09-723 Proximal probe techniques. Name___________________________

Homework #8.

Due by November 24, 2004

Download the Homework8_files.zip file which contains all necessary Matlab files except the model that will be sent via e-mail (we will be providing some tricky to build parts in a pre-built model, but you will still need to add some functionality to it) by clicking on it and choosing to save it on your computer (e.g. on your desktop). Open the archive on your computer and extract it to some folder. Make sure that a path is set in Matlab to this folder. To set a path, choose the set path option in the file menu of Matlab. Click the Add with subfolders button and locate the folder to be added to the path in the pop-up window and click OK. Then click Save and Close.

Questions? Contact Justin at:

justinl@andrew.cmu.edu
phone: 8-9175

Or Tomek at:

tomek@andrew.cmu.edu
Phone: 8-5927

Closed-Loop AFM (some assembly required)

All necessary parameters are entered through the provided Graphical User Interface (GUI), which also controls the execution of the Simulink model, which is partially provided via e-mail. You will be asked to make some minor adjustments to this model. Save the new model. Save the model in the folder that you extracted the contents of **Homework8.zip** into. You should have already set a path to this folder. Make sure that in Here is a list of variables in the model:

Mass: `Mtip` (As promised in previous homework, this is now the mass of the tip)
Spring constant: `Kspring`
Damping Coefficient: `Bdamp`
The simulation step (Fixed-step size): `SimStep`
The total simulation time (i.e. Stop Time)= `SimTime`
The initial velocity = `Zprime` (initial condition for the first integrator)
The initial position = `Zstart` (initial condition for the second integrator and the drive offset)
The position trajectory = `Ztraj`
The time = `time`
The driving force amplitude = `Fo`
The drive frequency (operating frequency in Hz) = `Foper`
The phase = `Phi`
The potential energy = Penergy
The kinetic energy = Kenergy
The damping power = Bpower
The damping energy = Benergy
The drive power = Dpower
The drive energy = Denergy
The ramp stop distance = Zend
The ramp position = RampPos
The Hamaker constant = Atip
The tip radius = Rtip
Sigma = sigma
The Lennard Jones force = FLJ
The ramping rate = RampRate
Indent constant = IndentConst
aDMT = aDMT
Starting deflection = Zstep
Proportional gain = Pgain
Integral gain = Igain
Derivative gain = Dgain
The tip amplitude = Amplitude

If these names are not used, the GUI will not work.
All other variables will be defined through the GUI.
Here are some notes on the additions in the provided model (DynamicAFM_with_feedback.mdl):

I. **The ability to add surface topographical features has been added.**

In the top right corner of the model, notice that there is a **Signal Builder.** This is used to input surface features like steps and holes, and this signal is added into the z position of the tip before going into the potential. The **gain** scales the feature to the step height you enter in to the GUI. For a hole, enter a negative step height. There are two step signals provided, the top signal has abrupt steps, and the bottom signal has gentler steps. To switch between these two steps, click on the **manual switch**. The actual surface topography is called SurfTop and sent to the workspace.

II. **The ability to modify surface modulus has been added.**

Notice that the DMT potential has been modified to allow surface modulus to change during the simulation. There is now a **Signal Builder** with the same signals provided as for topographical features that can be scaled to change the Young's modulus of the surface. This can be set from the GUI. The **E1 in GPa** value will set the Young's modulus at the beginning of the simulation. The **E2 in terms of E1** value will set the scaling of the changed Young's modulus of the sample. If you want a harder surface, put in a positive value as a multiplier. For example, an **E2 in terms of E1** value of 10 will give you a 10 times harder surface. If you want a softer surface, place in a value between -1.0 and 0. For example, a value of 0.9 will give you softer surface with E2=0.1*E1. A manual switch allows you to switch between the different signals. If you want to correlate a change in modulus with a step, make sure that you use the same signal for both topography and modulus.

III. **A switch to go from contact mode to non-contact mode**

A **manual switch** has also been added to the string of blocks that calculates the deflection/amplitude of the cantilever. Contact mode imaging requires monitoring of the mean of the deflection signal, and a **mean** block is added for this. Non-Contact mode imaging requires the monitoring of the amplitude of the deflection signal, and a **Max** and **Min** blocks are added to accomplish this. You will need to use the **Manual Switch** to use the appropriate signal for both imaging modes.

IV. **Proportional/Integral/Derivative feedback loop.**

You will be adding these loops and exploring the use of these gains in question #1.

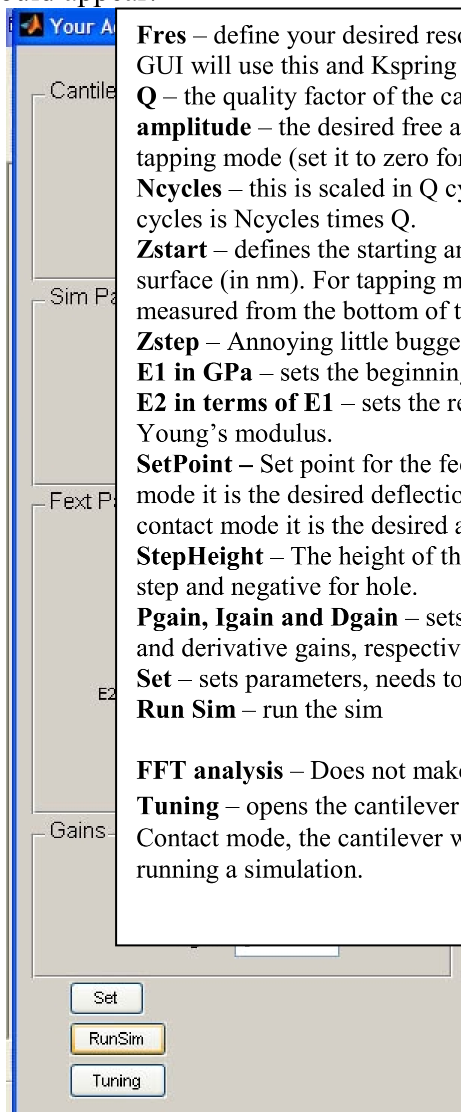
V. **Several signal specification blocks have been added.**

This is done to avoid crashes.

Before running the model you need to enter the values of other parameters using the provided GUI. To use the GUI provided, run `Homework8` from command line.

```
>> Homework8
```

The following interface should appear:



Fres – define your desired resonance frequency in Hz, the GUI will use this and Kspring to determine Mtip, etc.
Q – the quality factor of the cantilever.
amplitude – the desired free amplitude of the cantilever in tapping mode (set it to zero for contact mode force curves)
Ncycles – this is scaled in Q cycles now, i.e. the number of cycles is Ncycles times Q.
Zstart – defines the starting and ending distance from the surface (in nm). For tapping mode, the distance is measured from the bottom of the oscillation
Zstep – Annoying little bugger, isn't it.
E1 in GPa – sets the beginning surface Young's modulus.
E2 in terms of E1 – sets the relative changed surface Young's modulus.
SetPoint – Set point for the feedback loop. In contact mode it is the desired deflection to keep constant. In non-contact mode it is the desired amplitude to keep constant.
StepHeight – The height of the step in nm. Positive for step and negative for hole.
Pgain, Igain and Dgain – sets the proportional, integral, and derivative gains, respectively.
Set – sets parameters, needs to be run before simulations.
Run Sim – run the sim

FFT analysis – Does not make an appearance in this GUI
Tuning – opens the cantilever tuning menu. In non-Contact mode, the cantilever will need to be tuned before running a simulation.

PROBLEM SET

Carry out the tasks outlined in each problem. Answer the questions posed and, whenever necessary, illustrate your answers with the graphical output of the simulations.

PROBLEM #1 – EXPLORING GAINS IN CONTACT MODE

Set **Kspring** = 0.5, **Fres** = 30000, **Q** = 25, **amplitude** = 0, **Ncycles =** 20, **PtsPerCycle** = 256, **Zstart** = 0, **Zstep** = 0, **Atip** = 0.425, **Rtip** = 10, **aDMT** = 4e-10, and **E1 in GPa** = 100, **E2 in terms of E1** = 0, **SetPoint** = 100, **StepHeight** = 1, **Pgain** = 0, **Igain** = 0, and **Dgain** = 0.

Note: In this contact mode problem, do not tune the cantilever.

First, add the feedback loop with all of the gains. Place an **Add** block in the model. To this **Add** block connect the **Amplitude Signal** to a + and the SetPoint from a **constant** block to a – (the **constant** block will need to add to the model). Set the value of the **constant** block to **SetPoint**. Send the **signal** from the **add** block through a **gain** with the value of **PGain**. Take the **signal** from the **add** block again, and send it through an **integrator** and then a **gain** with the value of **Igain**. Take the **signal** from the **add** block a third time and send it through a **derivative** block and then a **gain** with the **value** of **Dgain**. Hook the **signals** from the three gains into the **Add** block labeled **add gains here**.

Leave all the gains set to 0. Press **Set** and then **RunSim** to run the AFM in open loop.

1.1 What happens to the deflection?

Vary the **Pgain** between 0 and 0.6, and leave **Igain** and **Dgain** set to 0. Press **Set** and then **RunSim** to see how well the **Pgain** is able to trace the surface**.**

1.2 Are you able to accurately track the step using just **Pgain**?

Vary the **Igain** between 1000 and 20000, and set **Pgain** and **Dgain** to 0. Press **Set** and then **RunSim** to see how well the **Igain** is able to trace the surface**.**

1.3 Are you able to accurately track the step using just **Igain**?

1.4 What happens if you set the **Igain** between -100 and -20000 and why?

Vary the **Dgain** between 0 and 1e-7, and set **Pgain** and **Igain** to 0. Press **Set** and then **RunSim** to see how well the **Dgain** is able to trace the surface**.** Are you able to accurately measure the step height using just **Dgain**?

1.5 Which gain appears to give the best tracking of the step?

Now, set **Pgain** = 0.1, **Igain** = 7500, and **Dgain** = 0. Press **Set** and then **RunSim.**

1.6 Notice that the deflection signal look like a derivative of the step. Why is there an overshoot in the trace at the beginning of the step? How would you overcome this?

#2 – COMPLIANCE BASED CONTRAST IN CONTACT MODE

Set **Kspring** = 0.5, **Fres** = 30000, **Q** = 25, **amplitude** = 0, **Ncycles =** 20, **PtsPerCycle** = 256, **Zstart** = 0, **Zstep** = 0, **Atip** = 0.425, **Rtip** = 10, **aDMT** = 4e-10, and **E1 in GPa** = 100, **E2 in terms of E1** = -0.9, **SetPoint** = 100, **StepHeight** = 0, **Pgain** = 0.1, **Igain** = 7500, and **Dgain** = 0.

Press **Set** and then **RunSim.**

2.1 Why does the trace signal show a difference in height even though there is no step in the surface?

Vary the **SetPoint** between 25 and 250 nm.

2.2 What happens to the feature observed in the trace and why?

#3 – EXPLORING CLOSED-LOOP NON-CONTACT MODE AFM

Set **Kspring** = 0.5, **Fres** = 30000, **Q** = 25, **amplitude** = 10, **Ncycles =** 20, **PtsPerCycle** = 256, **Zstart** = 2, **Zstep** = 0, **Atip** = 0.425, **Rtip** = 10, **aDMT** = 4e-10, and **E1 in GPa** = 100, **E2 in terms of E1** = 0, **SetPoint** = 9, **StepHeight** = 1, **Pgain** = 0, **Igain** = 0, and **Dgain** = 0.

If you only change the gains, you will not have to re-tune the cantilever. Press **set** and **RunSim**. If you change any of the other variables, you will need to re-tune.

Flip the switch to set your model into non-contact mode. Applying what you learned about gains in problem #1, try to adjust the gains for non-contact mode imaging of this 1 nm step.

3.1 Notice that now your set-point value has to be a value smaller than "free" cantilever amplitude. Why?

3.2 Can you achieve surface tracking using values of gains similar to those you have used for contact mode? What happens if you keep increasing the gains?

3.3 Now see what happens when you change the sign of the gain.
It should work now…<u>Try to find the value of the gains that will result in the satisfactory tracking of the step.</u>

<u>3.4 Why did you have to change the sign of the gain in order to achieve topography tracking?</u>