CHAPTER V

Crawl Gait Results

To test the efficacy of the Projected Profile crawling gait, a gait sequence was generated using MATLAB and simulated using the V-REP simulator by Coppelia Robotics. The crawling sequence was generated using the nominal crawling parameters. The Nao humanoid platform was then programmed using the NaoQI API in C++ to use this sequence to execute a crawling action. The robot was initialized to its crawling pose and placed on the ground where it crawled under a short ledge. During this experiment, the joint angles and joint currents were recorded for later analysis using the NaoQI API. To demonstrate a context under which the gait would be used, the Nao was then programmed to walk to a fiducial marker, position itself into the crawling position, crawl under a short ledge, and then return to a sitting position. NaoQI provides functions for the Nao to walk, position itself to a set of poses from any initial pose, and track a "red ball". The fiducial marker was a red circle on a piece of paper. The robot crawled under the ledge by executing a finite number of crawling sequences, before moving to a sitting position. A video of this experiment was recorded for later analysis.

Following this, the optimized crawling parameters were used to generate a new gait sequence in MATLAB. The optimized and nominal gait sequences were tested using the V-REP simulator and the simulated joint torques were recorded. The V-REP simulator provides a MATLAB API to record these torques. While the optimized gait sequence was formulated using the pseduo-static assumption, the gait was tested at different speeds to compare the increase in efficiency against the nominal gait for varying degrees of dynamic loading.

Details about the crawling environments, simulation, and crawling parameters can be found in Section 5.1. Data collected about the nominal crawling experiment is presented in Section 5.2 and the optimized crawling experiment in Section 5.3.

5.1 Experimental Setup

The crawl gaits were testing on the Nao humanoid using the V-REP simulator in MATLAB, and by having actual robot crawl under a small ledge using the NaoQI API in C++. The nominal parameters were tested both in simulation and on the actual robot, while the optimal parameters were only tested in simulation. MATLAB has tools to use genetic algorithms to optimize systems, which were used here to generate the optimal parameter splines.

5.1.1 Mobile Platform

The Nao humanoid was used to test the Projected Profile algorithm. Details about the Nao are discussed in Chapter II, but it makes a convenient platform to test crawling algorithms as NaoQI provides an extensive API to control a range of parameters from individual joint angles to full body positioning. Importantly, the kinematic configuration of the robot is amenable to the crawling paradigm and its relatively small size allows environments to be easily constructed for testing.

5.1.2 Crawling Environments

Prior to any testing on the actual robot, the V-REP Simulator by Coppelia Robotics was used to verify that the algorithms functioned correctly and exhibited the desired behaviors. The V-REP simulator is a good choice for this application for a variety of reasons. It uses mature and well known Open Dynamics Engine (ODE) as its physics engine, supports multiple operating systems such as Windows and Linux, and provides an API for use with many languages such as C++, Python, and MATLAB. What's more, it provides a model of the Nao humanoid that can be easily commanded. Figure 22 shows an example of the Nao in V-REP set at the initial crawling position. The V-REP API also simulates various sensors, and allows the torque at each of the Nao's joints to be accessed. As mentioned in Chaper IV, this data was used to generate the gait-parameter-triplet-to-joint-torque mapping for the genetic optimizer. The simulated joint torques were also accessed while running the optimized crawling gait at different speeds to compare its improvement over the nominal gait. Results from those experiments are presented in Section 5.3.3.

While traversing rough terrain or over small obstacles is one application of a crawl gait, these experiments were instead designed to show that the robot could access areas with demanding height constraints. For this, a small table was used whose sides were

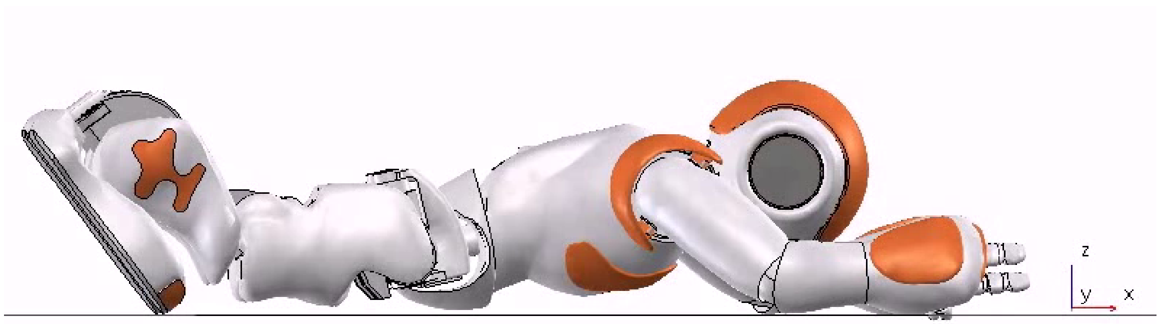

Figure 22: This figure shows the Nao humanoid V-REP simulation. The robot is positioned at the initial configuration of the close chain phase of the crawl gait.

Gait Parameter	**Value**
θ_3	$16.5°$
θ_4	$27.5°$
α	$-30°$ to $-90°$

Table 2: Table of gait parameters for the nominal crawl gait. θ_3 and θ_4 are held constant, while α is linearly decreased in proportion to time.

blocked with panels down to about $200mm$ off of the ground. Figure 23 shows the Nao crawling under this table, with its head almost touching the bottom of the panel. This represents the practical height constraint that the Nao can satisfy with this gait.

5.1.3 Gait Parameters

As discussed in Chapter **??**, the closed chain phase of the Projected Profile crawl gait can be parameterized on the three angles $[\theta_3, \theta_4, \alpha]$. These three variables are referred to as the gait parameters, or alternatively as the angle triplet. The nominal angle triplet can be seen in Table 2. They involve holding θ_3 and θ_4 constant, while varying α linearly as a function of time.

Results using these nominal gait parameters can be seen in Section 5.2.

In order to optimize the gait using these parameters, the procedure outlined in Chapter **??** was used, modeling the gait parameters as cubic splines. The Genetic Algorithm in MATLAB's Global Optimization Toolbox was used in conjunction with the V-REP torque table detailed in Chapter **??** to generate the optimal spline parameters. As the genetic algorithm cannot guarantee finding the global optima for any arbitrary function, the optimization was executed several times and the best spline parameters

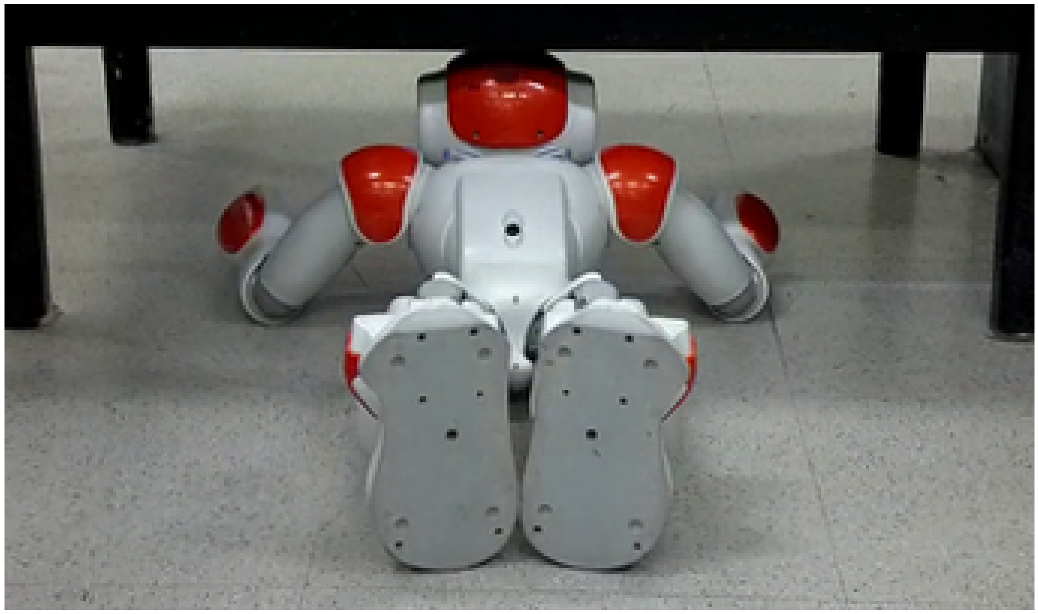

Figure 23: This figure shows the Nao as it crawls under the table with the low panels. The head of the robot can be seen to be nearly touching the panel, representing the lowest practical traversable height constraint.

were used in the experiment. Each optimization used 50 to 80 generations to converge on results, but as can be seen in Figure 24, after about 10 generations the optimization was very near the minima.

Table 3 shows the spine parameters resulting from the optimization procedure. Each gait parameter is now a cubic polynomial which is a function of time $[\theta_3(t), \theta_4(t), \alpha(t)]$. The coefficients from the table are used as $c_3t^3 + c_2t^2 + c_1t + c_0$. The c_0 coefficients are simply the angles corresponding to the starting pose for the closed chain phase of the gait. The splines are also constrained to end at the final pose for the closed chain phase, which correspond to the angle triplet of $[0.28798, 0.47997, -1.5710]$ for $[\theta_3, \theta_4, \alpha]$. The initial and final angle triplets are in radians.

Figure 25 shows the splines graphically, with the nominal gait parameters overlaid for comparison. While the optimized parameter trajectory for α is similar to its nominal trajectory, the trajectories for θ_3 and θ_4 dip significantly, with θ_4 having the most drastic deviation from the nominal.

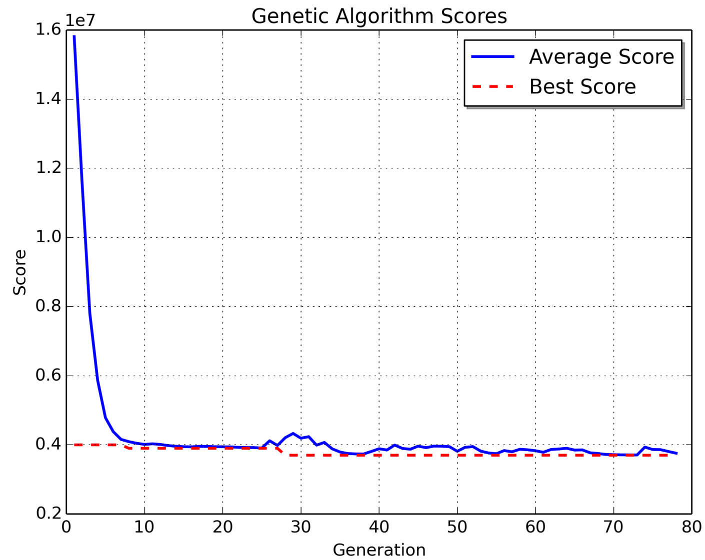

Figure 24: This figures shows one of the gait parameter optimization trials. As the genetic optimization procedure uses multiple childen, the plot shows the average score for the children and the score of the best child as the generations progress.

Gait Parameter	c_3	c_2	c_1	c_0
θ_3	0.2365	0.0893	-0.3267	0.28798
θ_4	1.8796	-0.1365	-1.7434	0.47997
α	-0.2134	1.1570	-1.9898	0.52360

Table 3: Table of gait parameter coefficients for the optimal crawl gait. The c_i coefficients are used in the cubic spline $c_3t^3 + c_2t^2 + c_1t + c_0$ to vary the gait parameters as a function of time. The c_0 coefficients are simply the initial starting angles for each parameter.

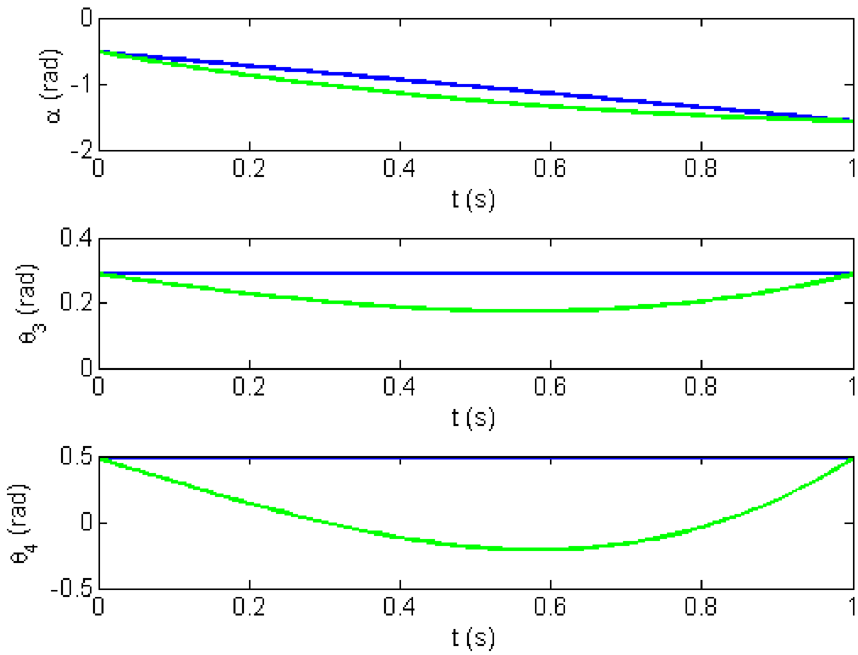

Figure 25: This figure shows the optimal gait parameter trajectories overlaid onto the nominal trajectories. The blue lines represent the nominal trajectories while the green lines show the optimal.

5.2 Nominal Crawl Gait Data

To test the Projected Profile gait using the nominal parameters reviewed in Section 5.1.3, the gait sequence was first generated in MATLAB and then simulated in V-REP. Following this the gait was tested on the Nao by setting it to crawl under a vertically constrained table in two different experiments. In the first, the robot was set to the initial crawling pose and placed on the floor in front of the table. The robot then executed the crawling gait. In the second experiment the robot was programmed to recognise a fiducial marker, represented by a red circle mounted to the table, and walk to it. The robot then moved to a prone posture and crawled under the table. This procedure is more throughly examined in Section 5.1. Joint angle and joint current data was collected during the table experiments and is presented in this section. These table experiments demonstate the efficacy of the gait to locomote the robot and the low profile nature of the gait to allow access to vertically constrained spaces.

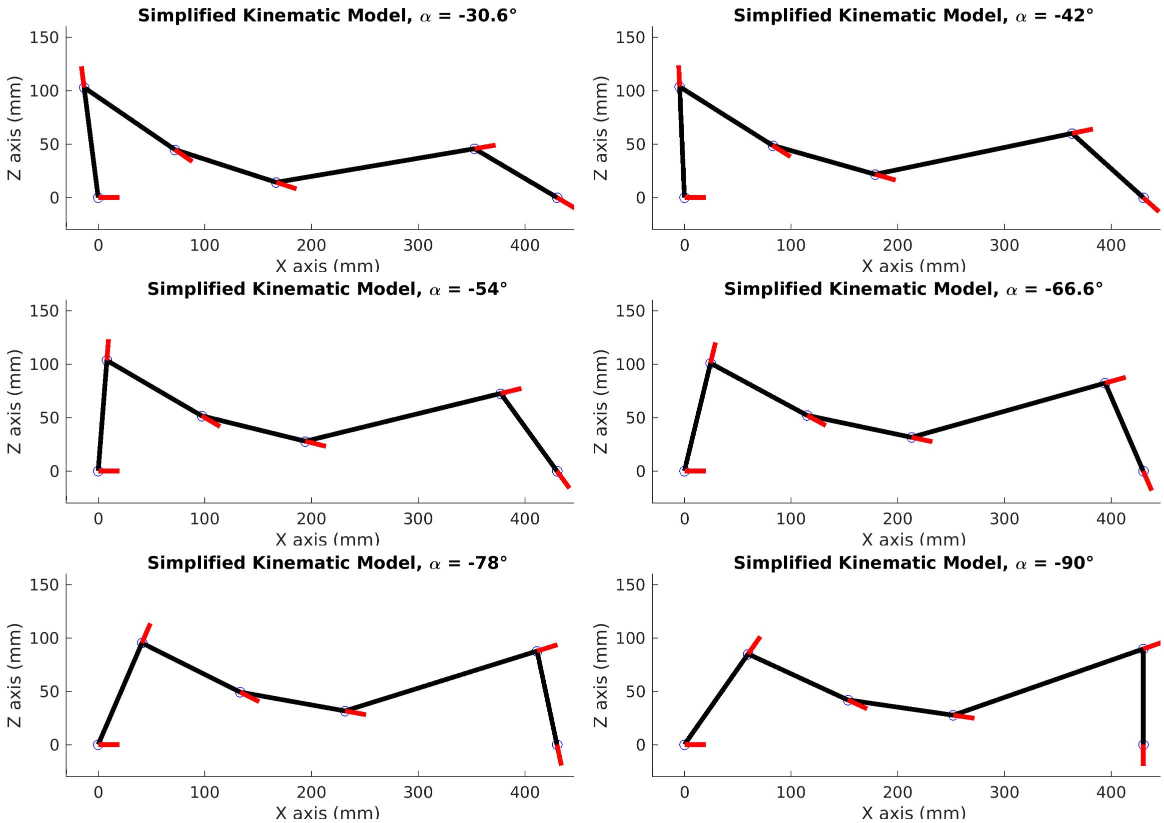

Figure 26: This figure shows a simplified kinematic model of the Nao as it executes the Projected Profile gait using nominal parameters. The model starts with $\alpha = -30°$ and terminates when $\alpha = -90°$.

5.2.1 Simulations

Figure 26 shows samples of the closed chain phase of the Projected Profile gait using the nominal parameters. It shows a simplified kinematic model representing a projection of the robot onto the saggital plane. The frames show the model starting in the inital pose, and by linearly increasing the α gait parameter and holding the θ_3 and θ_4 parameters constant, the model shifts forward until α has reached its terminal value. This places the model at the final closed chain pose. It can be seen the highest point of this gait occurs at about $z = 100mm$. This model of course does not include the limb thicknesses nor the head of the Nao, as it only models joint centers. This model was created using MATLAB and is used to view the results of the gait sequence generation.

Once the gait sequence is generated, it is tested using the V-REP simulation of the Nao humanoid. Figure 27 shows the simulated Nao executing the closed chain phase of the nominal crawl gait. As with the simplified kinematic model, the α gait parameter

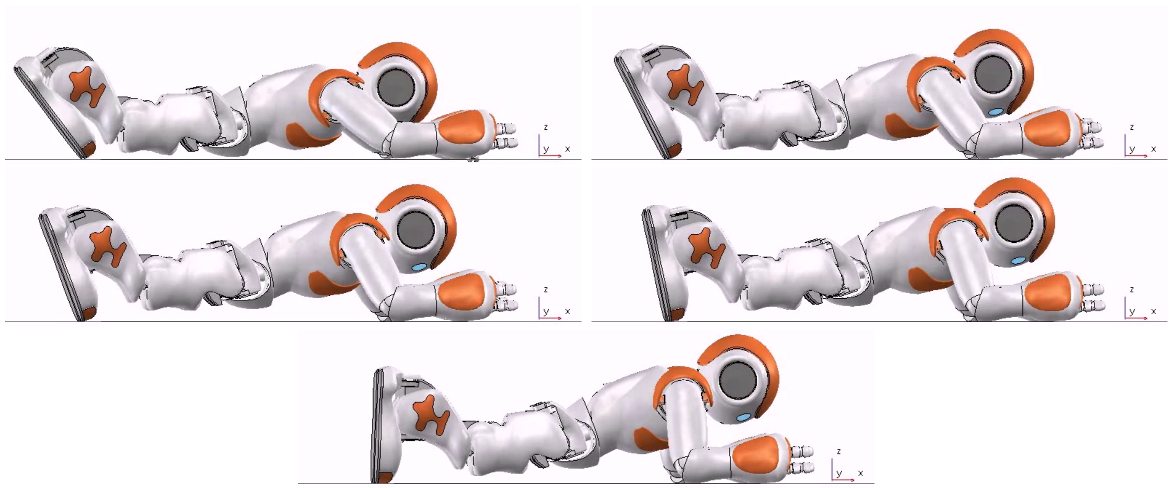

Figure 27: BLAH BLAH Simulated Nao with nominal gait.

is linearly increased from $-30°$ to $-90°$, which moves the robot forward. As detailed in Chapter **??**, the gait parameters and other constraints are used to position the arms of the robot, as unlike leg, hip, and shoulder pitch joint which have angles which directly correspond to joint angles in the simplified kinematic model, the shoulder roll and elbow joints do not. The head in this simulation can be seen to be the highest part of the robot throughout the majority of the gait, increasing the minimum value of the allowable vertical constraint.

5.2.2 Height Constrained Table

Following simulation, the Projected Profile crawl gait using the nominal parameters was tested on the Nao. Figure **??** shows the initial test of the robot crawling under the vertically constrained table. The Nao is set to the initial crawling pose on the floor outside of the table. The robot then executes the crawl gait for several sequences until the robot is under the table. For this experiment, the robot executed 9 sequences in 27 seconds. The robot travelled about 1 body lengths or $610mm$, equating to a velocity of about $22.6\frac{mm}{s}$.

Figure **??** and **??** show the second experiment performed with the vertically constrained table. The robot has detected and walked to the red marker affixed to the table. It then transitions to a prone posture and begins the crawling sequence. After having crawled to the other side, the robot transitions to a sitting posture. The ability to transition from posture to posture are provided through the NaoQI API. For this experiment, the robot executed 25 sequences in 58 seconds. The robot travelled about

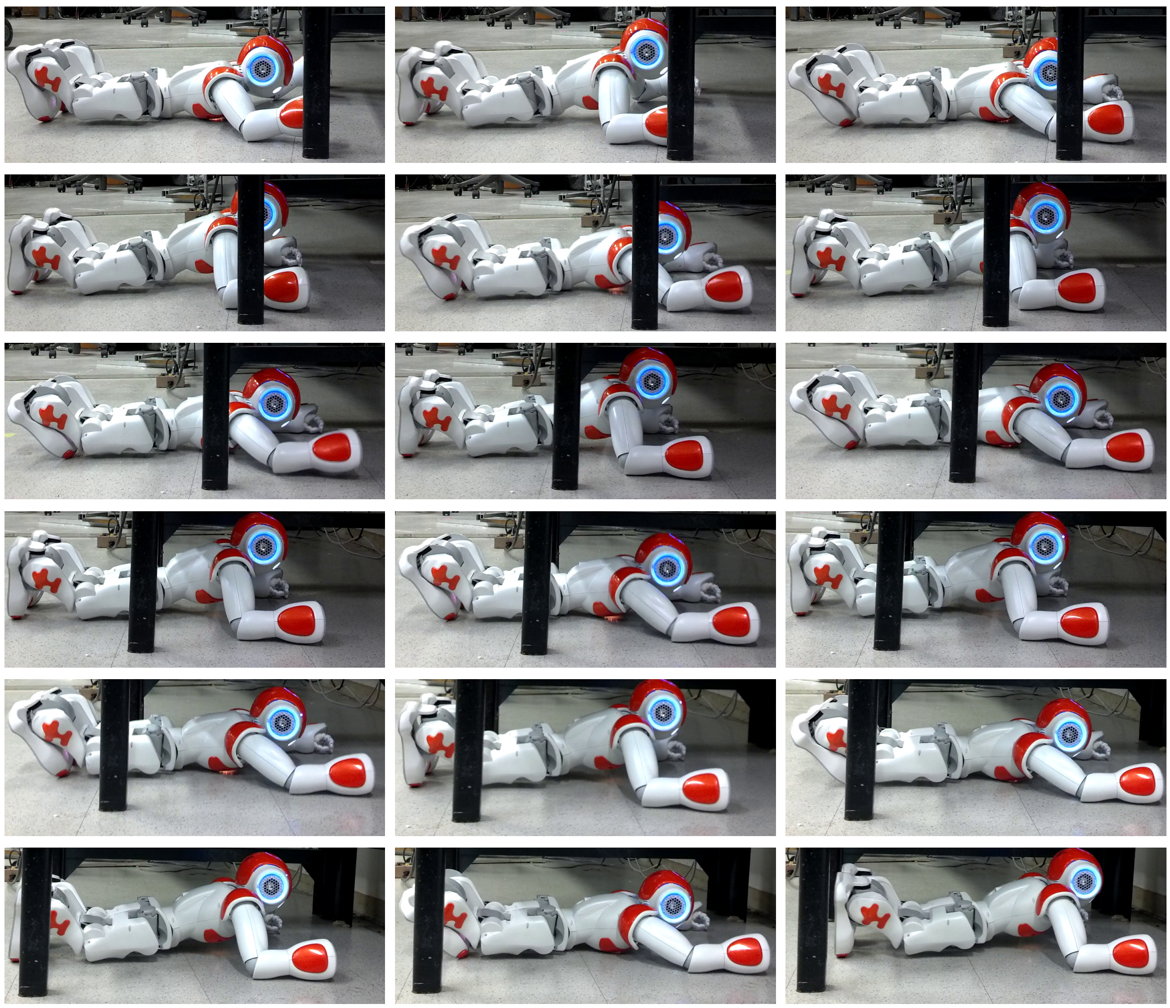

Figure 28: This figure shows the Nao crawling under the vertically constrained table.

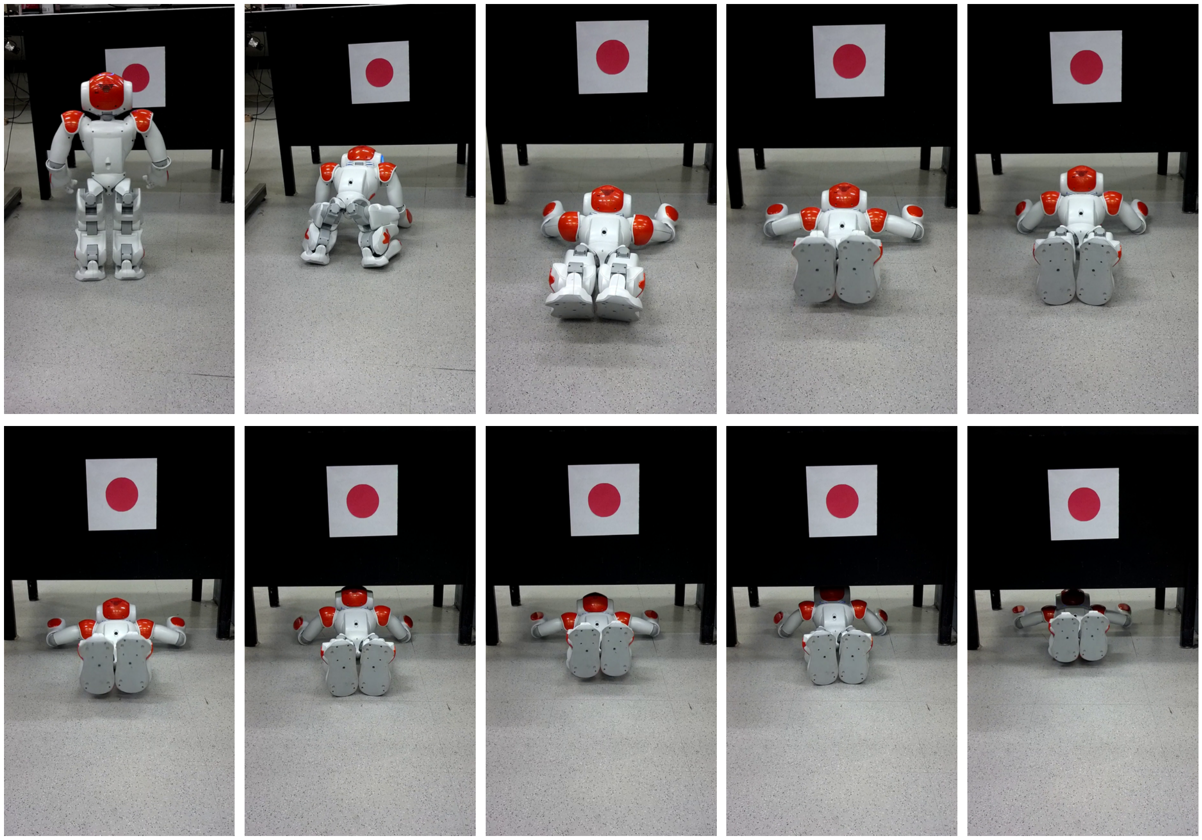

Figure 29: BLAH BLAH Approaching and crawling under an obstacle. A red dot is used as a marker for the direction in which the robot is commanded to move. When the robot approaches below a specified distance threshold from the obstacle, the crouch-down and crawl gait sequence is initiated.

2.75 body lengths or $1,676.4mm$, equating to a velocity of about $28.9\frac{mm}{s}$.

5.2.3 Joint Data

Here is a plot of the joint angles of the gait, showing its periodicity.

This is a full plot of the entire sequence, from walking, to crawling, to walking.

This plot shows the current draws of the arm and leg joints. There are some distinct patterns, but is hard to relate torques directly.

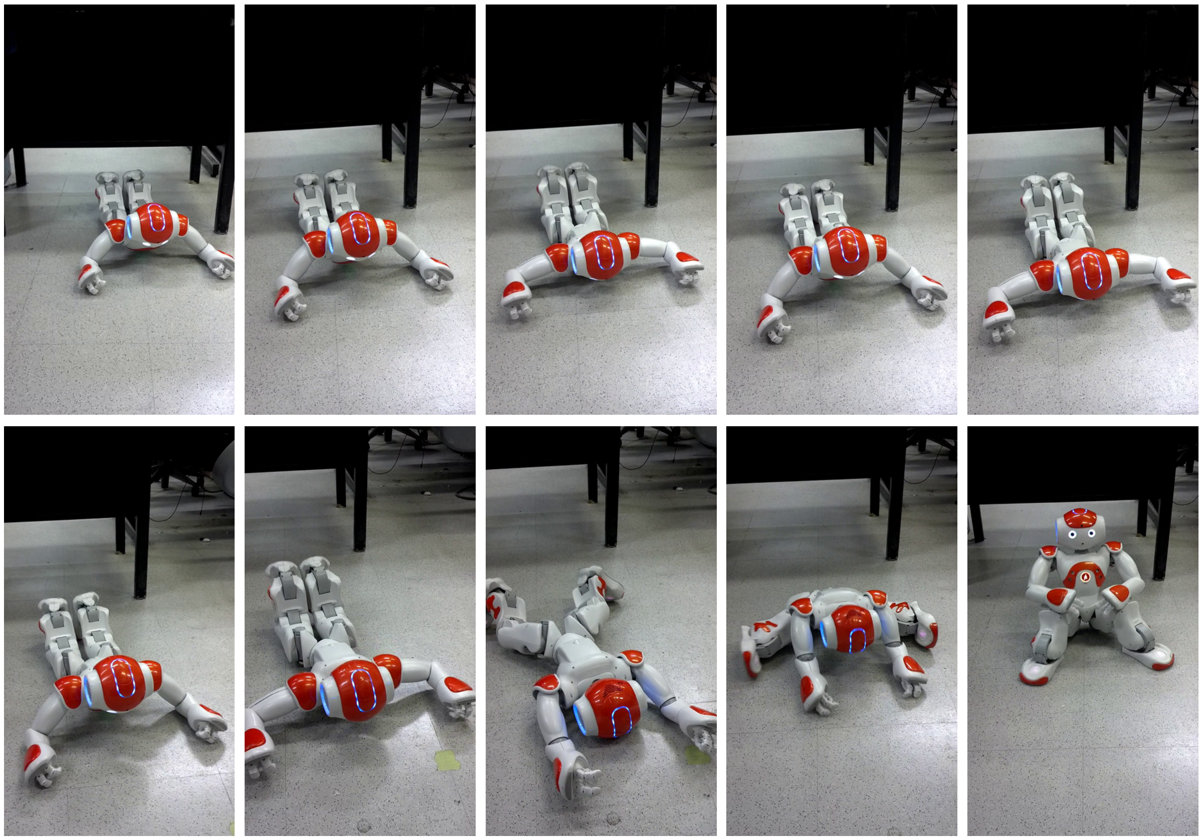

Figure 30: BLAH BLAH Crawling under an obstacle and transitioning back to stand posture.

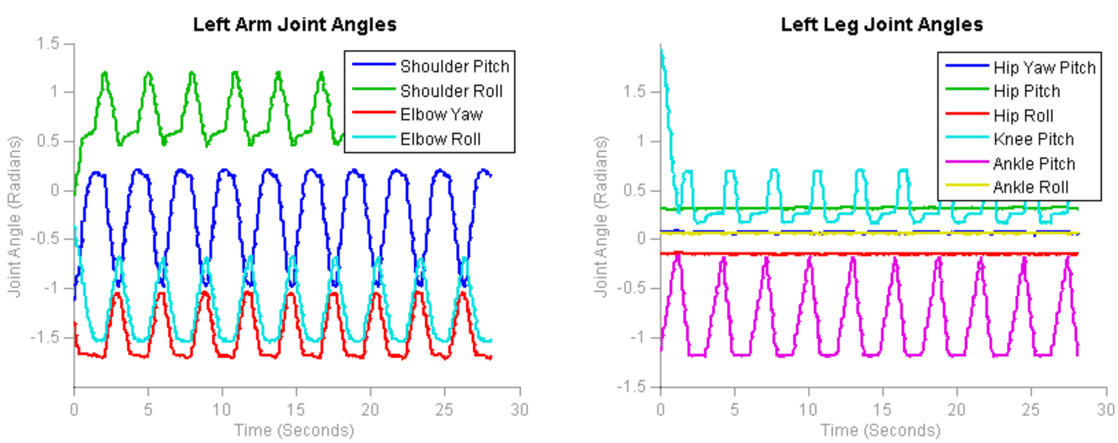

Figure 31: Measured motor joint angles during multiple iterations of the periodic crawling gait. While the crawling gait is laterally symmetric, the asymmetry in measured angles is due to the definitions of the robot frame and joint frame in the NAO API, which essentially forms a mirror asymmetry between the left and right joints.

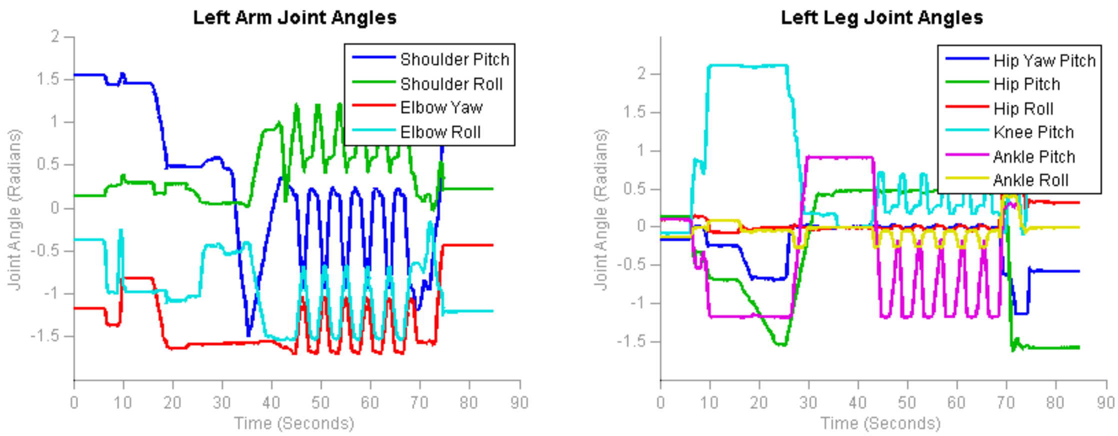

Figure 32: Measured joint angles for a sequence of transitioning from standing to crouch to crawling, crawling under a table, and then returning to crouch.

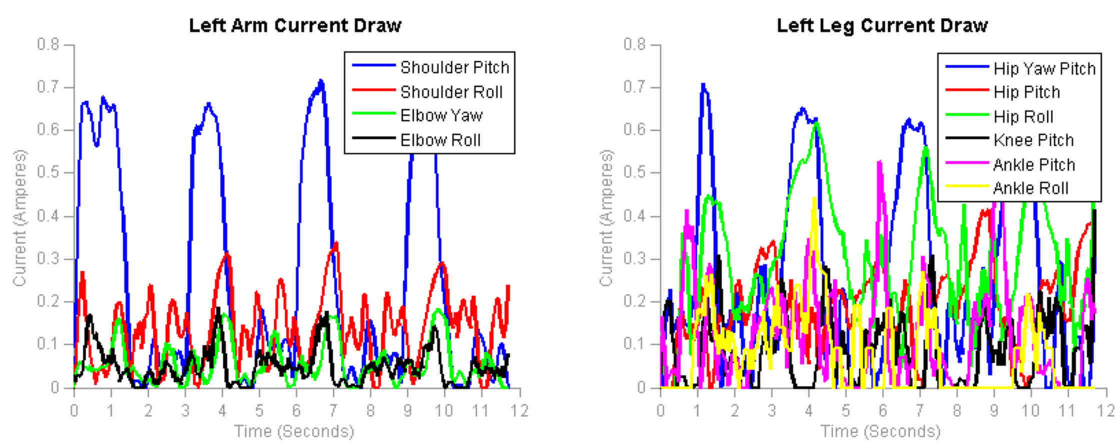

Figure 33: Measured motor current draws during multiple iterations of the periodic crawling gait.

5.3 Optimized Crawl Gait Data

This time, we optimized the gait. Here's what happened.

5.3.1 Optimization Cost

These plots show the theoretical optimization costs of the nominal and optimized triplet parameters.

5.3.2 Simulations

Here's the MATLAB simulation. The theta 3 and 4 are doing different things.

Here's the V-REP simulation. The knees and hips are doing different things. The arms are down the whole time.

5.3.3 Joint Torque Data

Ok, so now, how does it do with these new optimizations? We use the V-REP here instead of the currents from the robot because they are easier to interpret.

Here's the nominal and optimal joint data overlaid for each joint. It ranges, but you can kinda see the optimal ones look kinda like a compressed version of the nominal ones.

Here's the nominal and optimal joint data, side by side, on a duration basis. You can kinda see the compressions. You can also see that the short ones look similar, but the longer ones don't.

Actually summing up the costs you can see there's a difference between the two and the optimal one is always lower cost. Percentagewise, the 7 second gait had the biggest difference, but I'd say that anything after 7 seconds has probably converges and takes the most advantage of the optimization. These costs are for straight sum of torques. They're not weighted.

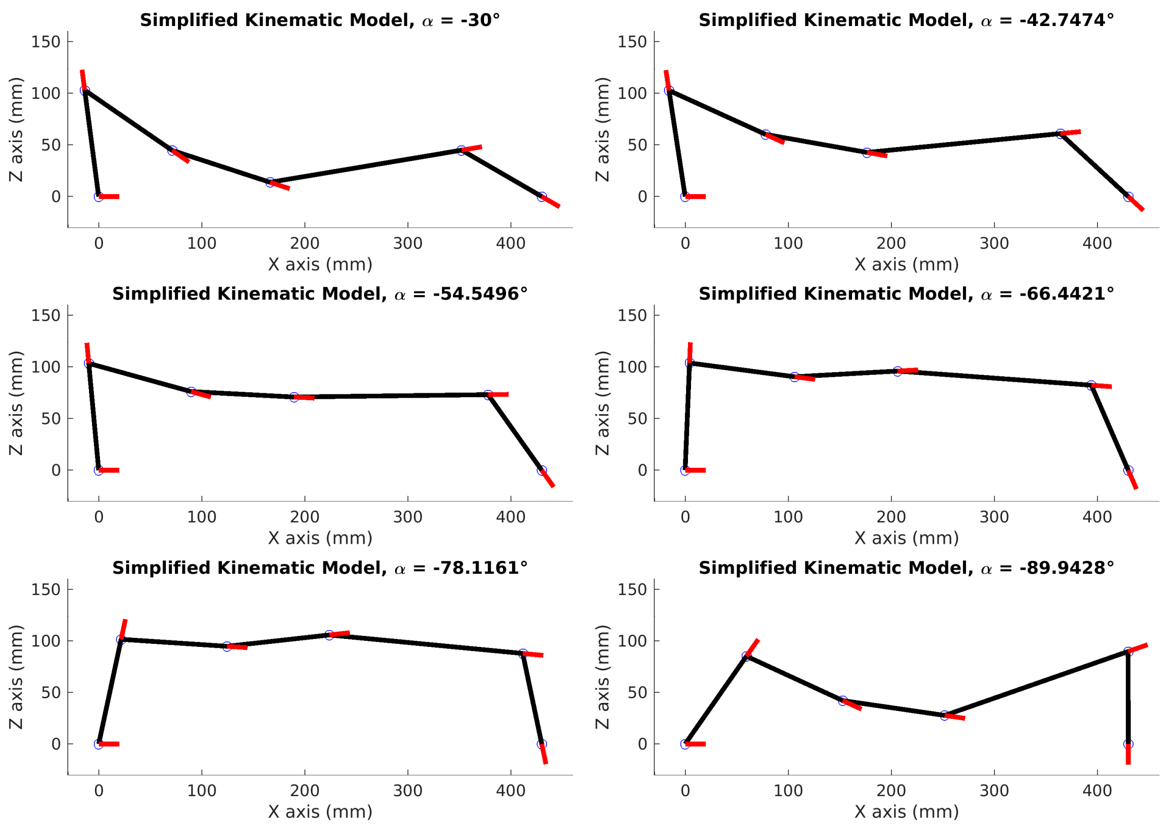

Figure 34: BLAH BLAH Projected Profile Nao with optimal gait.

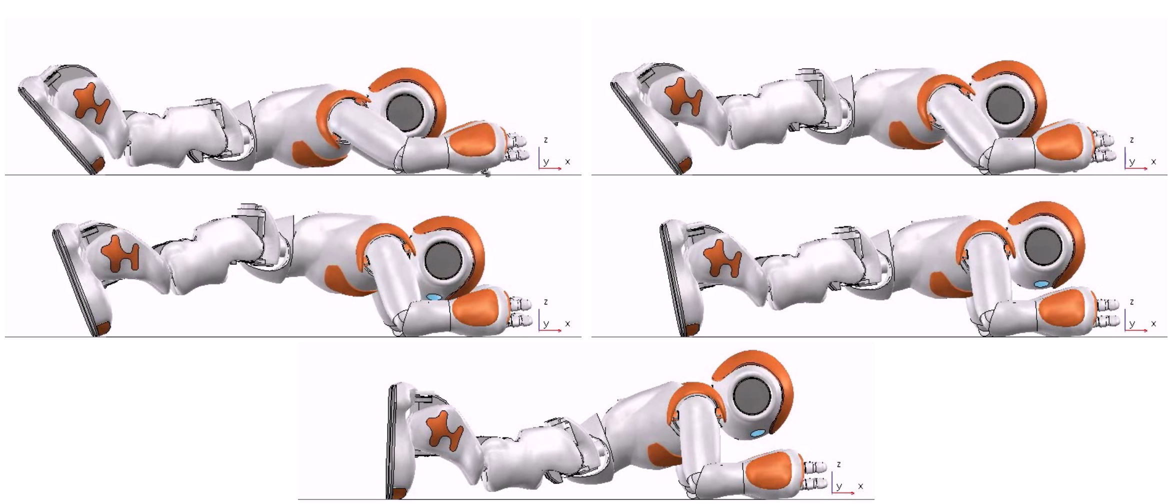

Figure 35: BLAH BLAH Simulated Nao with optimized gait.

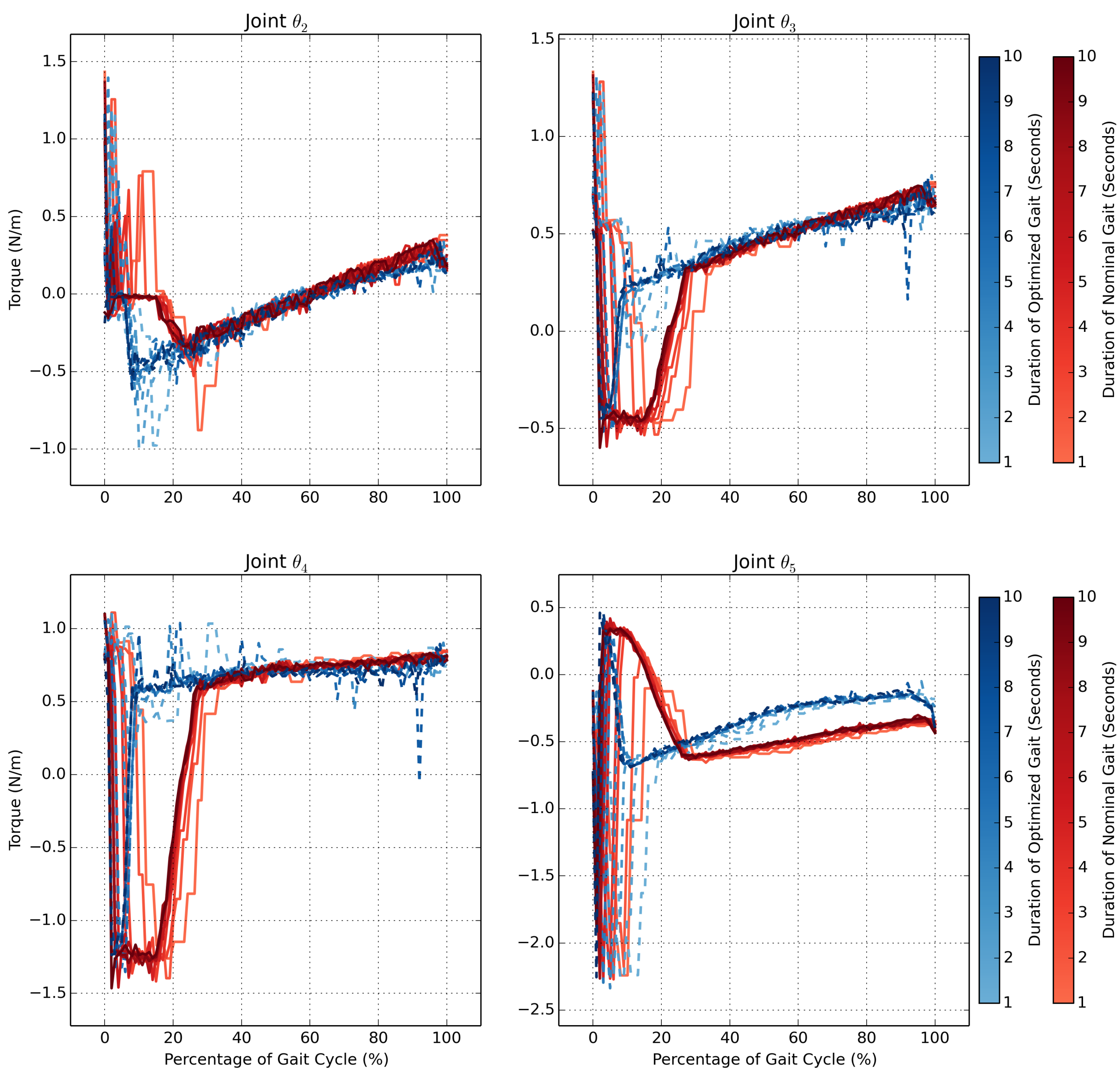

Figure 36: BLAH BLAH Simulated joint torques for the different times. Nominal and optimized.

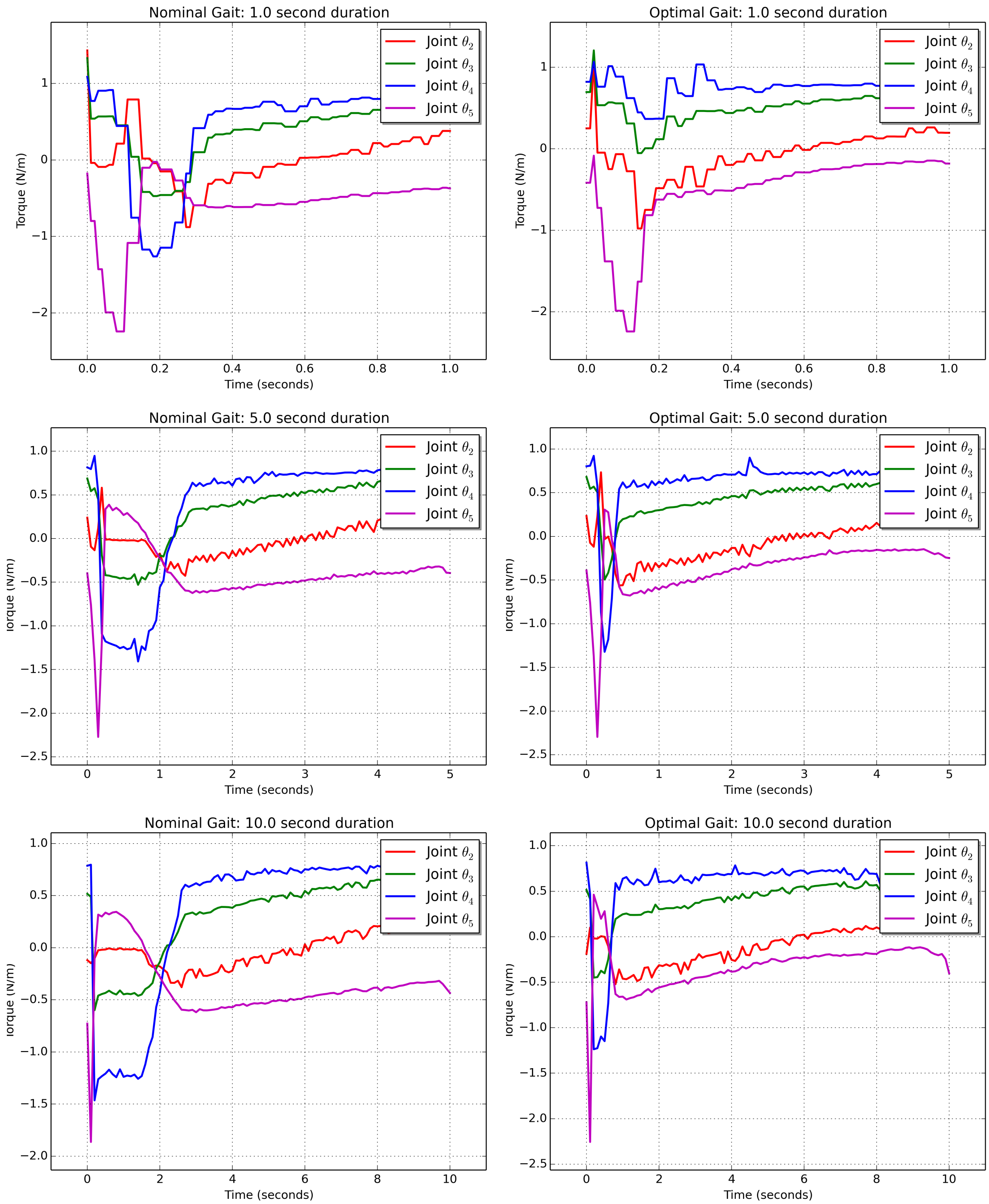

Figure 37: BLAH BLAH Simulated joint torques for the different times. Nominal and optimized.

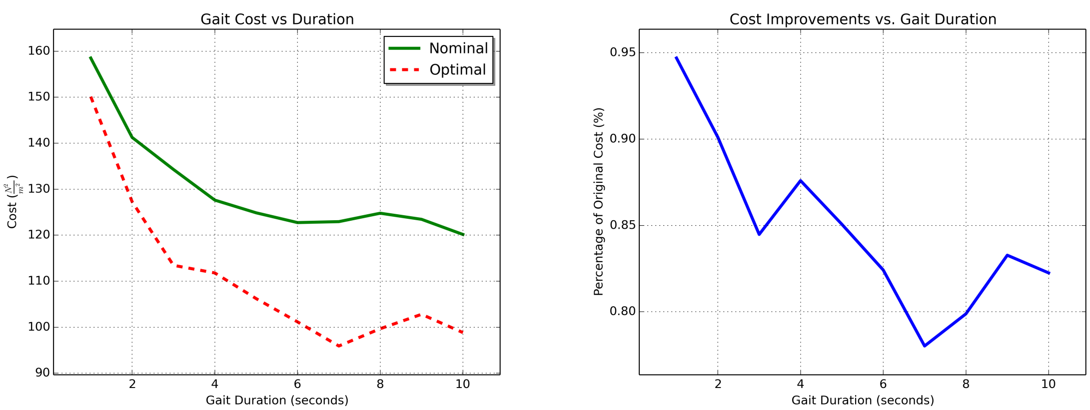

Figure 38: BLAH BLAH Things about gait cost comparisons.