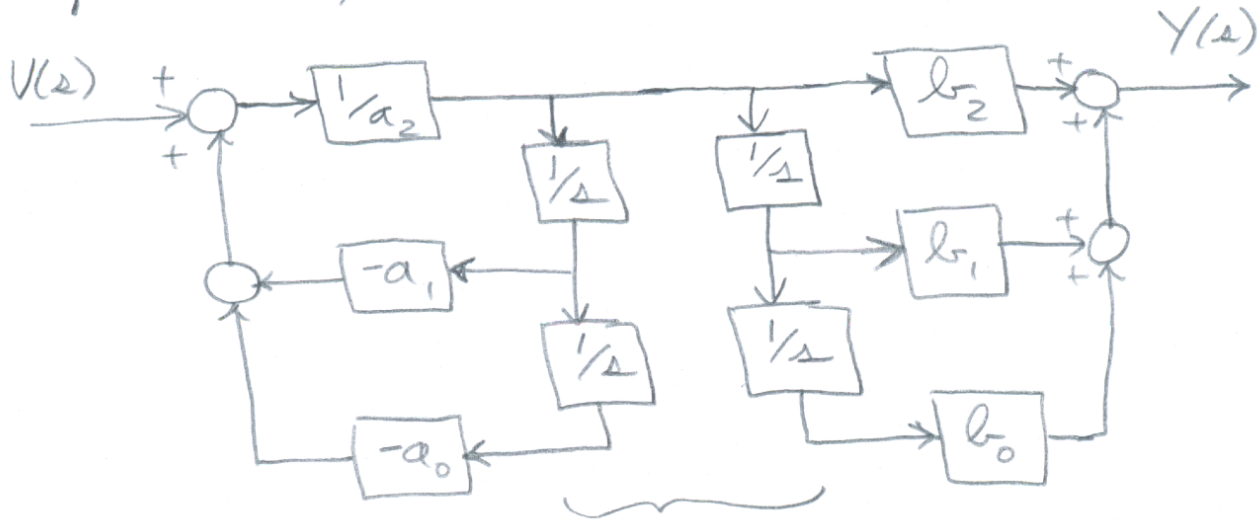
Note that I have also introduced the intermediate variable V(s).

© DL Trumper
3/04

Putting these together gives:



This is a <u>non-minimal</u> realization, since it has 4 integrators, whereas only 2 should be required for a 2nd-order system. To get a minimal realization, we first reverse the order of the blocks (allowable since these are linear)

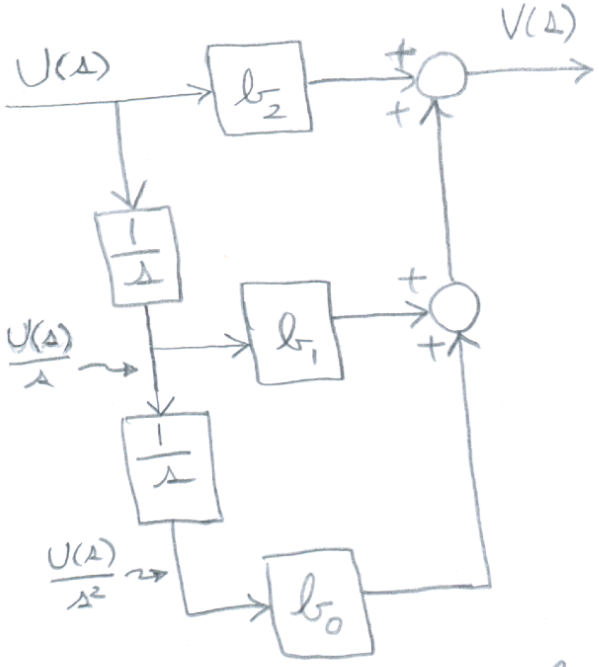


The double chain of integrators is redundant

Working on the left-hand block:

$$V(s) = \frac{b(s)}{s^2} U(s) = \left(b_2 + b_1/s + b_0/s^2\right) U(s)$$

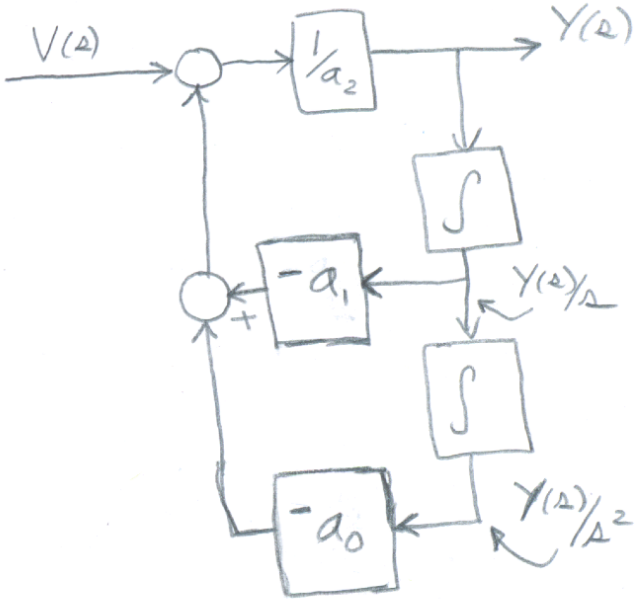
In block diagram form, this looks like:



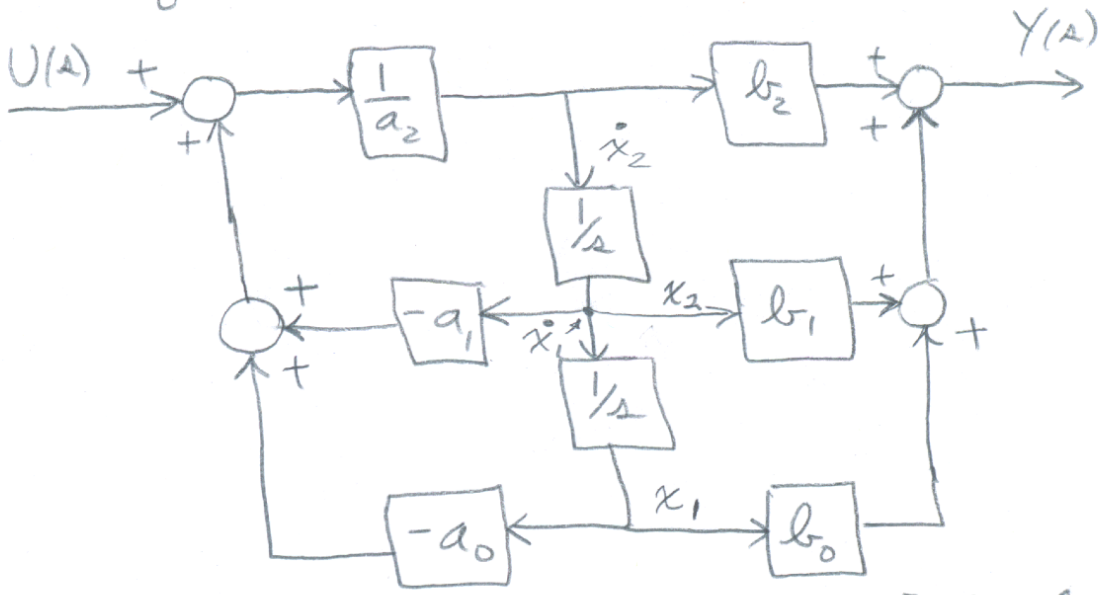
Working on the right-hand block:

$$Y(s) = \frac{s^2}{a(s)} V(s) \Rightarrow a_2 Y(s) = V(s) - a_1/s\, Y(s) - a_0/s^2\, Y(s)$$

In block diagram form:



Eliminating one set of the redundant integrators gives



which is a block diagram for a minimal realization. (There are infinite other realizations, but we just need one...)

Defining the outputs of the integrators as the states x_1 and x_2 as shown gives

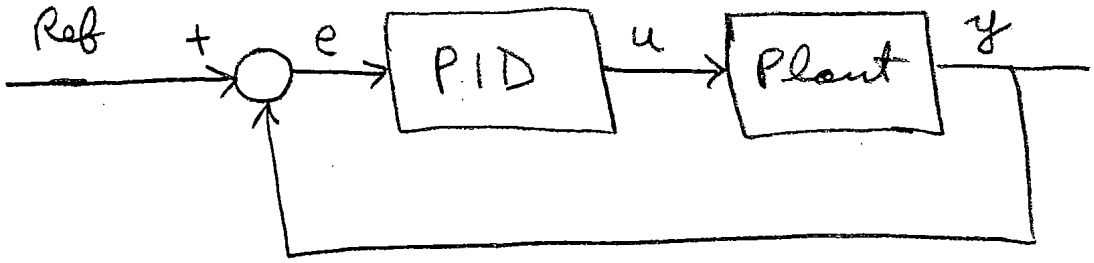
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{a_0}{a_2} & -\frac{a_1}{a_2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{a_2} \end{bmatrix} u$$

$$y = \begin{bmatrix} b_0 & b_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \frac{b_2}{a_2} u$$

(check the derivation yourself.)

This result extends naturally to n^{th} order.

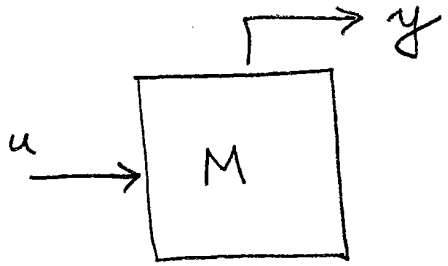
Simple PID control loop



$$u(t) = K\left(e + \frac{1}{T_i}\int e\,dt + T_d\frac{de}{dt}\right)$$

Note: In a real application, need to limit how large this can get → Anti-Windup

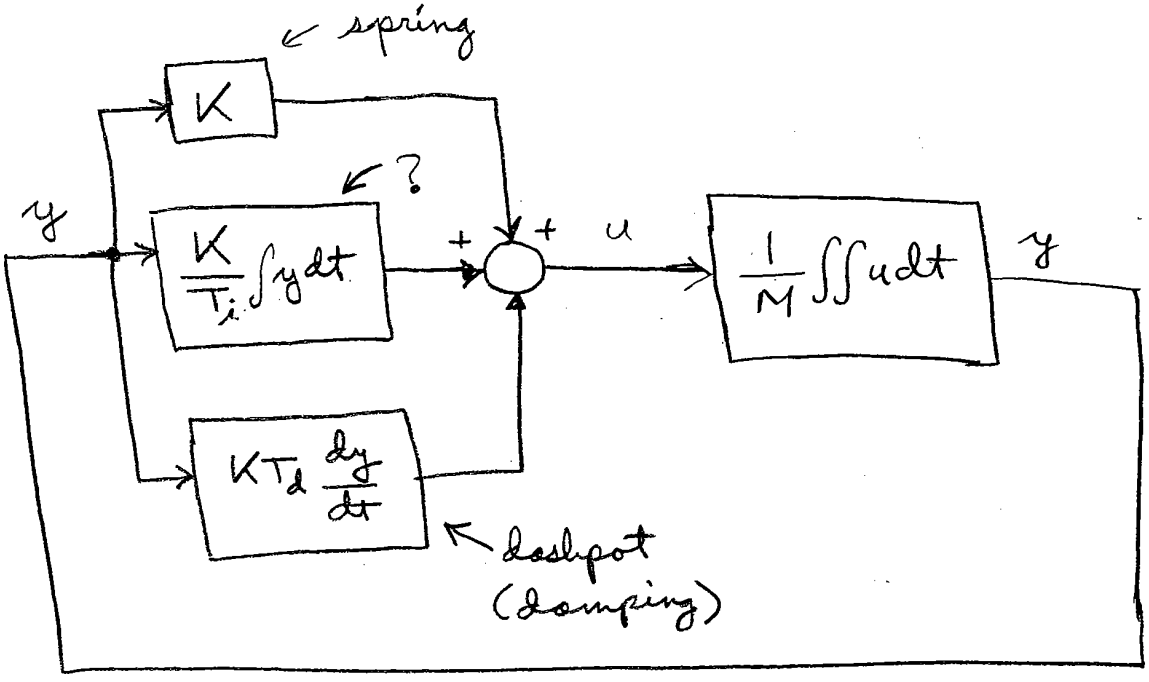
- If plant is a free mass, then control effort u is force, and plant output y is position:



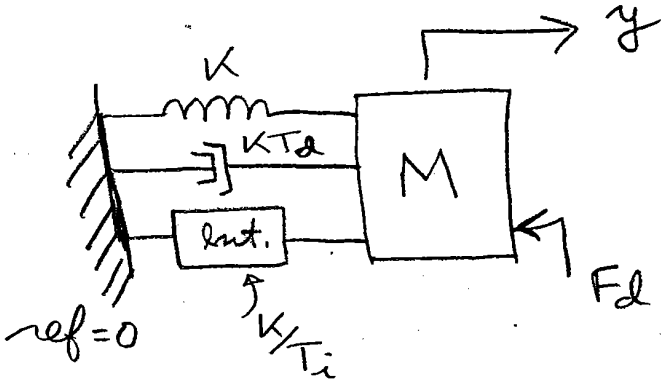
$$F = Ma$$

$$\Rightarrow u = M\ddot{y}$$

- Then the control actions can be interpreted as (let ref = 0):



- Mechanical equivalent:



- the integrator term picks up any long-term loads, such as Fd
- The P and D terms determine the high-frequency stiffness and damping

$$\omega_n = \sqrt{\frac{K}{M}}$$

$$\zeta = \frac{b}{2\sqrt{km}} = \frac{KTd}{2\sqrt{KM}}$$

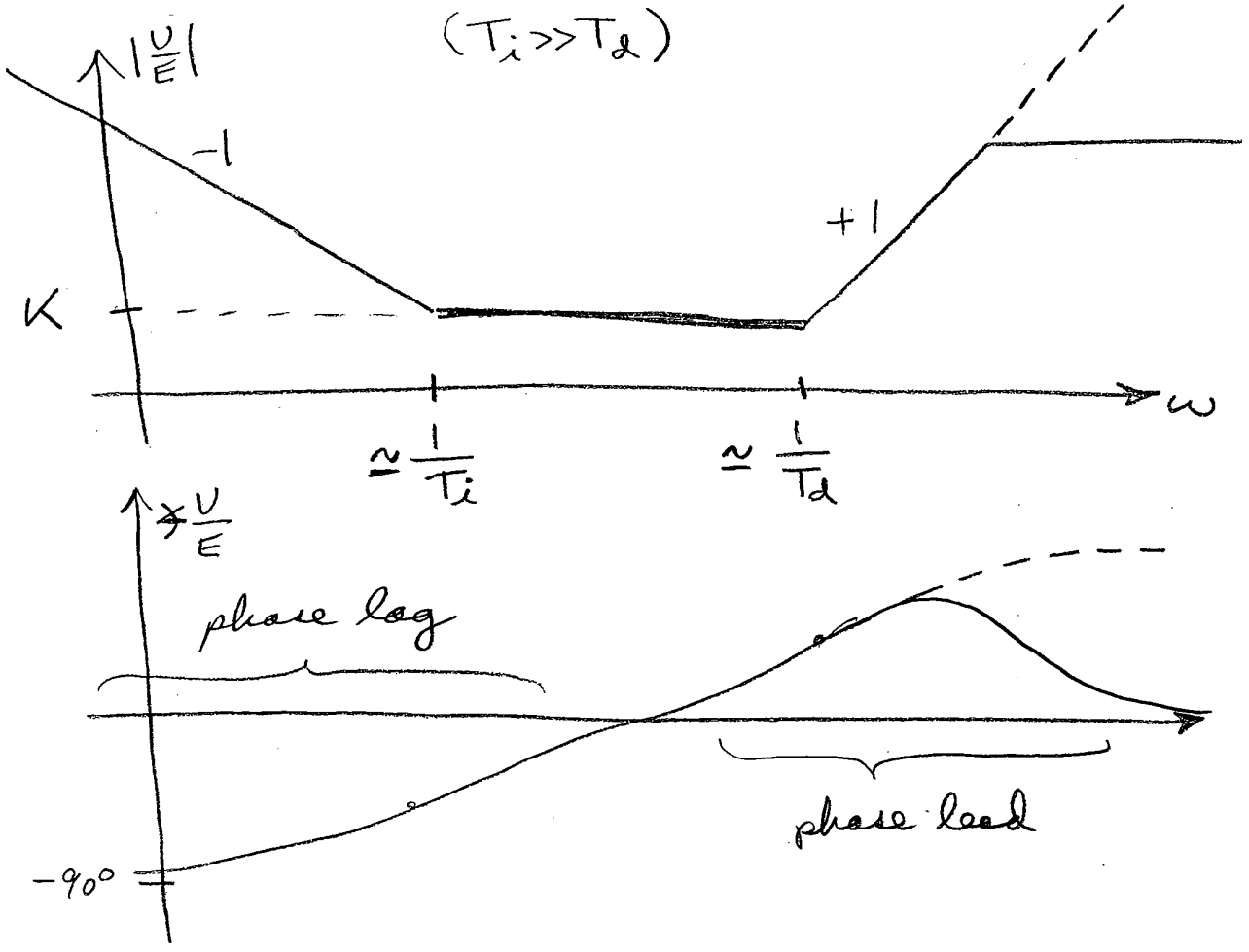
- you can think of this model when tuning a PID controller

- In the frequency domain, the PID transfer function is

$$\frac{U(s)}{E(s)} = K\left(1 + \frac{1}{T_i s} + T_d s\right)$$

but, you can't really implement a pure derivative term, so we use

$$\frac{U(s)}{E(s)} = K\left(1 + \frac{1}{T_i s} + \frac{T_d s}{0.1 T_d s + 1}\right) \qquad (T_i >> T_d)$$



- It is difficult to tune 3 independent controller parameters. The following procedure is helpful to simplify tuning:

1) Determine the desired system closed-loop bandwidth ω_c in rad/sec
Useful relationship:

$$t_r = \frac{2.2}{\omega_b}$$

$\omega_b \triangleq$ 3dB BW

$t_r \triangleq$ 10%-90% rise time

(for most positioning systems, the closed-loop BW is in the range of 10 - 100 Hz (60 - 600 rad/sec)

$(\omega = 2\pi f)$

2) Set $\frac{1}{T_d} = \frac{1}{2}\omega_c$

$\frac{1}{T_i} = \frac{1}{10}\omega_c$ (in preliminary work, you can even turn the integrator off)

3) Now play w/ K_1, monitoring step response to see if rise time and stability are in the desired range.

(also, watch out for sign of feedback loop)

Example:

- Let $M = 10\,kg$
- choose $\omega_c = 500\ rad/sec$ $(80\ Hz)$
- then

$$\frac{1}{T_d} = \frac{\omega_c}{2} = 250\ rad/sec$$

$$\Rightarrow \boxed{T_d = 4\,msec}$$

$$\frac{1}{T_i} = \frac{\omega_c}{10} = 50\ rad/sec$$

$$\Rightarrow \boxed{T_i = 20\,msec}$$

- all that remains is to choose K

Theory:

$$\omega_n = \sqrt{\frac{K}{M}} = \omega_c$$

$$\Rightarrow \boxed{K = M\omega_c^2 = 10 \cdot 500^2 = 2.5 \times 10^6\ N/m}$$

(Note that stiffness and bandwidth are tied together via load mass!)

- However, it is rare to have such a simple plant

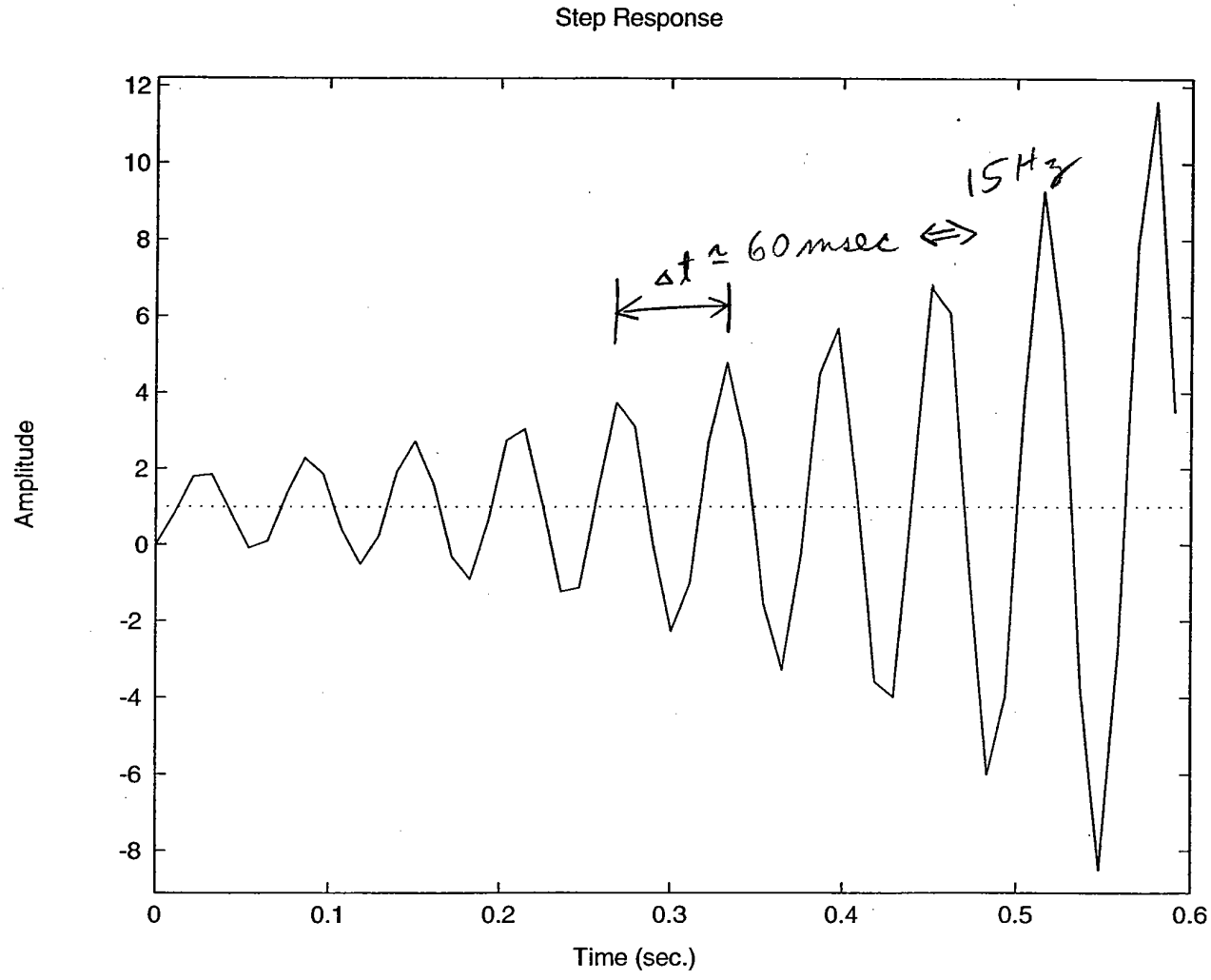
  $\Rightarrow$ usually end up tuning K empirically

  - use rise time and amount of ringing to judge bandwidth and relative stability, respectively.

- This tuning procedure is approximate, but gets you pretty close. Then all 3 parameters can be fine tuned.

(4.8)

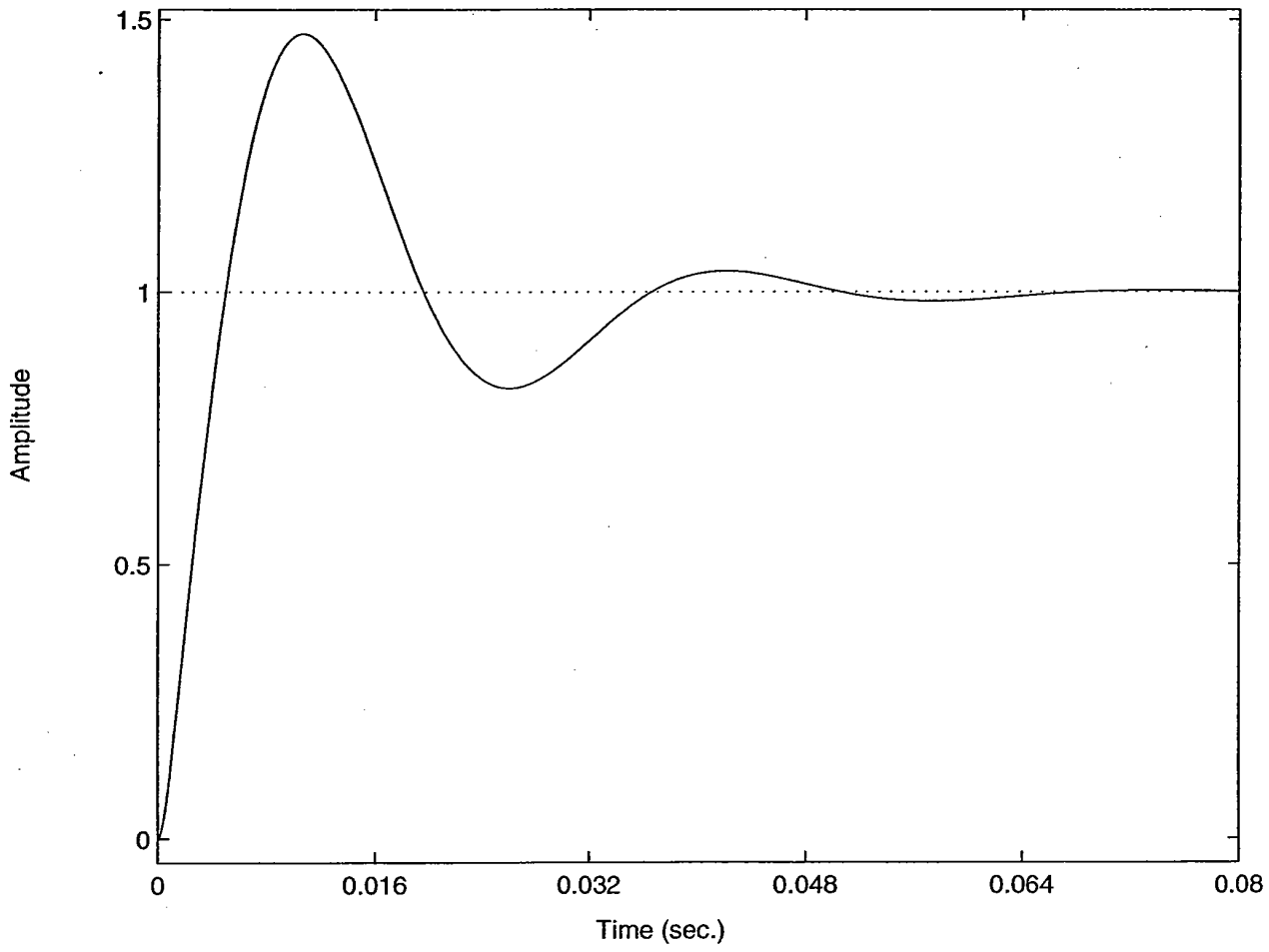
$K = 100{,}000$ (unstable!)



(4.9)

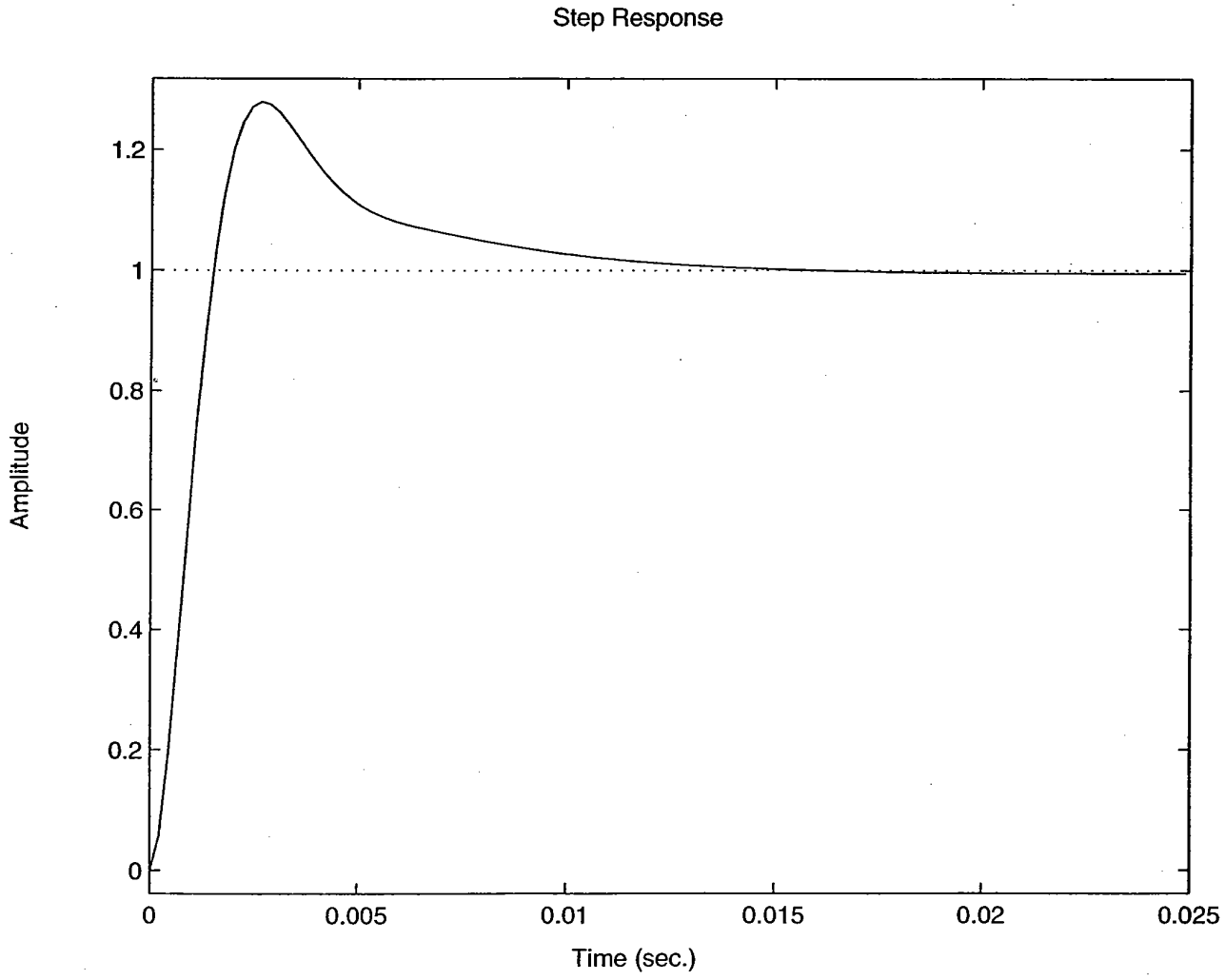
$K = 500{,}000$

$\Delta t \approx 30$ msec $\Leftrightarrow$ 30 Hz (too low)



4.10

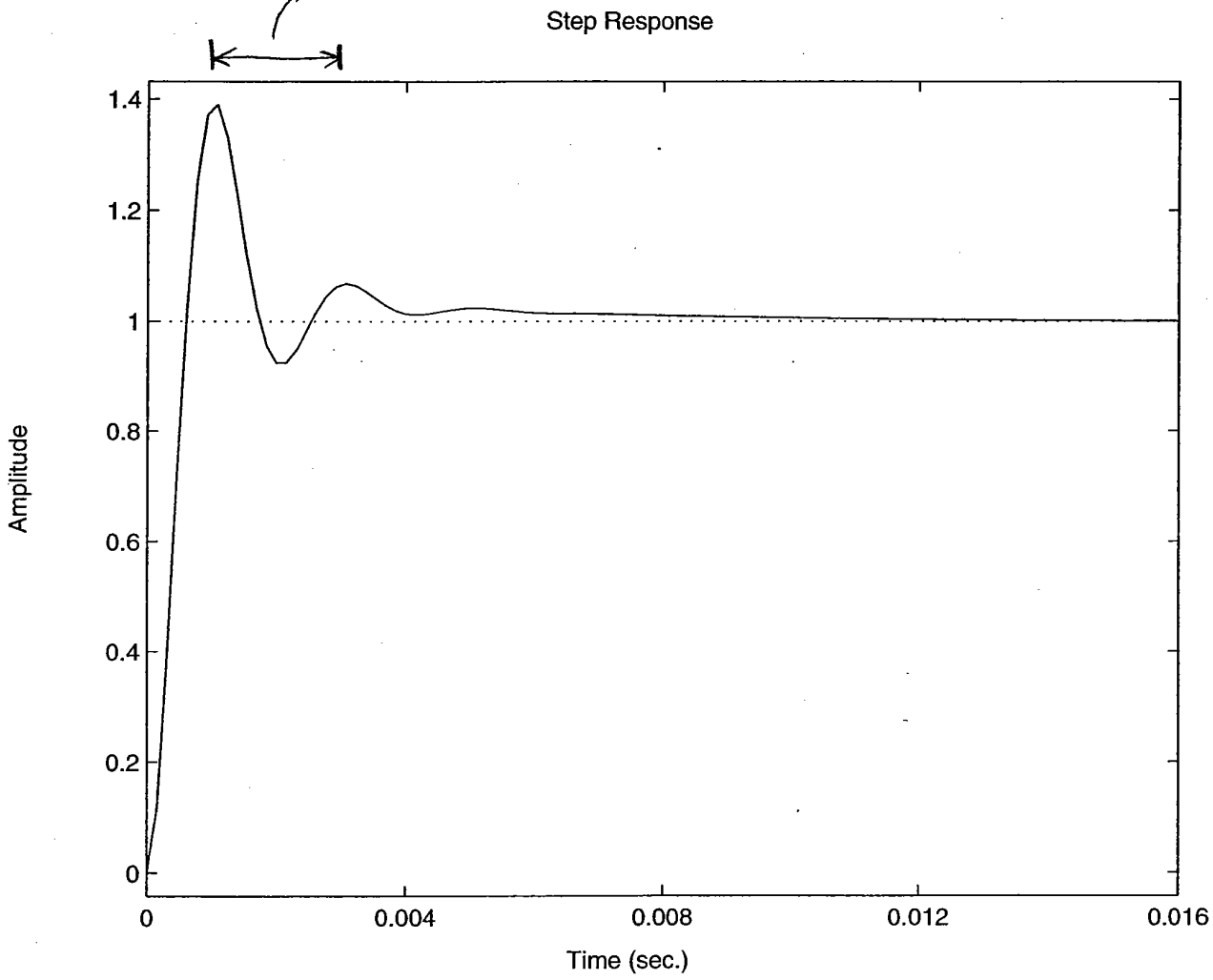
$K = 2.5 \times 10^6$



4.11

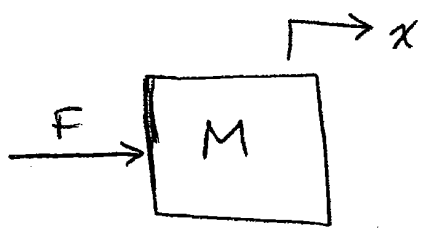
$K = 10,000,000$

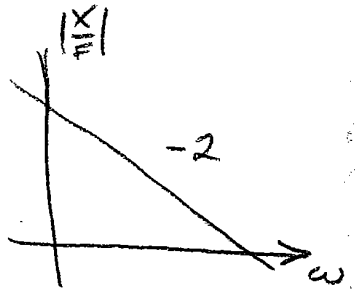
$\Delta t \approx 2$ msec $\Leftrightarrow$ 500 Hz (too high)



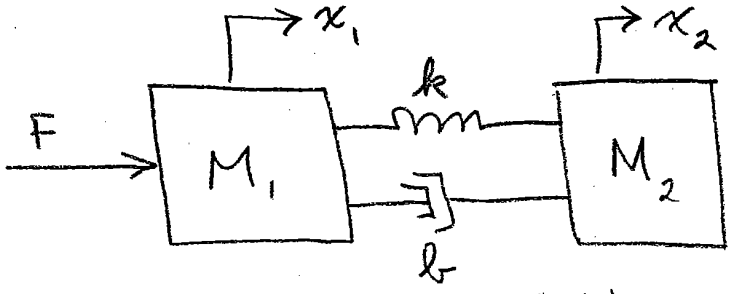
- Flexibility and resonances cause significant problems with stability

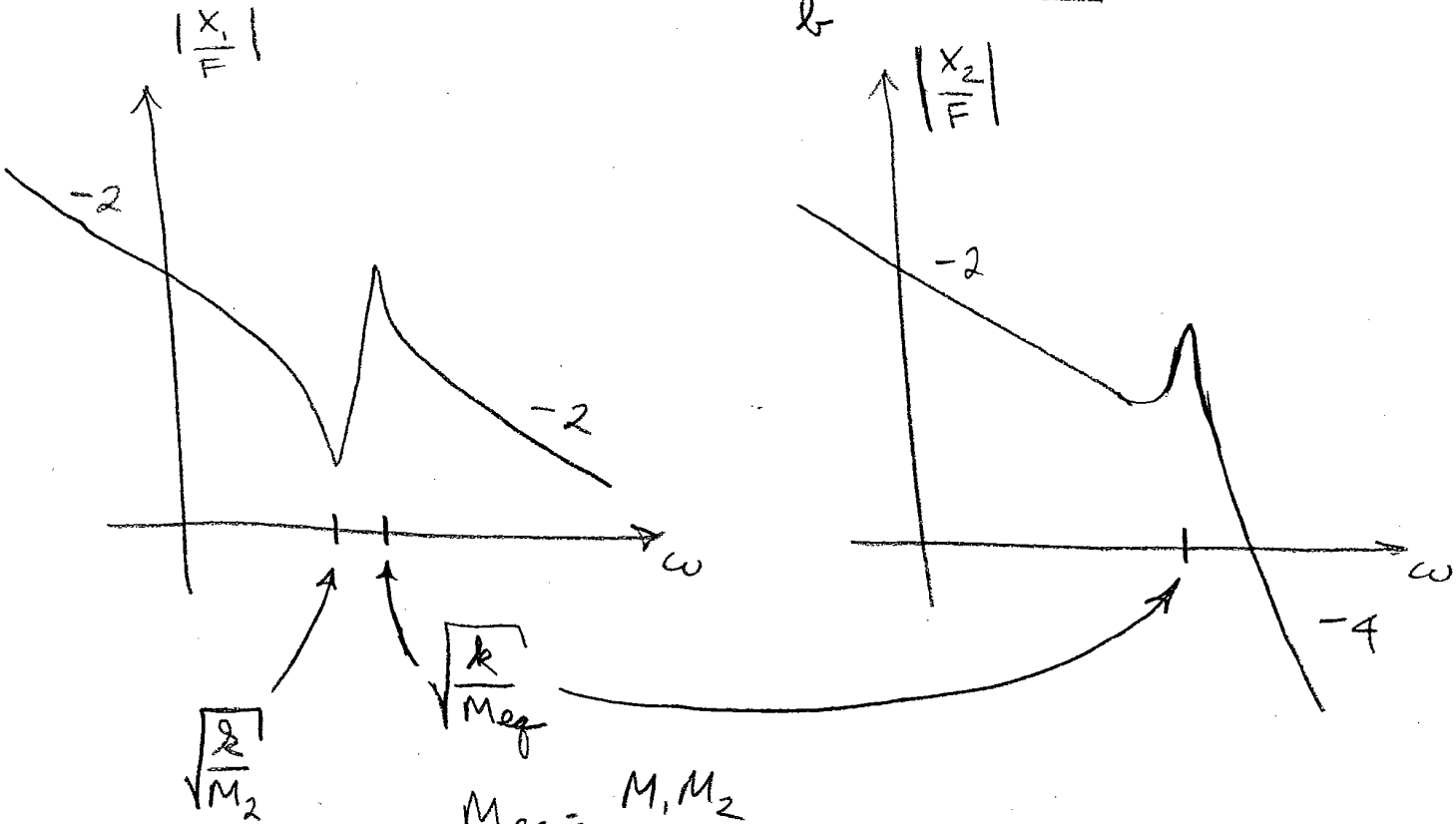
Model: (rigid)





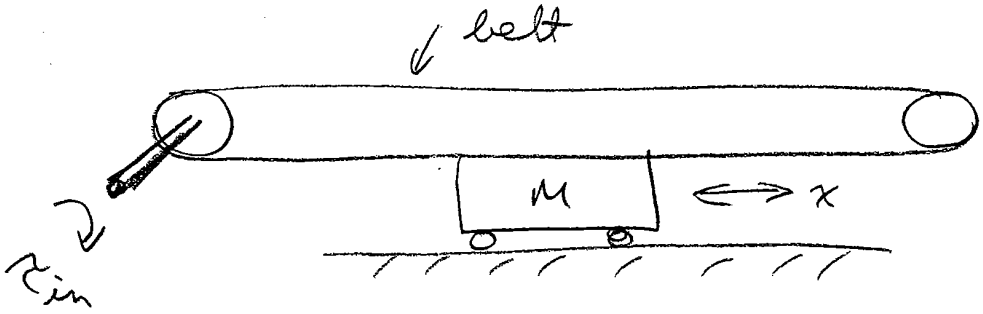
with flexibility:

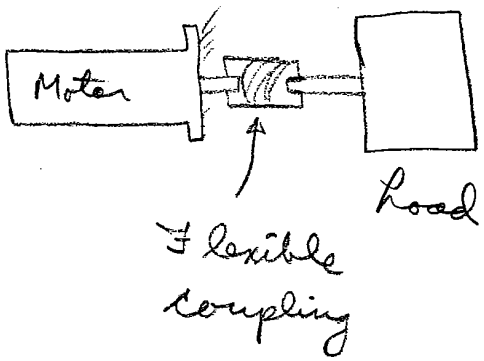


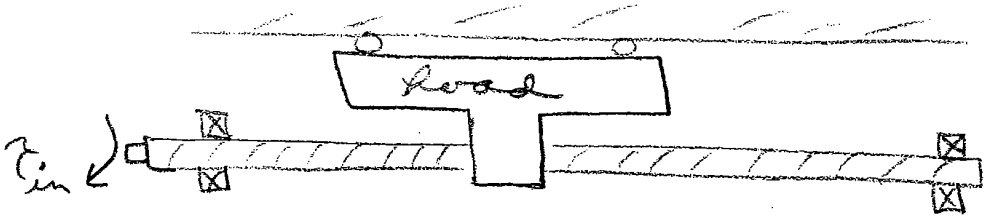


$$M_{eq} = \frac{M_1 M_2}{M_1 + M_2}$$

- This problem occurs in all mechanical systems at some range of frequencies







Ball-screw driving load

- Derive equations of motion for 2-mass model:

$$M_1\ddot{x}_1 = b(\dot{x}_2 - \dot{x}_1) + k(x_2 - x_1) + F$$

$$M_2\ddot{x}_2 = b(\dot{x}_1 - \dot{x}_2) + k(x_1 - x_2)$$

In state variable form:

$$\underbrace{\begin{bmatrix} \dot{x}_1 \\ \ddot{x}_1 \\ \dot{x}_2 \\ \ddot{x}_2 \end{bmatrix}}_{\dot{\bar{x}}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k}{M_1} & -\frac{b}{M_1} & \frac{k}{M_1} & \frac{b}{M_1} \\ 0 & 0 & 0 & 1 \\ \frac{k}{M_2} & \frac{b}{M_2} & -\frac{k}{M_2} & -\frac{b}{M_2} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ \dot{x}_1 \\ x_2 \\ \dot{x}_2 \end{bmatrix}}_{\bar{x}} + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{M_1} \\ 0 \\ 0 \end{bmatrix}}_{B} F$$

$$\dot{\bar{x}} = A\bar{x} + Bu$$

To get outputs x_1, x_2 need

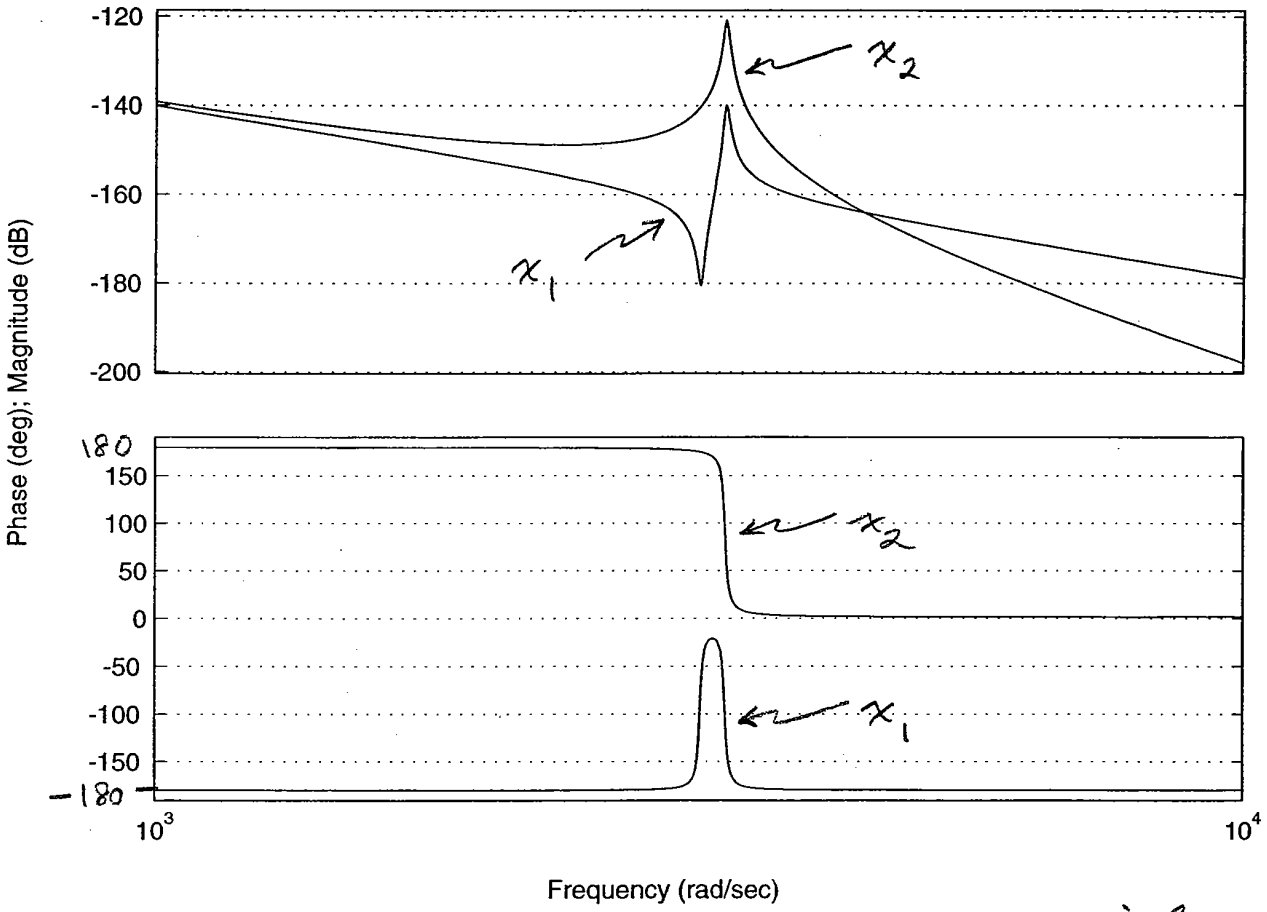
$$y = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{C} \bar{x}$$

(4.15)

$m_1 = 9$ kg
$m_2 = 1$ kg
$k = 10^7$ N/m
$b = 30$ N sec/m

Bode plots $\frac{X_1}{F}(j\omega)$ and $\frac{X_2}{F}(j\omega)$

← acts as rigid body $M_1 + M_2$



Note +180 and -180 are the same points

4.16

2-mass system w/ $K = 2.5 \times 10^6$

$m_1 = 9$ kg

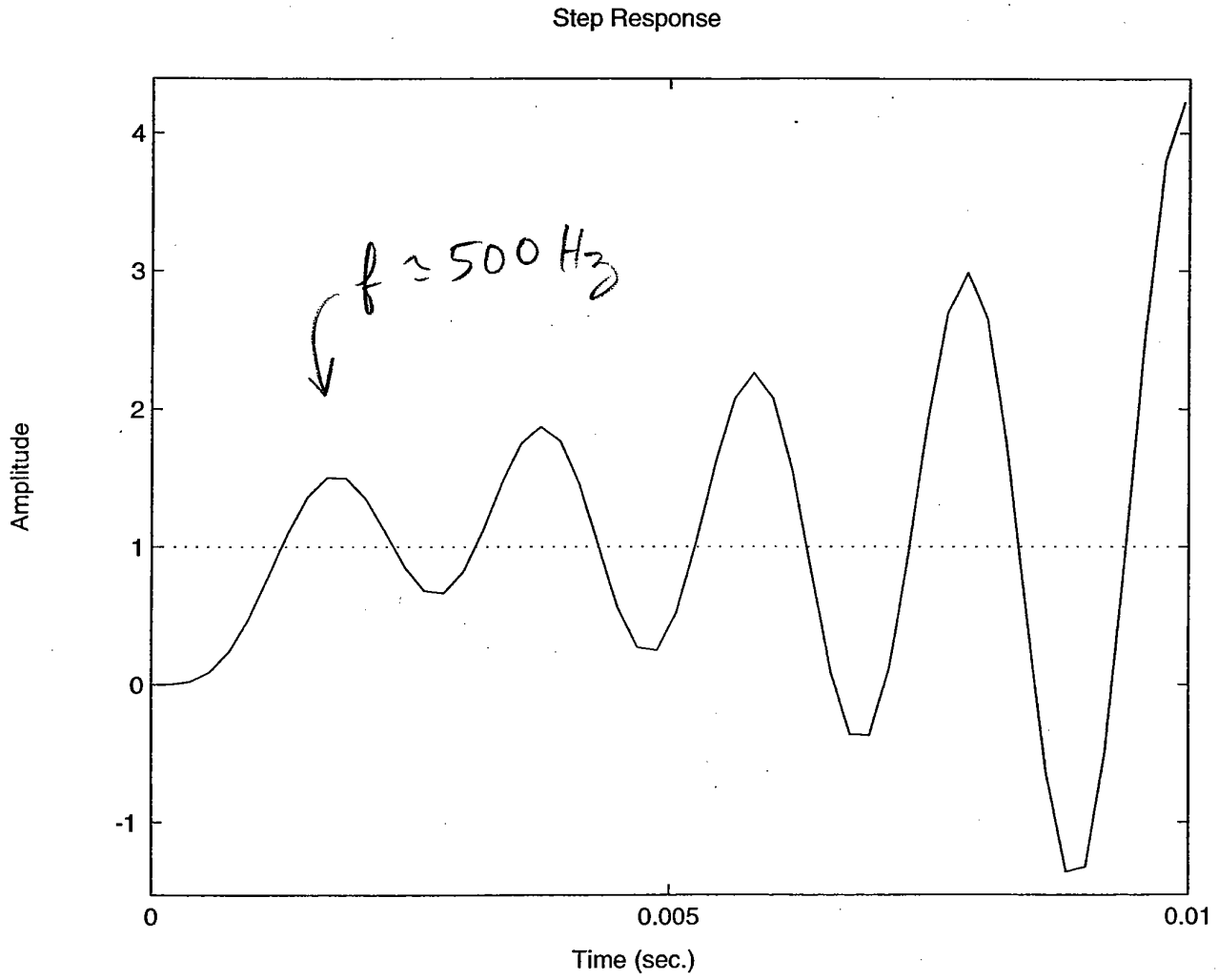
$m_2 = 1$ kg

$k = 10^7$

$b = 30$

$T_d = 4$ msec

$T_i = 20$ msec



4.17

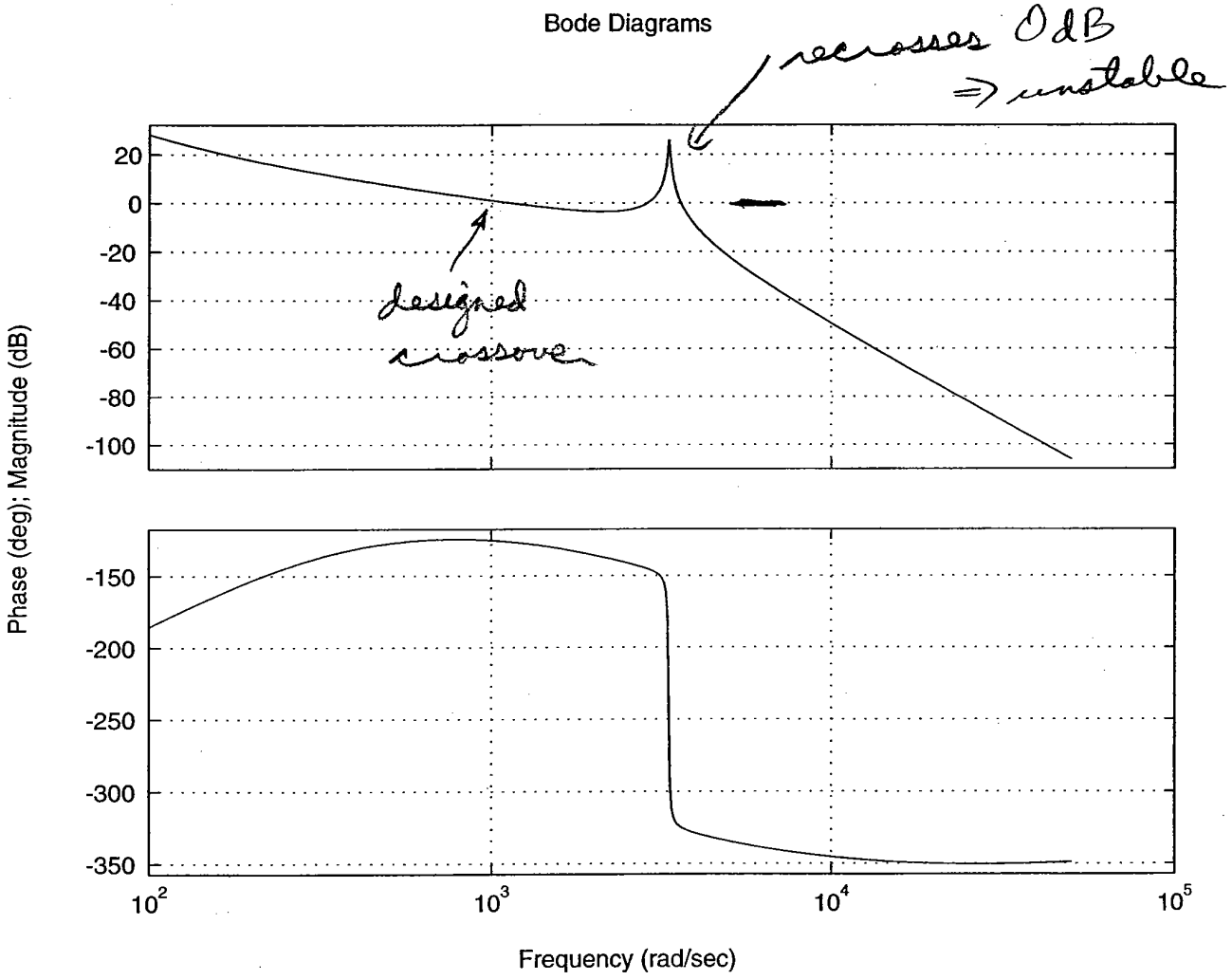
2- mass system

loop transfer function

$K = 2.5 \times 10^6$

$T_d = 4$ msec

$T_i = 20$ msec



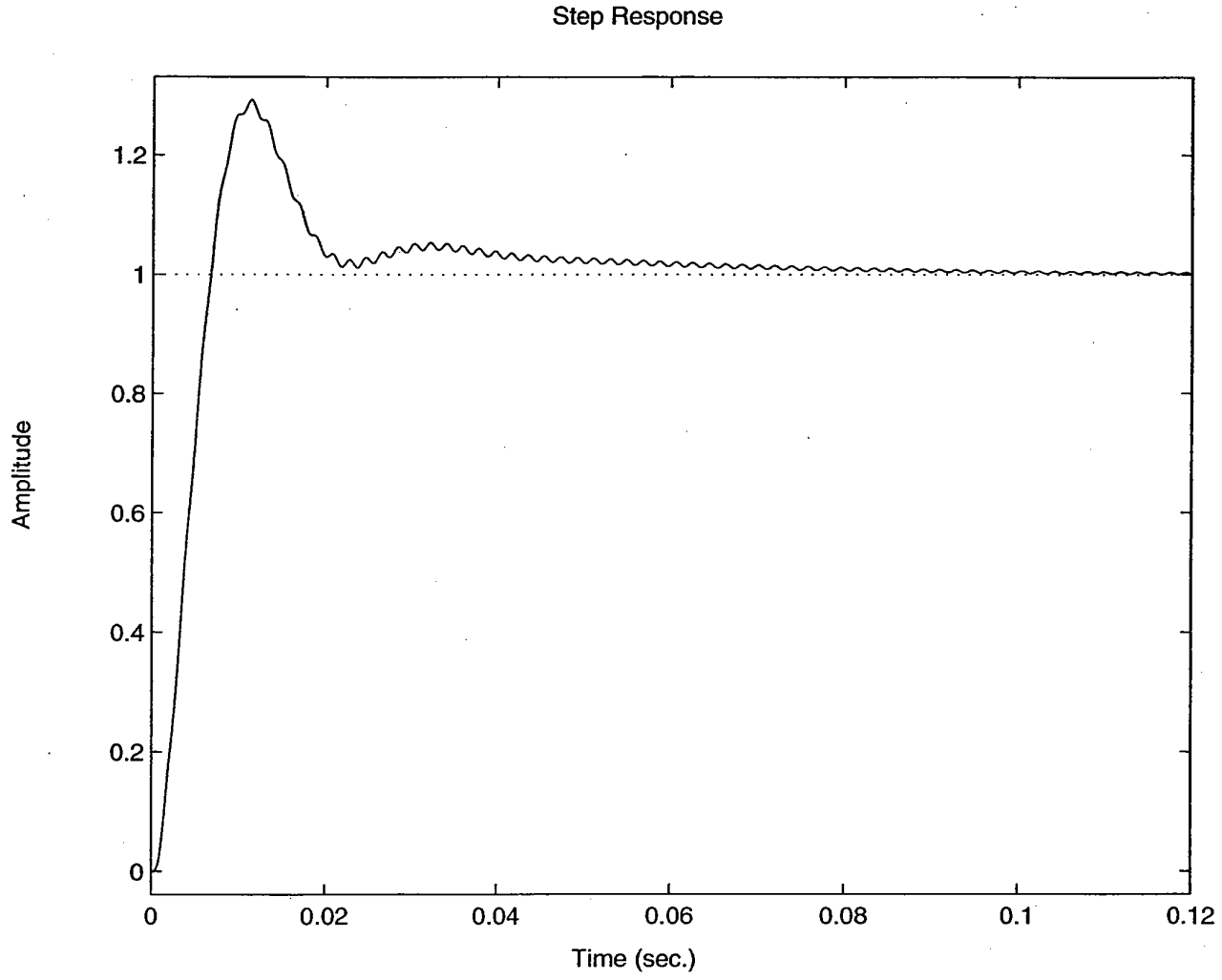
Detune controller

4.18

$K = 100{,}000$

$T_d = 30\ \text{msec}$

$T_i = 300\ \text{msec}$



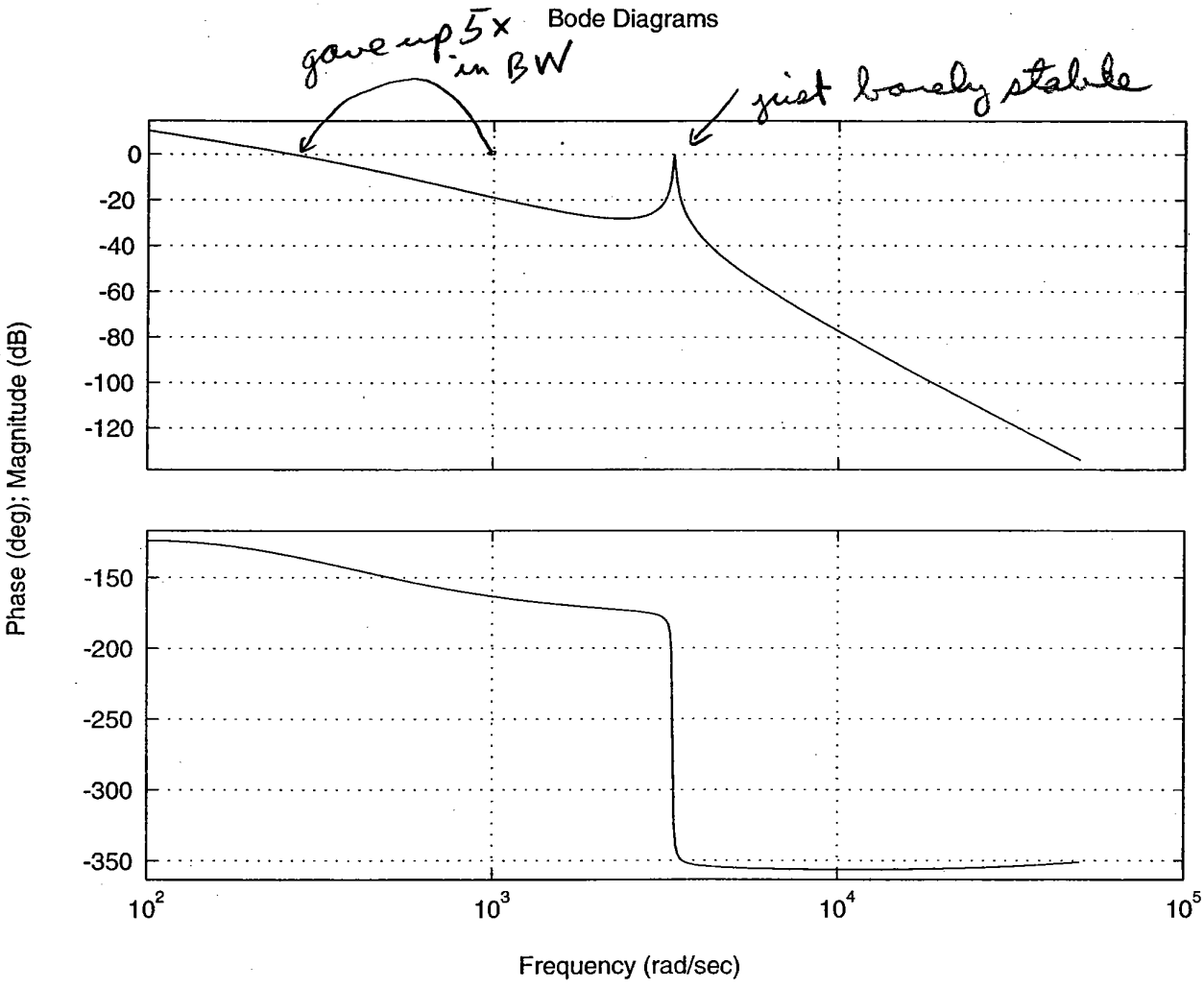
4.19

Loop transfer function

$K = 100{,}000$

$T_d = 30$ msec

$T_i = 300$ msec



4.20

Loop transfer function

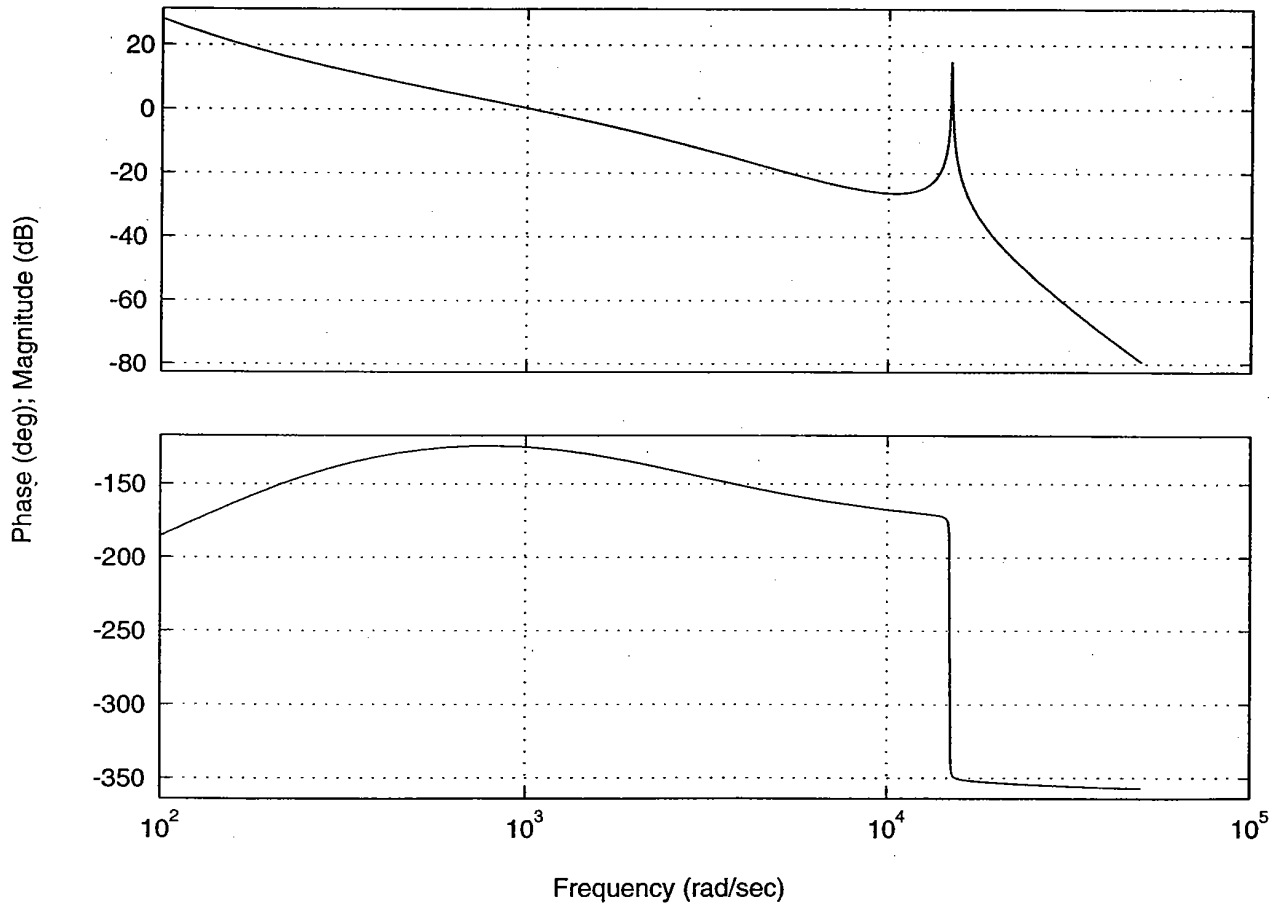
Stiffen k by 20× ⇒ $k = 20 \times 10^7$

$(b = 30)$

$K = 2.5 \times 10^6$

$T_d = 4$ msec

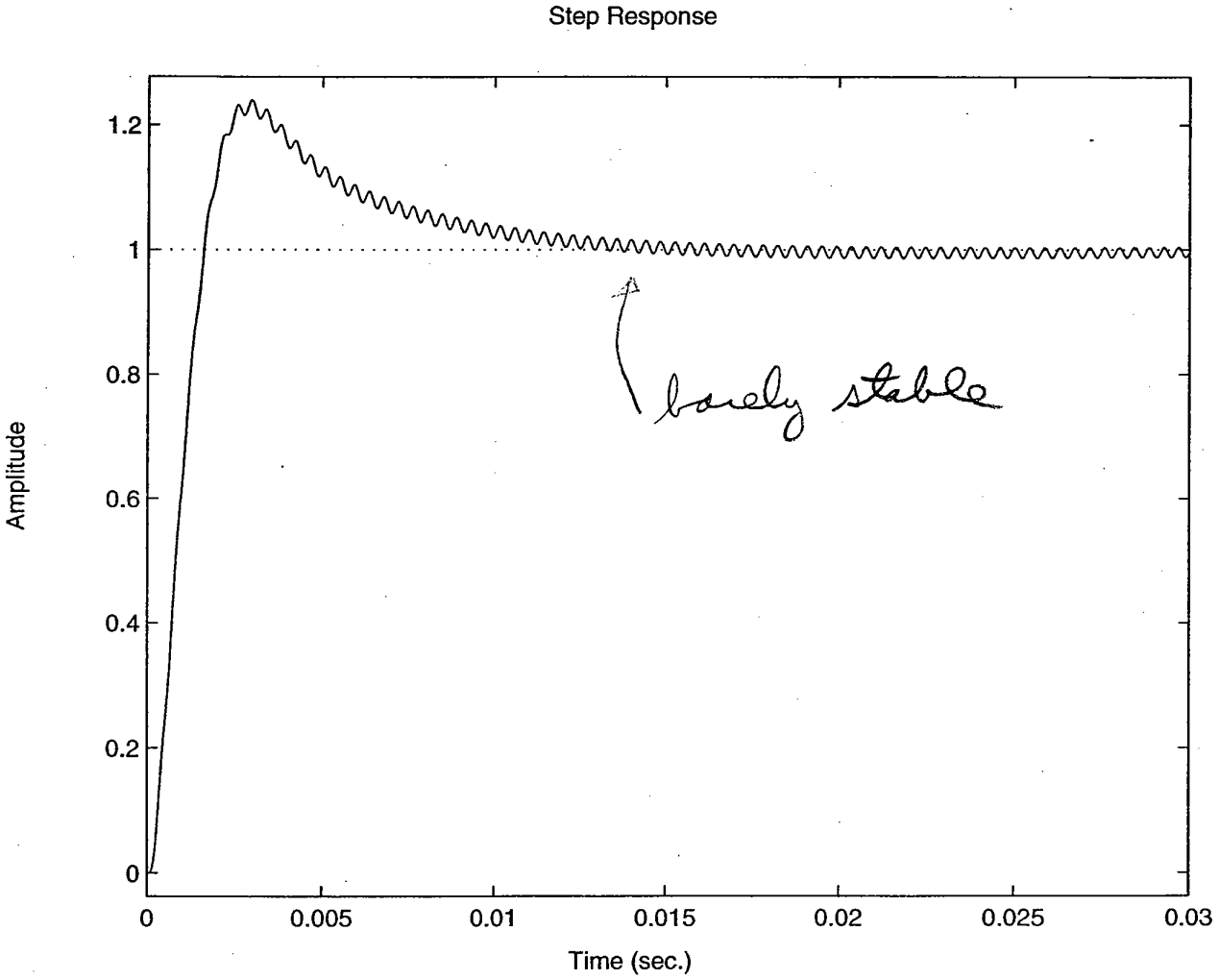
$T_i = 20$ msec



4.21

Step Response

Stiffen k by 20×



4.22

loop transfer function

Stiffen structure and increase damping

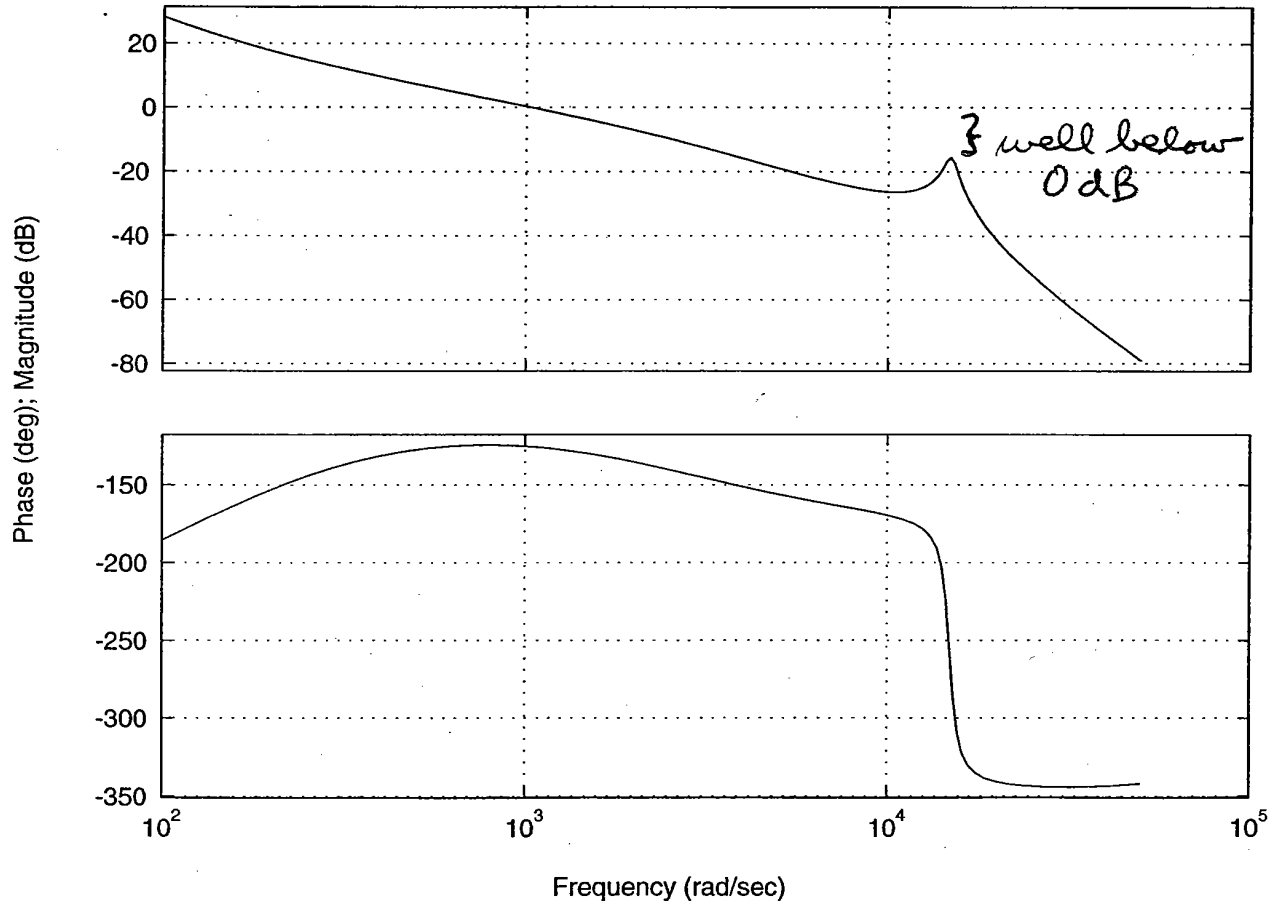
$k = 20 \times 10^7$

$b = 1000$

$K = 2.5 \times 10^6$

$T_d = 4$ msec

$T_i = 20$ msec.



4.23

Stiffen structure and increase damping

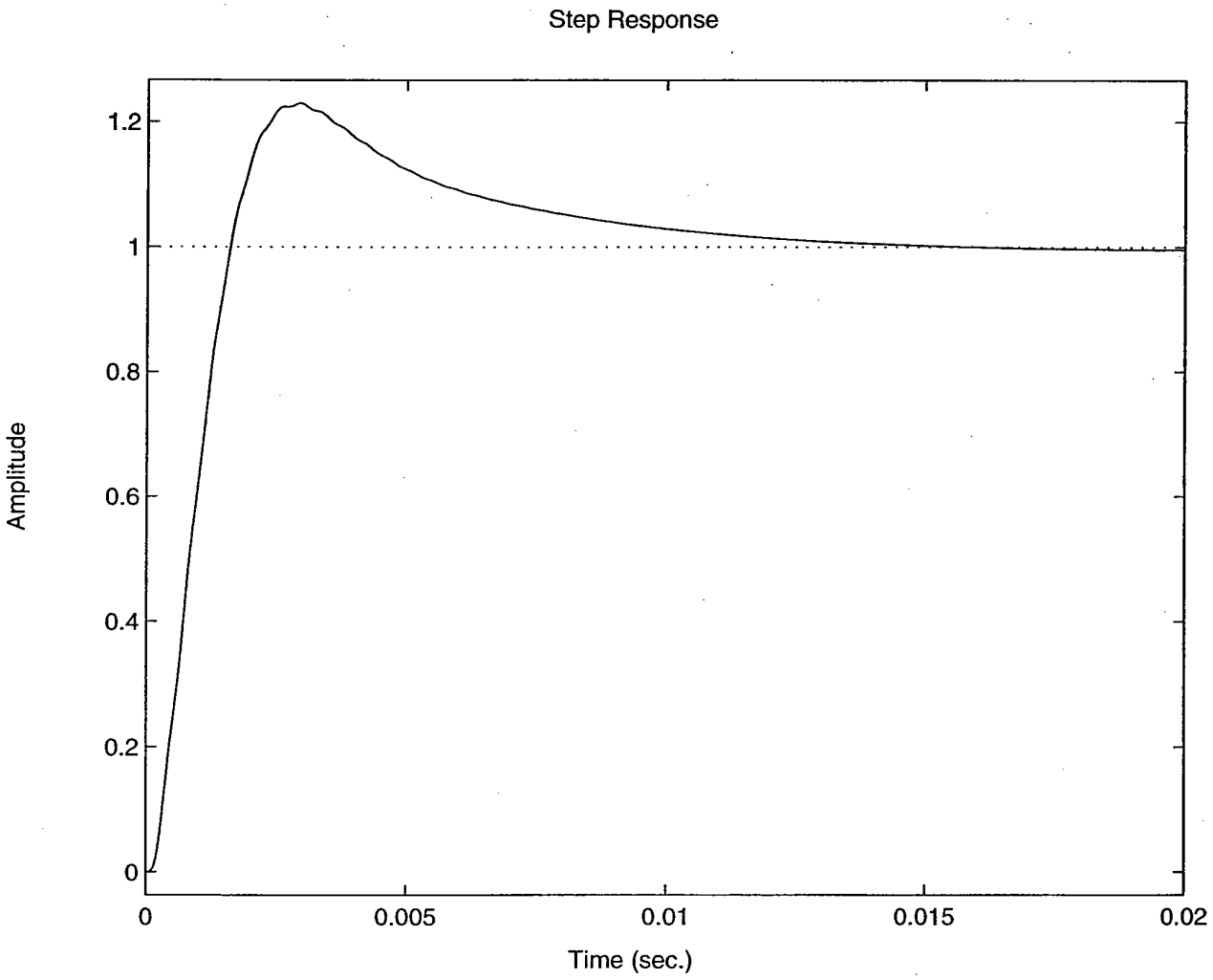
$k = 20 \times 10^7$

$b = 1000$

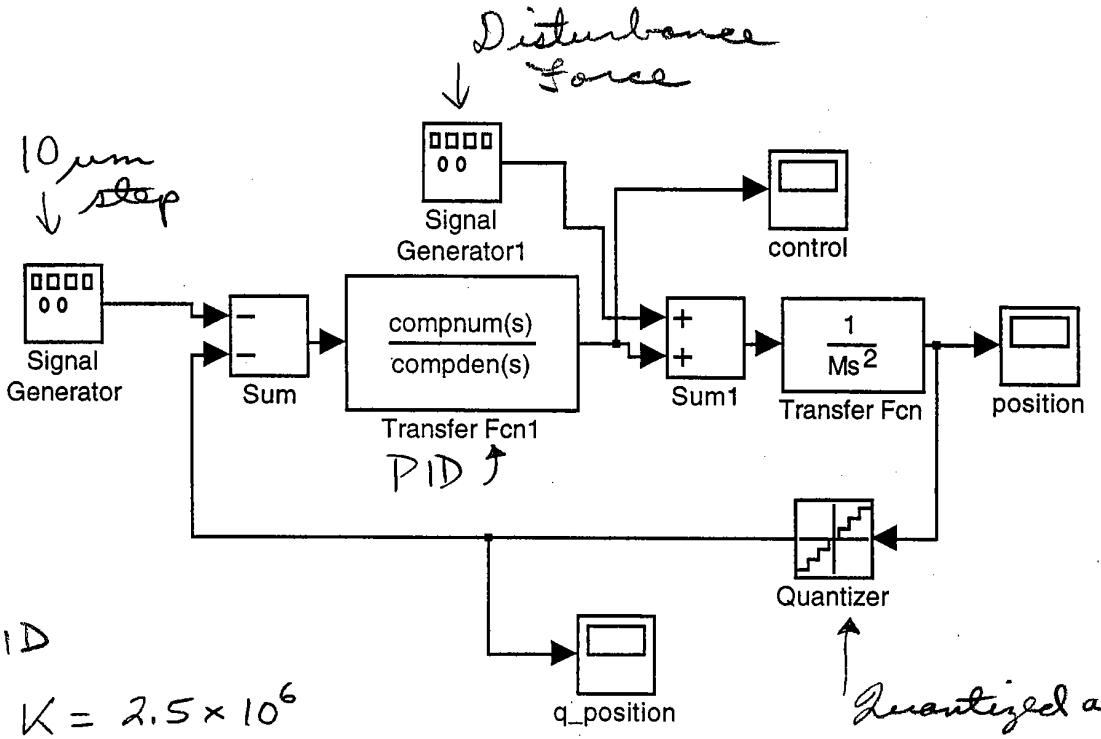
$K = 2.5 \times 10^6$

$T_d = 4\,msec$

$T_i = 20\,msec$



Model with quantization



PID

$K = 2.5 \times 10^6$

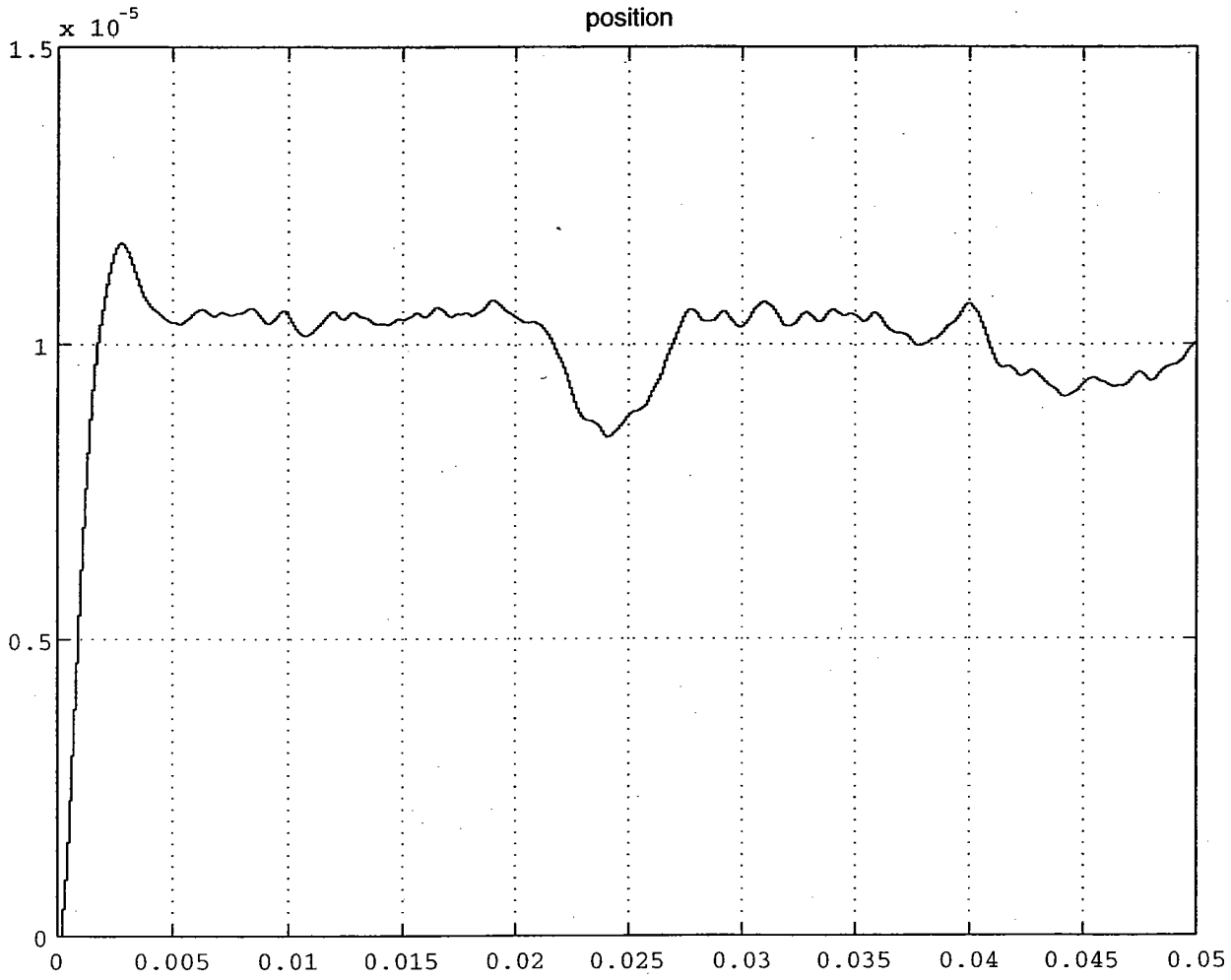
$T_d = 4$ msec

$T_i = 20$ msec

$M = 10$ kg

4.25

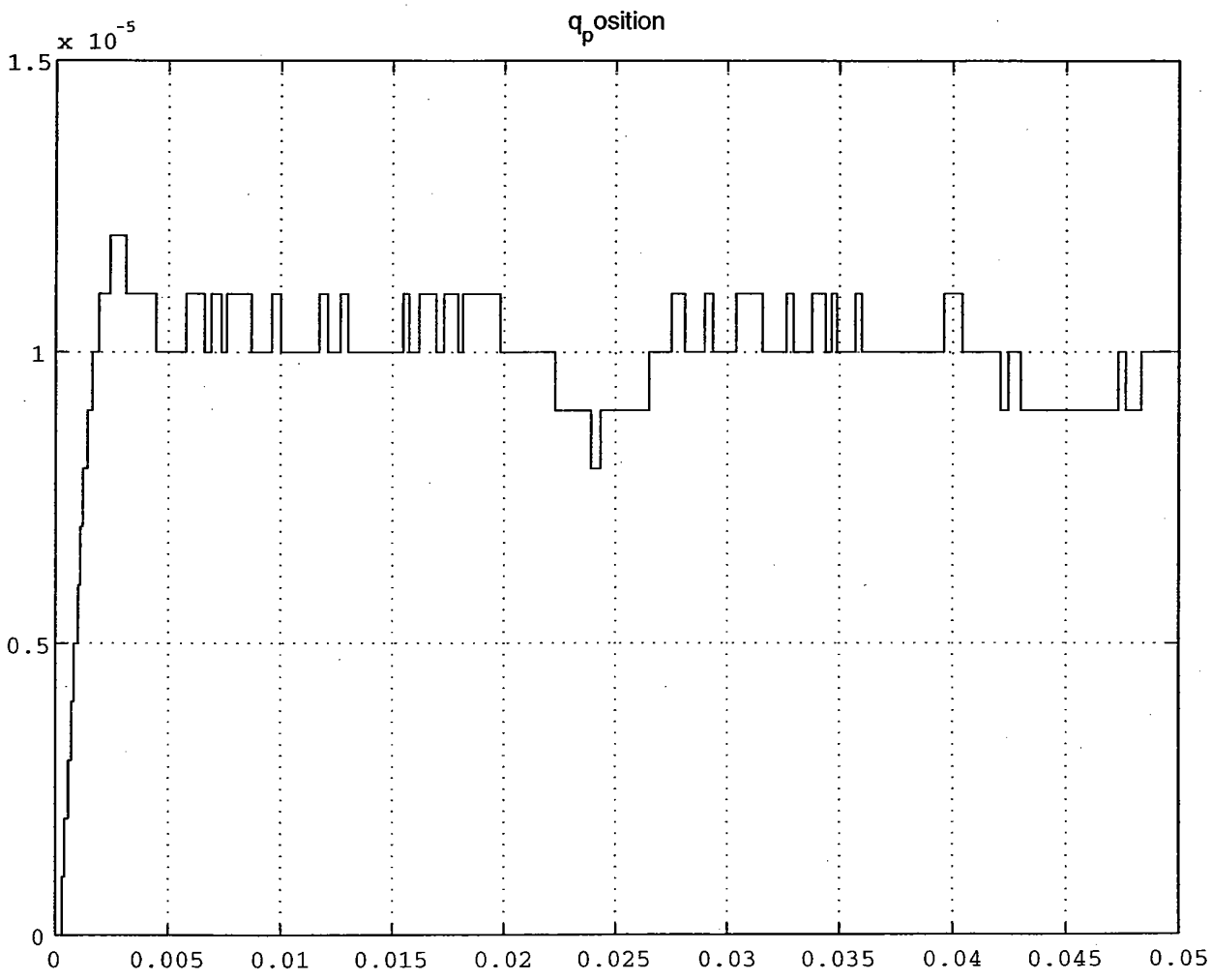
Position



me offset: 0

4.26

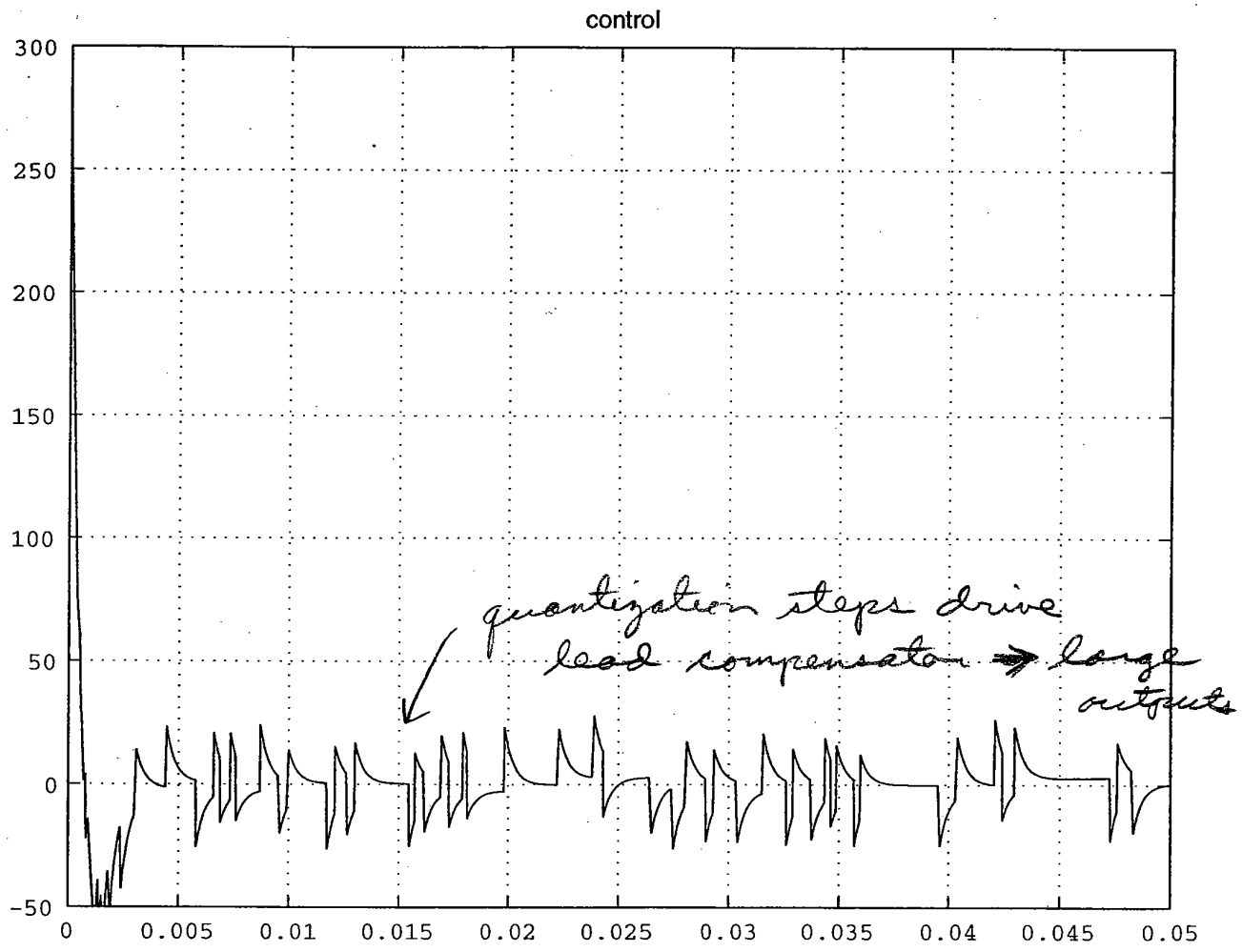
Quantized position



me offset: 0

4.27

Controller output



me offset: 0

Note large control action caused by quantization. Sounds like grit in gears.