Kinematic Snake Simulation and Hardware Interface

# Simulation

### Setting the Parameters:

* The model parameters can be set in the *robot\_configuration\_data.m* script.
* The orbit that is generated by the Orbit Gait Generator is set in the *orbit\_config.m* script. The orbit and its parameters are stored in the *orbit* structure.
* If the model is open, the scripts need to be executed once after changes have been made.

### Start Model

To set up the simulink model for simulation run: *init\_controller; controller; configure\_controller\_simulation;* in the command window. That will:

1. Run *init\_controller.m*
2. Open *controller.slx*
3. Run *configure\_controller\_simulation.m*

### Changes in the Model

Use the manual switches to set the desired gait generation method. To generate gaits, the *t* input of the generators must be set to the ramp. Setting the input to zero will keep the initial position of the gait.

There are some parameters of the gait generation and control, that can be changed directly in the simulink model via the edit fields:

* The Gait Start Time specifies at when the Gait Generators start to output the desired gait until then they output the initial position and zero velocity and acceleration. It is initially set to 15 s because of the hardware, but for software it can be reduced.
* The Gait Time factor sets the slope of the time input into the orbit gait generator. That allows for slowing down the desired trajectory.
* The Trajectory controller is turned on with the sliding switch. When off *theta\_d* is set to *q\_d.*
* The Controller Start Time sets at what time the controller is switched on. Initially this is a second before the Gait Start Time, for simulation it may be adjusted accordingly.
* The proportional and derivative gains for the PD+ controller can be set for each joint individually.
* The Velocity and Acceleration Fade Time can be used to slowly increase the desired velocity and acceleration passed to the controller to avoid large discontinuities. The desired velocity and acceleration are scaled by a factor that linearly increases from 0.1 to 1 over the given time. (Initially zero, which means this doesn’t have any effect.)
* The damping coefficient for the model joint can also be set.

The Joint speed limiter is set to limit the *theta\_d* sent to the motors. The limit is initially set to 1 rad/s for each motor and is increased to 6. These values can be set by changing the constants.

### Evaluating the Simulation

To process and plot the results, run *process\_simout* from the command line. This asks you to save the results in a file. To just plot the results, cancel the save operation.

If you have processed the results there will be a *topic\_snake\_states* variable in the workspace. To plot these results again, run *plotResults.*

To plot saved results, load the *topic\_snake\_states* variable from the .mat file and run plotResults.

# Hardware

## First time setup:

* Clone <https://rmc-github.robotic.dlr.de/quadruped/controller_quadruped.git> and checkout the snake branch
* Run *cissy deploy*

## Startup:

* Turn on the Realtime and the Powersupply (Powersupply Vmax = 14 V Amax = 3A)
* Connect the servos via the USB hub
* Run *./start.sh* in controller quadruped (that should start the ln manager)
* In the manager start *servo\_rk* (should become ready)
  + If there is a deamon error start a deamon on the mobileproxy: *ssh rmc-mobilproxy.robotic.dlr.de /volume/software/common/packages/links\_and\_nodes/0.13.16/bin/sled11-x86\_64-gcc4.x/ln\_daemon -d*
* Start *matlab\_2018b guihost*  to open matlab and the simulink model
* Change *controller* as required and save (See [Changes in the Model](#_Changes_in_the))
* Start *matlab\_2018b conanfile*  and run *rtwbuild(‘controller’)* to build the controller
* After building you can sync the controller manually to the realtime with *sync\_simulink* (should print that something changed)
* Start *lnrecorder\_idle*  (not sure what this does or if it's necessary but do it)
* Put the snake in the straight configuration (0,0) and *reset\_servo\_positions*

## Running an experiment:

* Start *controller* in the manager
* Connect guihost matlab if necessary
* To log the data start *log all* (or another logger, but no guarantee those work)
* Start camera recording
* Watch snake do stuff
* Stop camera recording
* Stop *log all*
* Stop *controller* in ln manager or stop controller in guihosts simulink
* Run *convert\_log\_file* (if it fails because ‘no file’, you must wait a second)
* Run *process*\_*simout* in matlab (save data to desired location)

## Other Hints:

* Changes in the edit field in the simulink model and the workspace variables are not synced to the build host. To change the orbit or variables from robot\_configuration\_data, change them in the script, save it and run the script both on the guihost and the buildhost. Alternatively, just change it on the guihost and connect the realtime via simulink to transfer the changes during the run.