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 the runner’s heart rate consistent, you are ready to start the experiment.
3. Start the treadmill with 5 kph and 0 degree slope, do not start to run, wait for treadmill to reach its steady-state value.
4. Now, you are ready to run. One of your group members should record the data. The transmitted heart rate information will be written in the treadmill’s screen **once in two seconds**. The data recording should not 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 group member. Once the heart rate reached its steady state value, you can stop running and recording.
6. Create speed and slope values for each corresponding heart rate value. It should look like as the following: (A: Heart Rate, B: Speed, C: Slope)

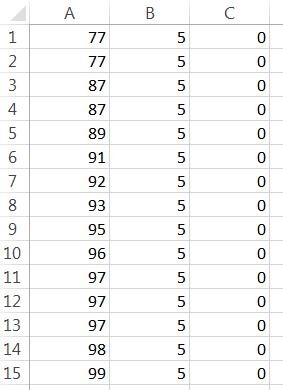


Figure …

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 the heart model. Now, open MATLAB and import one of the data sets into workspace using the “Import Data” option which is on the left upper corner at home screen, name the data accordingly so that you can remember what it represents.

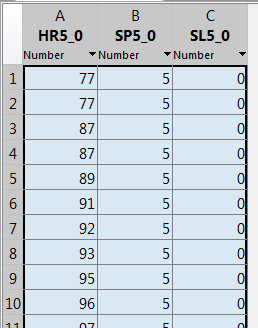


Figure …

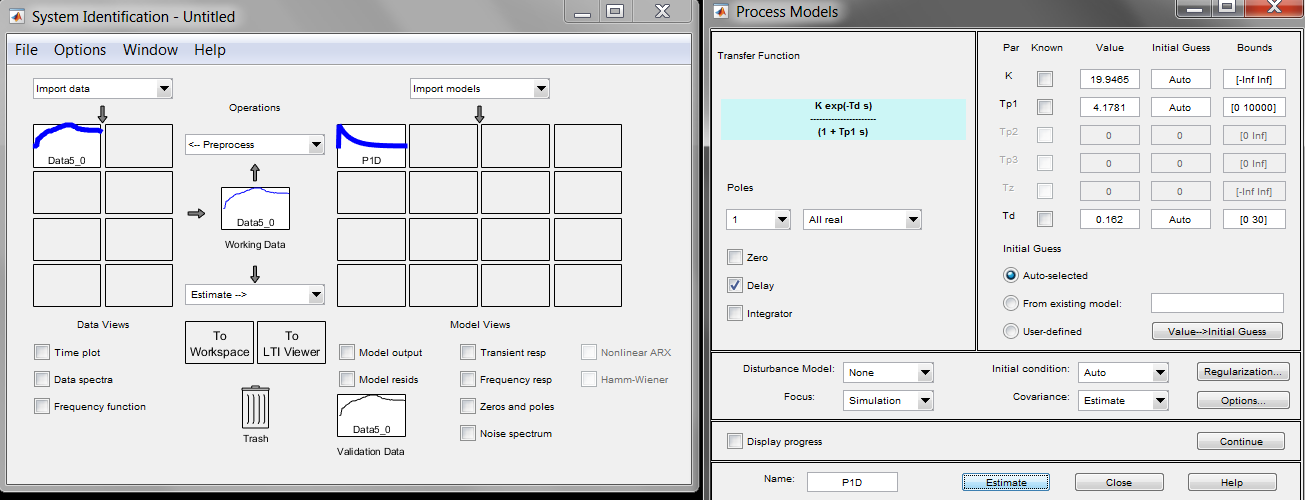
1. You can start to extract a model from the data. Open System Identification Toolbox from the APPS tab. On the upper left corner, choose “Import data>Time domain data”. From “Workspace Variable”, write the speed workspace variable’s (SP5\_0 in Figure …) name as input and the heart rate(HR5\_0 in Figure …) as output. Give a name to the data from “Data Information” and click import. Your data is ready for 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 the current data. (Working Data can be different if you import additional data sets) Under the “Working Data” choose “Estimate>Process Models”. According to your data, choose a appropriate model structure. (i.e. “1 pole + All real with delay” for FOPDT) Click on “Estimate”. Your estimated model is imported under “Import models” at the right side of the table. Right click on the estimated model. From the popped window, name it according to your running case. Then, export it into the workspace by clicking “Export”.

Figure …

Click on the “Model output” under the “Model Views” and examine the comparison of the fitted model and the actual response. If it is a reasonable 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 you will create a controller for your model. To do this, open the “exp\_X\_model\_step12.slx”. You will see a classical PID control loop. Note that you are using a discrete-time PID block. Here, you have “Idmodel” plant block which can be accessed from “Simulink Library>System Identification Toolbox>Models>Idmodel”. Double click on the “Idmodel” block and write one of your exported model’s name into the “Identified Model” and apply changes. You inserted your identified model as plant.
3. Now, use the tuning tool of MATLAB to create a suitable PID controller. Double click on the PID block and click on 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 controller is tuned.
4. Now you will use this PID to control **runner’s** heart rate when the running cases are the same with the conditions which PID and model are extracted. In other words, you will conduct the “Human in the Loop” method. 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, just change the parameters). Change the gain value with your heart model’s gain value (calculated Kp value of the FOPDT model). The added value is your running condition bias, also change it as your running speed when the model is extracted. (You will try to keep the heart rate around this operating point i.e. 10kph speed). Change the heart rate set point as the runner’s running condition’s steady state heart rate value.
5. Enter the current heart rate of the runner once in every 20 seconds just before each update(because the discrete PID’s sample rate is adjusted to 20 seconds). Right click on the “Controlled Speed” signal 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 20 seconds and you should enter new speed value into the treadmill just after every update.
7. Start running the program after the heart rate reached around the “Heart Rate Set Point” (± 5 BPM) The controller will equate the runner’s heart rate to the set point.
8. Keep running until you believe the PID controller accomplished its mission. Record the “Heart Rate Logger” into a figure.