

Documentation of Code Base Organization

Root Directory (UPennRSDev):

At the root directory, this code base has been structured to accommodate a full autonomous robotic system (vision, locomotion, localization). In addition, a testing suite has been developed, which allows for conducting tests on the robot through MATLAB.

You need only concern yourself with the LocomotionDev directory, which contains all locomotion related code, and the TestingSuite, which you will use often to unit test your code.

VisionDev and LocalizationDev are currently empty directories, but have been included with the plan to transition those components of our old code base into this new code base.

LocomotionDev:

LocomotionDev has been organized into the Modules directory, and Papers to Read. As it sounds, Papers to Read contains locomotion related papers that provide the majority of the theory behind bipedal locomotion and this code base. The Modules directory contains all of the locomotion related code.

Modules:

There are three main components / directories to Modules (Kinematics, Representations, StepHandler).

Kinematics:

Kinematics include forward kinematics and inverse kinematics. Forward kinematics takes as input a series of joint angles, and returns the position / orientation of the end effector. As an example, you can call the method forwardArmR, which takes as input the joint angles between the torso and the right hand. This method returns the position and orientation of the hand relative to the torso. End effectors of the Nao robot include hands, feet, and the head.

Inverse Kinematics takes as input the position / orientation of an end effector and returns the joint angles between the torso and said end effector to realize the desired position / orientation.

Representations:

The representations directory contains classes that have been created for the purpose of organizing information. These classes include Point, Transform, Trajectory, and COMContainer.

Transform:

The transform class is a convenient way of storing position / orientation, characterized by translation and rotation along the x, y, z axes (6 components).

Although you can consider the Transform class a black box, this information is actually stored in a 4x4 matrix, which allows for performing matrix operations to manipulate the Transform object. As an example, you can multiply two Transform matrices / objects in order to compose the two transformations into a single transformation. If you would like to read more about transforms, I highly recommend this resource:

http://www.codinglabs.net/article_world_view_projection_matrix.aspx

Trajectory:

During a single footstep, we are concerned with the position / orientation of the center of mass (COM) and swing foot relative to the planted foot (the foot contacting the ground). A trajectory stores this information as two transforms (COMTransform, swingTransform). A trajectory is relevant to a single instant in time. The walking motion of the robot is composed of a sequence of trajectories, one for every 10 ms in time.

Point:

The point class stores all information pertaining to a single footstep. This includes the position / orientation of the foot step (relative to the previous footstep). A point object also stores a sequence of trajectories pertaining to this footstep, which characterizes the motion of the robot for a single step.

COMContainer:

The center of mass motion of the robot is characterized by two position equations, one for x position (horizontal position) and one for y position (forward/backward position).

$$x(t) = x_0 \cdot \cosh(k \cdot t) + \dot{x}_0 \cdot \frac{1}{k} \cdot \sinh(k \cdot t) \quad (1)$$

$$\dot{x}(t) = x_0 \cdot k \cdot \sinh(k \cdot t) + \dot{x}_0 \cdot \cosh(k \cdot t) \quad (2)$$

The constants / parameters in this equation are stored in a COMContainer object, which pertains to a single footstep and is therefore stored within a point object.

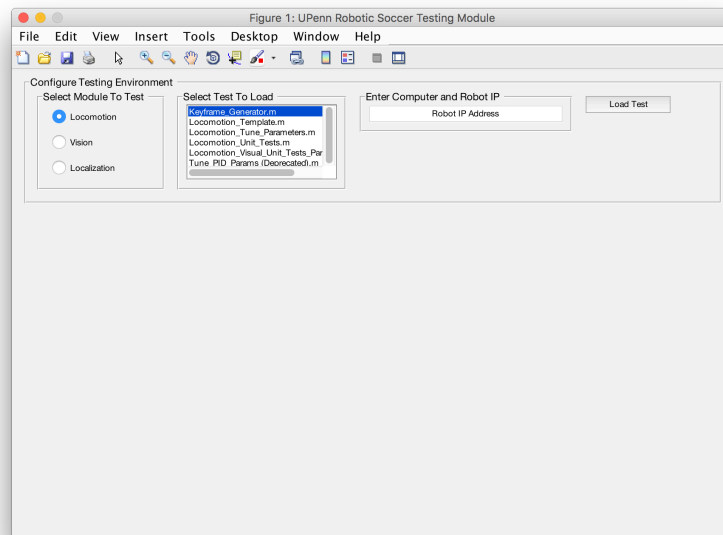
TestingSuite:

Within the testing suite directory, you should be aware of the TestingModule.m file as well as the TestingModules directory.

TestingModule.m:

TestingModule. is the main testing script, which can be run through MATLAB. Through this script, you can run all locomotion, vision, and localization unit tests, which will be executed on the robot. The test results are returned to your computer and presented through this script.

You can select unit tests from either locomotion, vision, or localization. After selecting a specific unit test and entering the robot's IP address, you can load and execute that test.



TestingModules:

All unit tests are stored in this directory. More specifically, they are stored within the relevant Locomotion/Vision/Localization subdirectory.

Once stored here, the unit test is automatically loaded into the TestingModule.m script and is ready for use.

There are template unit tests available within UPennRSDev/TestingSuite/TestingModules/Locomotion that you can reference in creation of a new unit test.