

Quick Start:

The search for gaits involves modified versions of the simulation functions designed to search through range of parameters.

Running the Simulation

The *main.m* script file contains sample code for running the simulation using the primary function *monkeySim*. Modifications to the controller can be done in the *controlAlgorithm* function.

Gait Finding:

3D Plot for Error Values

The *search.m* script file contains everything necessary to iterate over a range of values and create 3D plots of the fitness function.

Modify the values for the range/arrays *dt1_rng* and *dt2_rng*. The script will iterate over all combinations of values, and store the error values in the *results* array. Everything is saved to a time-stamped *.mat* file, and the square root of the values is plotted in 3D automatically.

Minimization/Optimization of Initial Values

The *minimize.m* script file contains the procedure necessary for finding and evaluating gaits.

Modify the seed values for the minimization: *dt1_guess*, *dt2_guess*. The script will use the Matlab function *fminsearch* to find the optimal solution. This solution is then iterated (and shown visually) for several cycles. Results for error value, kinetic energy, and, vertical height, are plotted.

Gait-Search Specific Code

The main component in the above two scripts is a modified version of the simulation function, called *runSim_fitness*. This function runs the simulation only through the steps of a single gait (swing up, impact, motor impulse, and finish at vertical). At the end of the single gait, it returns the error value based on the difference from initial conditions. The 3D plot uses *runSim_fitness* for every point, while the optimization script minimizes it.

This fitness simulation also uses a modified version of the controller, *controlAlgorithm_search*, that acts depending on the state of the robot. It swaps the magnets at when the second link has reached the horizontal ($\theta_2 = \pi - 2\theta_1$), and impulses the motor until pre-impact energy matches post-impact energy. The timings of these events are recorded in global variables, which are obtained later by the script file as necessary.

A second modified simulation function, *runSim_stability* runs the robot through several iterations using the standard open-loop controller. It calculates some potentially useful values at each iteration.

Modifying the Search / Looking for different types of Gaits

Modifying the search or looking for different types of gaits involves four main steps:

1. Change the search controller, *controlAlgorithm_search*, to act and record timings as necessary for the gait.
2. Change the fitness simulator, *runSim_fitness*, to stop integration at the right point, and return the correct fitness value.
3. Change the actual control algorithm, *controlAlgorithm*, to reflect the modified gait type for simulating the gait after it is discovered.
4. Modify the script files *search.m*, *minimize.m* as necessary to use new gait parameters / controller timings.