Dynamic Optimization Hw 2

Christopher Hazard and Jaskaran Grover

Part 1

The walking controller for the case of no disturbances is shown in the video part1optimized.mp4. This was done using the cma-es c code from Nikolaus Hansen's website. The code for cma-es is in cmaes.c and cmaes-interface.c, while the code to run it for the simulation is in simulate.c. Starting from the provided initial seed p0, the controller was evolved over 20 generations with 50 population members per generation with the objective function provided (the same objective function from Jack Wang's papers). We scaled each of the variables between -1.0 and 1.0 prior to entering them into the CMA optimization (and unscaled the sampled values returned by CMA to convert them back to meaningful parameters), and set an initial standard deviation of .1 for each of the variables we optimized over. The original seed (p0) was:

swing_time 0.89 opt end

thrust1 -0.047 opt end

swing_hip_target 0.33 opt end

swing_hv1 -0.37 opt end

swing_ha1 -0.10 opt end

swing_knee1 -0.43 opt end

swing_kv1 0.29 opt end

swing_ka1 0.26 opt end

swing_knee_target -0.080 opt end

swing_kv2 0.11 opt end

swing_ka2 -0.14 opt end

stance_hip_target -0.22 opt end

stance_hv1 -0.031 opt end

stance_ha1 0.28 opt end

pitch_d 0.085 opt end

stance_kv1 0.41 opt end

stance_ka1 -0.38 opt end

stance_knee_target -0.046 opt end

stance_kv2 -0.29 opt end

stance_ka2 -0.36 opt end

stance_ankle_torque -0.12 opt end

The evolved parameters are (saved in part1optimizedparams and animated in part1optimizedanimation):

swing_time   0.79282696319790968431 opt end

thrust1 -0.016324750164846468792 opt end

swing_hip_target   0.19564731929818846012 opt end

swing_hv1   -0.3960030047124435959 opt end

swing_ha1   -0.10552955124003893239 opt end

swing_knee1   -0.44837865084125017034 opt end

swing_kv1   0.28144472909365148006 opt end

swing_ka1   0.30837610797348369296 opt end

swing_knee_target 0.0092944016302201060142 opt end

swing_kv2 -0.035826707073682462212 opt end

swing_ka2   -0.22091277358902677852 opt end

stance_hip_target   -0.16589008932902554738 opt end

stance_hv1   0.054798718522000022801 opt end

stance_ha1   0.39774053139798160039 opt end

pitch_d   0.0389586804698438538 opt end

stance_kv1   0.37068635466441901549 opt end

stance_ka1   -0.43685614972883574092 opt end

stance_knee_target   0.046175185191842160748 opt end

stance_kv2   -0.22342696789794125323 opt end

stance_ka2   -0.28379722279676622421 opt end

stance_ankle_torque -0.022461064171331257544 opt end

The animation for part 1 is in part1optimized.mp4

Part 2

Applying a quick test to the parameters optimized in part 1 shows that the controller is not robust to external disturbances. For example: if you apply 1 newton of force continuously throughout the entire simulation, the walker falls over. Applying larger forces (~10 newtons) over time periods of .1 seconds also causes it to fall down. To make it more robust, we reoptimized using the seed generated from part 1 with an initial standard deviation of .03 for each variable, a population size of 75, and 25 generations. To reduce sensitivity on the variables thrust1,swing_hip_target, swing_knee_target,stance_hv1, pitch_d, stance_knee_target we rescaled them between 0 and .1 (it is fine if they go below 0 or above .1 since these parameters are just for rescaling purposes, they don't actually mean anything). The other parameters were scaled between 0 and 1.

The simulation used to evolve the parameters (starting from the seed in part 1) consisted of applying increasingly stronger forces in each direction throughout the 20 second simulation, all the way through a user-specified maximum force. At times `{2.5,5.5,8.5,11.5,14.5,17.5}`, a force was applied for .1 seconds on the walker. With a maximum force of 25 newtons, the forces applied at each of the times were `{5.0,-5.0,15.0,-15.0,25.0,-25.0}`. If at any time the walker were to fall down, the penalty for crashing in the objective function was already set high enough to make it clearly inferior to a controller that was able to walk for longer. The objective function essentially ranked each of the controllers by how long the simulation lasted, and to differentiate between controllers that crashed at the same point, the other energy minimization terms would come into play to pick the more efficient walker (so robustness matters more than efficiency).

The resulting parameters were:
swing_time 0.70821711069881920775 opt end
thrust1 -0.01455242342778002157 opt end
swing_hip_target 0.19217030429174183914 opt end
swing_hv1 -0.42267642896031326627 opt end
swing_ha1 -0.11534046315631402146 opt end
swing_knee1 -0.49421175094760022573 opt end
swing_kv1 0.29955307102539202591 opt end
swing_ka1 0.30082089426000269139 opt end
swing_knee_target 0.0062210791753515744251 opt end
swing_kv2 -0.062785784497098470758 opt end
swing_ka2 -0.27250360571328069437 opt end
stance_hip_target -0.16255034712180027601 opt end
stance_hv1 0.056728915104983096418 opt end
stance_ha1 0.4343952840271997351 opt end
pitch_d 0.039454090428163626769 opt end
stance_kv1 0.35288690031035074846 opt end
stance_ka1 -0.39005769390127431606 opt end
stance_knee_target 0.047956665386955373009 opt end
stance_kv2 -0.2311056155155337688 opt end
stance_ka2 -0.23002258773071751485 opt end
stance_ankle_torque -0.029664223954954341278 opt end

The accompanying videos show the evolved controller under several test conditions:

10n5s6trials: maximum force applied was 10 newtons, so in the simulation scheme above, at the times `{2.5,5.5,8.5,11.5,14.5,17.5},` forces of {2,-2,6,-6,10,-10} were applied for .5 seconds each. It managed to survive

20n1s6trials: max force 20 newtons applied for .1 seconds, with forces {4,-4,12,-12,20,-20} applied at each of the key points in time—successful

25n1s6trialsfailed: max force 25 newtons applied for .1 seconds, with forces {5,-5,15,-15,25,-25}---crashed on the 25 newton force

40n05s6trialsuccess: max force 40 newtons applied for .05 seconds, with forces {8,-8,24,-24,40,-40} --success

50n05s6trialfailed: max force 50 newtons applied for .05 seconds----crashed at 50 newtons