Project 6: Perfect 10

Due: Nov 29, 2018, 12:00PM

The goal of this project is very simple: control your character such that it can safely get across the gap and land on the moving platform. Your motion will be judged based on the time and the total torque it takes to get across.

Details

To achieve the task, you need to figure out a way to gain some momentum when the character is hanging from the bar. Once the momentum is enough, you have to decide the best time to release given the vision input (details below). You don't need to worry about whether the character is in a balanced standing pose at the end. However, if you can get it to balance, you will get some extra credits. Warning: Landing with a balanced state is not easy.

The skeleton code contains a controller that successfully gets the character to the bar. It gives you an example code for a standing balance control using stable PD and ankle strategy. It also provides a jump controller using virtual forces. The grasping in this project is modeled by enforcing a constraint between the hand and the object. You can simply use functions, leftHandGrab(), rightHandGrab(), leftHandRelease(), rightHandRelease(), whenever you want the character to grab or release an object. When you call the grab functions, a weldJointConstraint will set between the hand and the object, if the object is in contact with the hand. Otherwise, nothing will happen. Your job begins at the moment when the character grabs onto the bar. That is, you probably only need to change the code in the Controller::swing() function.

You can design the controller using any method we learned in the class. For example, you can manually design a controller by trial-and-error, you can frame it as a reinforcement learning problem with vision input, or you can do something completely different. It's up to you. It's ok to use any existing library for this project.

The motion of the character must be the result of simulation from DART under gravity (0, -9,81, 0).

The character can use any amount of internal forces on the actuated degrees of freedom, but is not allowed to apply any internal forces on the 6 global degrees of freedom. You are not allowed to "grab" the platform.

You are not allowed to change the default platform properties in the skeleton code.

You are allowed to query the state of the character, but not the state of the platform. Instead, a vision input, stored in MyWindow::mInputSensor, will be available to you. This vector stores the current image seen from a camera on top of the bar. The pixel values (RGBA) from the bottom left corner to the upper right corner of the screen are stored sequentially in mInputSensor. You can hit key 'd' to dump mInputSensor to PNG image files in the bin directory (the same directory where your executable is). This will give you an idea on what your character is seeing.

You will see two numbers on the screen when you start the simulation. The top one records the current accumulated joint torque, while the bottom number indicates the elapsed time since the simulation starts.

Your will present your result at the final project presentation on Nov 29 (details about presentation will be provided later).

After testing with the default platform in the skeleton code, your character will be challenged with a new platform he/she never seen before during the presentation. This new platform will have the same dimensions, moving on the same plane, but with different speeds.

You can choose to work by yourself or with a partner on this project.

Skeleton code

This project will run in DART. To start, place the swing folder in dart/apps, place swing.skel in dart/data/skel, and place ContactConstraint.cpp in dart/dart/constraint.

Then, delete the dart/build directory and re-run CMAKE.

UI control provided by the skeleton code:

space bar: simulation on/off

'p': playback/stop
'[' and ']': play one frame backward and forward
'v': visualization on/off
‘d’: dump images in the executable directory
‘m’: release the bar
Left click: rotate camera.
Right click: pan camera.
Shift + Left click: zoom camera.