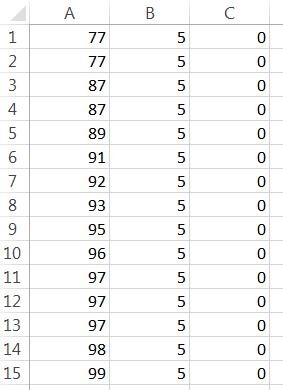
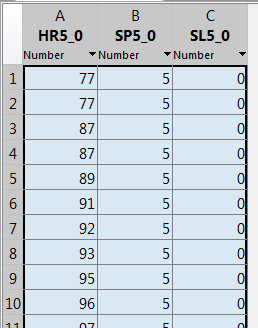
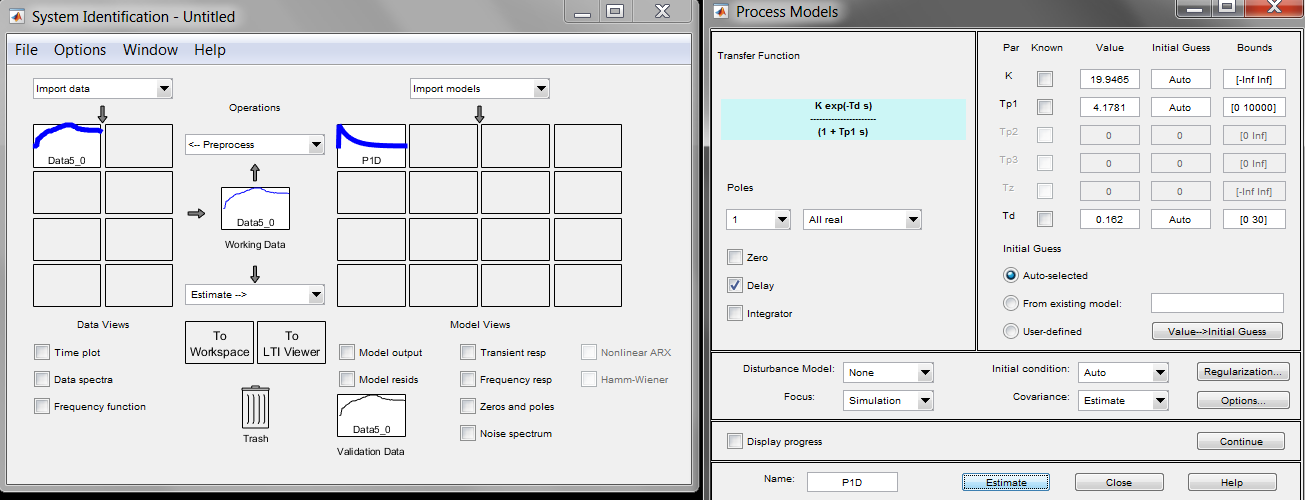
In this experiment, you will use a treadmill and heart rate transmitter(HRT). Remember that the runner shouldn’t change with someone else during the experiment.

1. Wrap the chestband around your chest and be sure the length adjustment is suitable your body size.
2. Plug the transmitter onto the chestband and start the treadmill with quick start option. If you see your heart rate with consistent results, you are ready to start to experiment.
3. Start the treadmill with 5 kph and 0 degree slope, don’t start to run, wait for treadmill to reach its steady-state.
4. Now, you are ready to run. One of your friends should record the data, the transmitted heart rate information will be written in the treadmill’s screen **once in two seconds**. The data recording shouldn’t be problematic, remember that your data will be imported into MATLAB therefore, you might consider using Excel for this purpose.
5. Start running while data is recorded by your friend. Once your heart rate reached steady state, you can stop running and recording.
6. Create speed and slope values for each corresponding heart rate value. It should look like the following:



1. Repeat the steps 3, 4, 5, 6 for “10 kph/0 slope”, “5 kph/5 slope”, “10 kph/5 slope”
2. You collected the necessary data to identify your heart model. Now, open MATLAB and import one of the data into workspace using the “Import Data” option which is in the left upper corner at home screen, name the data accordingly so that you can remember what it represents.



1. We can start to extract a model from the data. Open System Identification tool from the APPS tab. On the upper left corner, choose “Import data>Time domain data”. From “Workspace Variable”, write the speed workspace variable’s name as input and the heart rate as output. Name the data from “Data Information” and click import. Your data is ready to modelling. Examine the time response by clicking the “Time plot” on the left lower corner of the window.
2. Make sure that the Working Data is correct data. (Working Data can be different if you import additional datas) Under the “Working Data” choose “Estimate>Process Models”. According to your data, choose a model structre. (i.e. 1 pole + All real with Delay for FOPDT) Click on Estimate. Your estimated model is imported into the table in the right side of the table. Right click on the estimated model and from the popped window name it according to your running case, then export it into the workspace by clicking “Export”.

Click on the “Model output” under the right table and examine the comparison of the fitted model and the actual response. If it is a good fit (minimum %75), record your model parameters.

1. Repeat the steps 8, 9, 10 for the remaining running cases.
2. The identification is completed, now we will create a controller for our model. To do this, open the “exp\_X\_model\_step12.slx”. You will see a classical PID control loop.(Note that we are using a discrete-time PID block). Here, we have “Idmodel” plant block which can be accessed from “Simulink Library>System Identification Toolbox>Models>Idmodel”. Double click on the “Idmodel” block and write your one of the exported model’s name into the Identified Model and apply changes.
3. We inserted our identified model as plant and we use the tuning tool of MATLAB to create a suitable PID. Double click on the PID block and click the “Tune…” button. From the toolbar of the popped window, click on options, under the title “Design” choose “Focus>Reference Tracking”. Then play around the “Response Time” and “Transient Behavior” sliders to have no overshoot in the response. Then from toolbar, click “Update Block”. Now your PID is tuned.
4. Now we will use this PID to control YOUR heartrate when the running cases are the same with the conditions which PID and model are extracted. In other words, we will conduct the method “Human in the Loop”. To do this, open the Simulink model “exp\_X\_model\_step14” and change the existing PID parameters with your tuned values(do not replace the PID block since it has calculated limits at output). Change the gain value with your heart model’s gain value. The added value is your running condition bias, also change it according to your running case. (we will try to keep your heart rate around this operating point i.e. 10kph speed). Change the heart rate setpoint according to your running condition’s steady state heart rate value.
5. Your friend will enter your current heart rate once in every 8 seconds (because the discrete PID’s sample rate is adjusted to 10 seconds) Right click on the “Controlled Speed” variable and choose the “Show Value Label of Selected Port”. You will monitor the controlled speed from the resulting yellow label.
6. The controller speed signal will be updated once in every 10 seconds and you should enter this value into the treadmill as its speed control in every update.
7. Start running and adjust your heart rate around the “Heart Rate Setpoint” but do not make it equal exactly. (The controller will equate it)
8. Now, you can start and conduct the steps 15, 16, 17. Your heart rate should stay around the set point. If steady state is reached, stop running and simulation. Record the “Heart Rate Logger” into a figure.