ASU Ira A. Fulton
Schools of Engineering
ARIZONA STATE UNIVERSITY

MAE 598/494 DESIGN OPTIMIZATION (SPRING 2016)

OPTIMIZATION OF ADVANCED MECHANICAL LEG STRUCTURES

BY

Nathan Cahill
Peng Wei
Hope Yao
Hosain Bagheri
Michael Naso

ABSTRACT

Specialized mechanical designs are capable of decreasing the power cost in robot limbs that perform repetitive tasks. This paper outlines an approach to designing a robot limb for a specific repetitive task. The task that is being designed for here is legged gait. In this paper every aspect of this problem has been tackled and optimized. The geometry of an advanced hybrid leg mechanism has been optimized to reduce power required by 8.8%. Compliant parallel springs have been added to the design and can reduce the power required by upwards of 50%. Notably: this compliance optimization problem has been reduced to a quadratic programming problem that converges to the global optimum in just seconds. An analytical formula for the optimal gear ratio has been discovered that finds the optimal gear ratio for the motor by minimizing electrical losses due to armature resistance, while constraining the velocity and torque of the motor to within its continuous operating region. Finally, a method for finding optimal gait patterns has been developed using a dynamic simulator and a parameterization of the gait patterns.

INTRODUCTION

Specialized mechanical designs are capable of decreasing the power cost in robot limbs that perform repetitive tasks. This paper, outlines an approach to designing a robot limb for a specific repetitive task. The task that is being designed for here is legged gait.

Traditionally legged robots have been actuated in serial, with each leg freedom being controlled by a single actuator [1]–[4]. These robots tend to have cost of transport values much higher than their biological counterparts [5]. This is due at least in part to inefficiencies in their mechanical system and structure.

There are researchers who have explored other forms of actuation. These researchers use motors to drive mechanisms that are designed for a specific application. The robotic systems are capable of relatively long use-life from an onboard fuel source and low cost of transport. However, many of them are not intended for advanced and robust terrain navigation or other high level functionalities [6] [7]. One system that has seen successful in both autonomy and robust terrain navigation is the Atrias robot by the Dynamic Robots Lab (DRL) at Oregon State University. This robot has shown the capability of traversing uneven terrain while having a relatively long battery life. This success is due in no small part to their advanced leg design which they used to engineer the passive dynamics of their system. They chose to design their robot’s passive dynamics to conform to the Spring Loaded Inverted Pendulum Model. This allows them to model their system in a simple known way, and bypass many controller design problems [8]. Their leg design is a planar, four-bar mechanism in series with a revolute joint. This system uses a simple planar, serial mechanism structure. Our actuation method, on the other hand, employs a three dimensional mechanism that gives the potential to maximally utilize the motors that drive the system. In order to do this the geometry of the mechanism must be optimized.

An advanced mechanism such as the one described here presents a difficult design task. Spatial mechanisms have counter-intuitive properties that make them difficult to design without proper tools. This paper describes a method to use optimization techniques to select a unique configuration of the linkage geometry and stiffness that suits the specific requirements of the system. This includes not only an optimization of the actual geometric properties of the leg (link lengths, offset angles etc), but an optimization of other parameters, such as the gear ratio of a motor gear box and the stiffness of springs added across the joints of the limb. Each subsystem will be described in detail below:

Subsystem 1 - will be the optimization of the leg geometry, this subsystem will vary the parameters that define the linkage of the leg in order to reduce the mechanical power needed to achieve the desired gait pattern.
Subsystem 2 - will be the optimization of a stiffness matrix of the leg. This subsystem adds several parallel springs to the leg, and then tunes their stiffness values to minimize the mechanical power requirement.
Subsystem 3 - is the optimization of the motor gear ratio. This subsystem will optimize the gear ratio of a gear box attached to the motors to reduce their electrical power usage.
Subsystem 4 - is the gait pattern generation subsystem. This subsystem will tune parameters that define the gait pattern of the robot and use a dynamic simulator to find if which gait pattern is both stable and takes minimal effort from the motors.

Subsystem - Leg Geometry Optimization (Michael Naso)

The first step in optimizing the Leg Geometry is determining the parameter it is being optimized around. In this case the Leg Geometry is being optimized to minimize power consumption of the motors. A given gait pattern will include leg angles, velocities, and output force. This means that there is some fixed output power for this gait pattern. For any generic leg, each combination of position, velocity and force, will require various torques and velocities from the leg actuators. Based on the actuator configuration some motors may be doing positive work and some may be doing negative work (eg. braking). However, when determining the objective function only positive work will be accounted for. This is done because if the positive work is minimize so will the negative work be. There is already prior work done for this optimization such as function to calculate the torques and velocity at joints for a given gait pattern and leg geometry. Furthermore, another function to convert these torque and velocities at joints into torques and angular velocity for the motors. Improvements will be made to this optimization by reducing the variables using monotonicity as well as exploring different methods to find the minimum. These methods include gradient methods such as newton's as well as possible using CMA-ES.

2.1 Nomenclature:

1. R, Conditioning number. This is the 2 norm conditioning number, it is the ratio of the largest eigenvalue to the smallest eigenvalue of a matrix.

2. J, Jacobian of the leg. This is the standard Jacobian matrix, it transforms angular velocities of the motors to the linear velocity of the end effector.

3. $t_m[s]$, Minimum step time. This is the time taken to traverse the gait pattern when the motors are running at full capacity.

4. $P_m[W]$, Power. This is the power of the motor.

5. $\tau_m[Nm]$, This the torque exerted from the motor.

6. $\omega_m \left[\frac{rad}{s}\right]$, This is the angular speed of the motors.

7. $\tau_o[Nm]$, This is the torque out at the joints of the leg geometry.

8. $V_0 \left[\frac{m}{s}\right]$, This is the velocity outputs at the joints of the leg geometry

9. L [m], This the link lengths for the leg geometry

10. θ_l [rad], This is the angles at joints of the leg geometry

11. F[N] , Output forces.

12. G , This is the gait pattern.

13. P_m , This is the positioning of the motors with respect to the joints.

14. τ_{max} , Maximum torque the motors can operate with.

15. ω_{max} , Maximum angular velocity the motors can operate with.

16. θ_{max} , Maximum angles for which each joint can bend.

2.2 Mathematical Model

The mathematical model used during modeling the leg geometry mainly had to do with the physics behind the kinematics. This was a complex model developed by Nathan Cahill that used reverse kinematics to return velocity and torques at the joints of the leg given a gait pattern. The gait pattern used was data from a cat walking. Further information of the equations behind the kinematics can be found in the appendix.

$$[\tau_o, V_o] = f(\mathrm{G, L}) \quad (1)$$

Equation (1) is a simplistic representation of the reverse kinematic function. G is the gait pattern, and L is the lengths and joint angles of the leg geometry. This function outputs the torques and velocities at the joints in the leg throughout the gait pattern. Equation (2) takes this as an input and converts them to torques and angular velocities for the 3 motors.

$$[\tau_m, \omega_m] = f(L, \tau_0, V_o) \quad (2)$$

Simplification were used to reduce the degrees of freedom in the model. There were 44 degrees of freedom which was reduced to 32 by creating fixed values for certain parameters. This was done to reduce the run time of solving for the solution using multistart and fmincon. Furthermore, by reducing the degrees of freedom this allowed Latin hypercube sampling to be more effective in finding feasible solutions faster.

Objective Function

The objective function of this optimization is the equation for the power of the motors. This is the summation of all three motors power outputs over a period of a gait pattern.

$$P = \sum \tau_m \omega_m$$

This equation will be dependent of the variables described in the aforementioned function that were used to obtain the torque and angular velocities of the motors

2.2.2 Constraints

1. Conditioning number of Jacobian the leg must be less than 10 throughout the gait pattern. This will ensure that the leg is not close to singularity. When completing
2. The lengths must be positive numbers.

$$l_1 > 0$$

$$l_2 > 0$$

$$l_i > 0$$

3. The power output of the each motor must stay within the safe operating region of the motor.

$$0 < P_{max} < \tau_{max}\omega_{max}$$

4. Joint-leg angle limitation for each joint

$$-\theta_{max} < \theta_{l_1} < \theta_{max}$$

$$-\theta_{max} < \theta_{l_2} < \theta_{max}$$

$$-\theta_{max} < \theta_j < \theta_{max}$$

5. Conditioning number of the Jacobian.

   Since the inverse of the Jacobian is needed to calculate motor torques, the Jacobian needs to be invertible. Constraining the R conditioning number of the Jacobian makes sure the optimization will not pick a geometry that has a singularity in the region of the gait pattern.

   Note: The original optimization problem performed on this leg structure was to increase this parameter of the leg because early designs were close to singular throughout the workspace of the robot due to the complicated actuation method. Optimization techniques helped to make increase the conditioning number and give the leg full mobility in three dimensional space.

2.2.3 Design Variables and Parameters

The design variables for this system are the leg geometry parameters. They will be listed here: The thigh and shank lengths of the legs are included in the optimization. There is also one offset angle for the thigh that is also included. Each motor is attached to the leg by a two link RSS mechanism. These mechanisms can be defined by two length parameters. Each motor is positioned relative to the hip joint by two lengths and an angle, these parameters are also included in the optimization. Each RSS linkage must connect to the leg structure, each connection point is parameterized by 3 parameters relative to the leg segment. These are also included in the optimization. In all there are 27 parameters included in the optimization.

2.2.4 Model Analysis

Based on the degrees of freedom and the amount of variables in the objective the function will yield a minimum. Furthermore, all 32 design variables will have upper and lower bounds to ensure the space is well constraints and yields feasible solutions. In addition, a constraint on the R condition number is used to reduce the singularities at a given point in the gait pattern. This constraint is necessary due to limitation in actuation methods.

2.2.5 Optimization Study

The objective function has many local minimum therefore Latin hyper cube sampling was used to gather initial points. In order to ensure these point are feasible, a range between the lower bounds and upper bounds was used. In addition to make sure these initial states were feasible the R condition constraint was used in the Latin hyper cube sampling. Furthermore, once these initial states were found a multistart method with a 1000 initial points was used to optimize the solution around these feasible domains. Given that function was well constraint the method used for optimization was MATLAB fmincon.

Due to the there being very few possible configuration of the 32 dimensioned leg geometry that follows the constraints only 5 initial points were feasible out of 100,000 samples. These 5 initial points were then inputted into multistart optimization as the initial geometry and power outputs were determined.

Table 1: Initial Points from Latin Hyper Cube Sampling

Initial Points	Initial Total Power	Optimized Total Power
1	10.2541	1.4273
2	11.6984	3.8032
3	10.1132	6.6763
4	14.1094	1.3649
5	10.5678	3.2486

Optimization around an educated guess of the leg geometry was done to see the sample space around this initial guess was more viable. In order to sample the space around this given point a random number was added to the initial leg geometry and iteration was done to find a more optimal solution using multistart and fmnicon.

Table 2: Optimization around educated guess

Initial Points	Initial Total Power	Optimized Total Power
1	**1.2905**	**1.2855**
2	1.4789	1.3567
3	1.5376	1.4106
4	1.4103	1.4026
5	1.4576	1.3987

This proved to be the most effective method as it yielded a result of 1.2855 for the total power consumption. This shows that for complicated optimization problems sampling around an intuitive initial point will yield better results. This due to the fact the Latin hypercube sampling was very costly in run time due to the 32 dimensions in the design variable space. Furthermore, due to the R conditioning constraint there weren't many feasible points scattered within the space. If more computations could be done faster sampling over a million points may have yielded better results for the points found using Latin Hypercube Sampling.

Sensitivity Analysis was done on the constraints to determine which constraints were active and the Lagrange multipliers of those constraints.

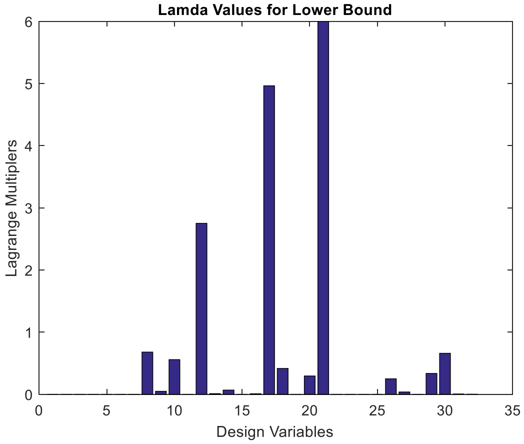


Figure 1

In Figure 2.5.1 it can be seen that the lower bounds on the design variables had very high Lagrange multipliers on certain design variables. For instance the 21st and 17th design variable had a Lagrange multiplier of 5.9982 and 4.9628. By relaxing this constraint this could yield high improvement in the minimization of power however the lower bounds for these variables are 0 and cannot be relaxed. It is shown that these constraints in the lower bound were active.

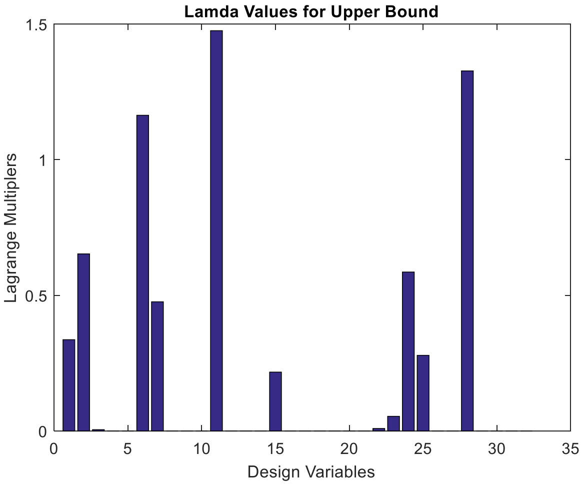


Figure 2

In Figure 2.5.2 the Lagrange multipliers were not as high yielding a max of 1.4756. Relaxing the constraint on the design variable with the Lagrange multiplier of 1.4756 would minimize the optimization result slightly. However, the upper bounds cannot be relax due to limitation of the size of the leg.

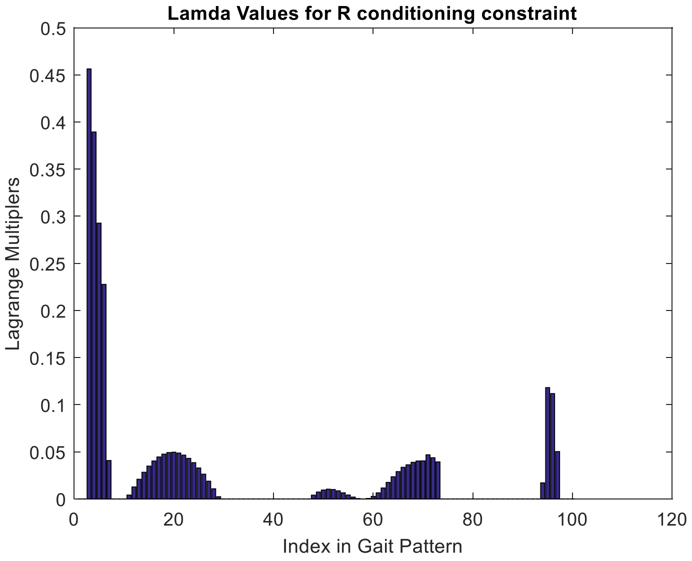


Figure 3

The Lagrange multipliers for R conditioning number yielded the smallest Lagrange multipliers. There is very small improvement that can be made by relaxing this constraint.

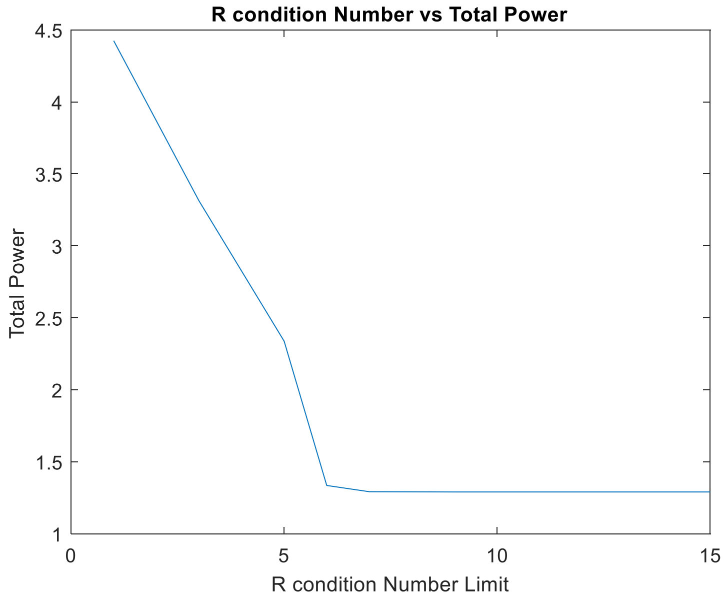


Figure 4

As can be seen in Figure 2.5.4 the R conditioning constraint has a significant effect on the total power. However, with an initial R conditioning number of 7 for my solution there isn't a lot of room for improvement by relaxing this constraint any more. Furthermore, at an R conditioning number of 10 the Lagrange multiplier for the R conditioning number constraint is reduced to 0.0657.

2.2.6 Parametric Study

Due to the objective function containing 32 design variables the space of function is very large. Furthermore, the function has very many local minimums making it difficult to find a global minimum. Due to these characteristic of the function different initial parameters will change the optimization results. By using fmincon it finds the closest local minimum to that initial point. In order to look for a global minimum effectively Latin Hypercube sampling was done on the space. 100,000 points were sampled but only 5 of these points were in the feasible domain created by the bounds and constraint on the design variables. Theses 5 points were then optimized using multistart will a 1000 points to look for an optimal solution around the feasible domains.

2.2.7 Discussion of Results

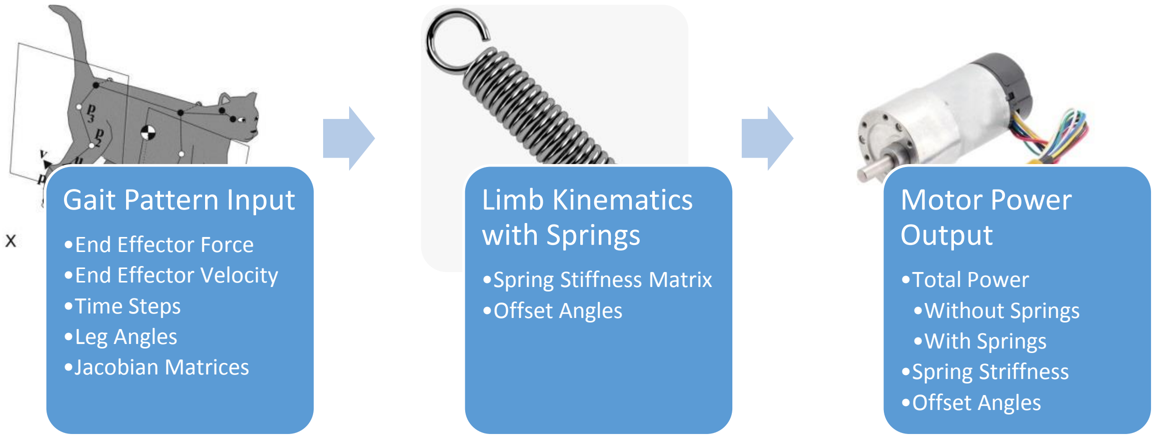
The results of the optimization were fairly good. The power output was reduced 8 % from the previous optimization done. Furthermore, the solution obtain was realistic and could be applied to real world application. When designing the leg geometry the main design implication was creating a constraint on the R conditioning number. This allowed the optimization results to be valid leg geometry that can work in real application. The lower bound and upper bound of some of the design variables were active. However, these constraints could not be relaxed due to impractical leg geometry that could not be created. These bounds allowed the design variables to be constraint to produce realistic results with the objective function used. In addition, the main constraint that was active and could be relaxed was the R condition number. In Figure 2.5.4 it shows that the optimal R conditioning number is around 7. Further relaxing this constraint doesn't yield better results. The R conditioning number is the main constraint that allows the optimization to create realistic leg geometry that can complete the gait pattern efficiently.

Improvements that could be made in the model is reducing the dimensionality of the problem. 32 design variables reduced the computational ability to sample the space effectively. Furthermore, if higher processing computer was available the Latin Hypercube sample could be increased to a million points. In addition, when optimizing these initial points a multistart method with 10,000 point would be more effective in finding an optimal solution within the feasible domain of the initial point. This would allow the optimization to be closer to finding a global solution. In addition, by reducing the dimensionality of the problem the resulting function could have less local minimums increasing the chances of finding a global minimum.

3 Subsystem – Spring Optimization (Hosain Bagheri)

3.1 Design Problem Statement

The ideal is simple, yet ingenious; through the implementation of springs upon a legged robot, we intend not only to improve the performance, but more importantly decrease total power cost. One of the continuous dilemmas faced when designing efficient and effective robots is how to approach the problem at hand. Do we design a lightweight robot that lacks robustness and agility in order to decrease the power required to operate it, or do we design a robust and agile robot that will require greater power consumption? The situation at hand is a double edged sword. However one method of finding the middle ground and providing the harmony between the better of the two worlds is by implementing springs into the system. Springs are excellent in storing potential energy, and are usually used for returning something to its original position. In this specific application, springs will take the kinetic energy created during motors breaking and absorb it in the form of potential energy and transfer it back to kinetic energy in the (passive) dynamic of motion. While the applications of springs in robotics are not something new, they have been commonly seen to be used in damping systems. However, in systems which the inertia (of the legs) dominates, springs are quite beneficial with their ability to overcome this inertia. We indeed to use it in the designed linkages to provide flexibility and to decrease total motor power cost. The general flow of the process is as follows:



Gait Pattern Input

- End Effector Force
- End Effector Velocity
- Time Steps
- Leg Angles
- Jacobian Matrices

Limb Kinematics with Springs

- Spring Stiffness Matrix
- Offset Angles

Motor Power Output

- Total Power
  - Without Springs
  - With Springs
- Spring Striffness
- Offset Angles

Using the empirically obtained gait pattern data of a cat, which includes such things as force, velocity, time, and leg angles, the effect of springs will be considered while taking into to account their stiffness and offset angles, to determine if the total motor power can be reduced when comparing the limb linkage design with and without the implementations of spring. Note that a series of Jacobian matrices will be utilized to transfer of information from one coordinate system to another.

3.2 Nomenclature:

1) T_{motors} $[\mathrm{N \cdot m}]$, motor torque
2) F_{motor} $[\mathrm{N}]$, motor force
3) F_{total} $[\mathrm{N}]$, total force

4) $F_{total,motor}$ [N], total force in motor coordinates

5) F_{spring}[N], spring force

6) $F_{spring,motor}$ [N], spring force in motor coordinates

7) V_{motor} [m/s], motor velocity

8) P_{motor}[W], total motor power

9) K $[N/m]$, spring stiffness matrix

10) k $[N/m]$, spring stiffness constant

11) β $[rad]$, offset angle

12) θ $[rad]$, leg angle

13) J_{leg}, Jacobian of the leg. This is the standard Jacobian matrix which transforms angular velocities of the motors to the linear velocity of the end effector.

3.3 Mathematical Model

3.3.1 Objective Function

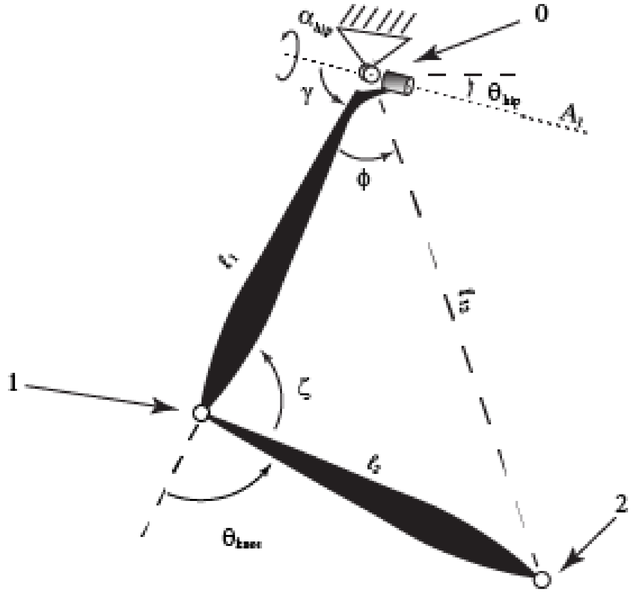


Figure 5. Robot Leg Schematic

With the absence of springs, the total end effector force (at the robot's ankle) will only be generated by the motors. Although through the implementation of springs, the total end effector force will be a collective effort of effort from the motors and spring. Using Hooke's law and considering the offset angle at the robot's leg angle, the spring force can be formulated. The gait pattern provides the end effector force and velocity. With the provided information and calculated spring force, the motor force can be obtained and multiplied by the velocity to obtain the power. It should be noted that a series of Jacobian matrices will need to be used to transfer the coordinate system from the end effector to the motor coordinate system.

$$\begin{gathered} F_{motors} = F_{tot\square l,motor} - F_{springs,motor} \\ F_{motors} = B_m - A_m x \end{gathered} \tag{1}$$

$$\begin{gathered} P_{motors} = F_{motors}^T \cdot V_{motors} \\ P_{motors} = (B_m - A_m x)^T C_m \end{gathered} \tag{2}$$

$$\begin{aligned} P_{motors}^2 &= (C_{md}B_m - C_{md}A_m x)^T (C_{md}B_m - C_{md}A_m x) \\ &= B_m^T C_{md}^2 B_m - B_m^T C_{md}^2 A_m x - x^T A_m^T C_{md}^2 B_m + x^T A_m^T C_{md}^2 A_m x \\ &= \frac{1}{2} x^T H x + f^T x + c \\ H &= 2A_m^T C_{md}^2 A_m \\ f &= -2B_m^T C_{md}^2 A_m \\ c &= B_m^T C_{md}^2 B_m \end{aligned} \tag{3}$$

Please see appendix for detailed derivation.

3.3.2 Constraints

The main physical constraint is that the motor torque has a maximum allowable torque value based on the motor's velocity. The maximum allowable motor torque can be calculated from the information provided in the Motor Optimization Subsystem, where end effector velocity acquired from the gait pattern would be mapped to motor coordinates and evaluated at every time step. The other physical constraints is based on the provided gait pattern information, in which it is known that the robot does not have abduction and adduction about the hip joint, thus the leg angles and springs associated in this plane can be disregarded. Lastly, a physical/practical constraint is that the stiffness matrix is positive symmetric matrix.

$$
\begin{aligned}
&\tau_{motors} \leq \max \tau_{allowable} \\
&J^T F_{motor} \leq T \\
&F_{motor} \leq (J^T)^{-1} T \\
&B_m - A_m x \leq (J^T)^{-1} T \\
&-A_m x \leq (J^T)^{-1} T - B_m \\
&-A_m x \leq b
\end{aligned} \tag{4}
$$

$$T = \left\{ \left(\frac{217\ -\ 108}{0 - 8000} \cdot \frac{2\pi}{60} \right) V_{motors} + 217 \right\} / 1000 \tag{5}$$

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} \theta_1 \\ 0 \\ \theta_3 \end{bmatrix} \tag{6}$$

$$\beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} \beta_1 \\ 0 \\ \beta_3 \end{bmatrix} \tag{7}$$

$$K = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{12} & k_{22} & k_{23} \\ k_{13} & k_{23} & k_{33} \end{bmatrix} = \begin{bmatrix} k_{11} & 0 & k_{13} \\ 0 & 0 & 0 \\ k_{13} & 0 & k_{33} \end{bmatrix} \tag{8}$$

3.3.3 Design Variables and Parameters

The gait pattern provides us with the parameter values of the leg (flexion/extension) angles, which consists of the hip and knee joint. The gait pattern also provides us with the end effector force and velocity, which will have to put into motor coordinate systems. The design variables are the stiffness matrix and the offset angles, which after applying the constraints there will be 5 degrees of freedom.

Table 2. Design Variables and Parameters

Design Variables	**Design Parameters**
β_1, Offset Angle at Hip Joint	θ_1, Hip Angles - Flexion/Extension
β_3, Offset Angle at Knee Joint	θ_3, Knee Angles - Flexion/Extension

k_{11}, Spring Stiffness at Hip Joint	$F_{total,ankle}$, End Effector Force
k_{33}, Spring Stiffness at Knee Joint	$V_{total,ankle}$, End Effector Velocity
k_{13}, Spring Stiffness across the Hip and Knee	

3.3.4 Summary Model

Objective Function:

$$\min_{K,\beta} P_{motors}^2 = \frac{1}{2} x^T H x + f^T x$$

$$H = 2A_m^T C_{md}^2 A_m \tag{9}$$

$$f = -2B_m^T C_{md}^2 A_m$$

Constraints:

$$s.t. A_m x \leq b$$

$$b = (J^T)^{-1} T - B_m \tag{10}$$

$$k_{13} \geq 0 \tag{11}$$

$$k_{13} \geq 0 \tag{12}$$

$$k_{13} = k_{31} \tag{13}$$

$$k_{12} = k_{21} = k_{22} = k_{23} = k_{32} = 0 \tag{14}$$

$$\theta_2 = \beta_2 = 0 \tag{15}$$

Note that earlier in the design formulation, the length and locations of the springs were incorporated into the equations. However, they were removed in order to decrease the complexity of the problem and provide the ability to formulate a linear equation for the total motor power. Therefore, the current problem simply considers the influence of springs, but another optimization process can be established in the future in which based on the output spring stiffness values it would find the ideal location to place the spring on the robot's leg.

3.4 Optimization Study

Following the mathematical model, a code was written in Matlab in which fmincon and quadprog were utilized to obtain the stiffness matrix and offset angles while minimizing the total motor power. For fmincon, the linear equation for power was utilized with the option of SQP to increase the efficiency of the process, while for quadprog the quadratic equation for power was used. The solution obtained from both methods matched, and it was seen that through implementation of springs, the total power was decreased by about 58%. The values of the stiffness matrix were reasonable, especially considering the values of empirical end effector force and velocity, which are normalized to the weight of the robot and have a max value of about 0.6 and 0.7, respectively. The knee joint was seen to have the greatest spring stiffness, when compared to that of the hip and the one across the hip and knee. This is due to the fact that 2 body links are working at the knee joint, thus requiring greater spring stiffness, while the others only work on body link. Through debugging the code, it was observed that the motor gear ratio and moment of inertia of the leg have influence on the optimization process. Such components like motor gear ratio and moment of inertia were incorporated to make the problem more realistic. For example there is a point in the gait cycle which the robot leg does not have contact with the ground, making the reactionary force and thus the end effector zero. Although, by incorporating the inertia of the leg we can still account for the forces in instants such as these. When calculating the maximum allowable motor torque, initially gears were not considered. Although since one of the existing subsystems deals with the optimization of the gear ratio, we decided to incorporate it into the calculation by multiplying the gear ratio to the motor velocity. It was seen that by increasing the gear ratio, the stiffness matrix was seen to decrease in value, while the total motor power stayed the same. This indicates that there a given threshold for gear ratio, in which no additional power can be saved. Furthermore, when increasing the gear ratio, more of the work is being done by the motors and less by the springs. In the situation in which gear ratio was decreased, the stiffness matrix increased along with total motor power. The adverse was seen when decreasing the gear ratio. When decreasing the gear ratio, more of the work is being done by the motors and springs. It was seen that by increasing the moment of inertia, the stiffness matrix and total power would increases. The adverse effect is seen when decreasing the moment of inertia. This indicates that the greater moment arm and thus mass, will increase the springs stiffness to counterbalance it. Furthermore, the increase of moment of inertia will required greater work to complete a task and thus an increase in motor power. These are all sound results based on engineering interoperations and intuitions. Unfortunately, little parametric study was performed since only one gait pattern was provided. Even though the other subsystems are on the optimization of the geometry and gait pattern, it is very computationally costly to run them for results. However, it would be quite interesting to confirm if the implementation of springs on the gait pattern would indeed decrease the total required motor power. Also, to determine if applied spring would hinder the gait pattern. Furthermore, it would be nice to see how the stiffness matrix would change in respect to the leg geometry, and what the length and location of the springs would have to be to obtain a more optimal solution in decreasing the total motor power. If time permitted, those are some areas where future investigation would have been performed.

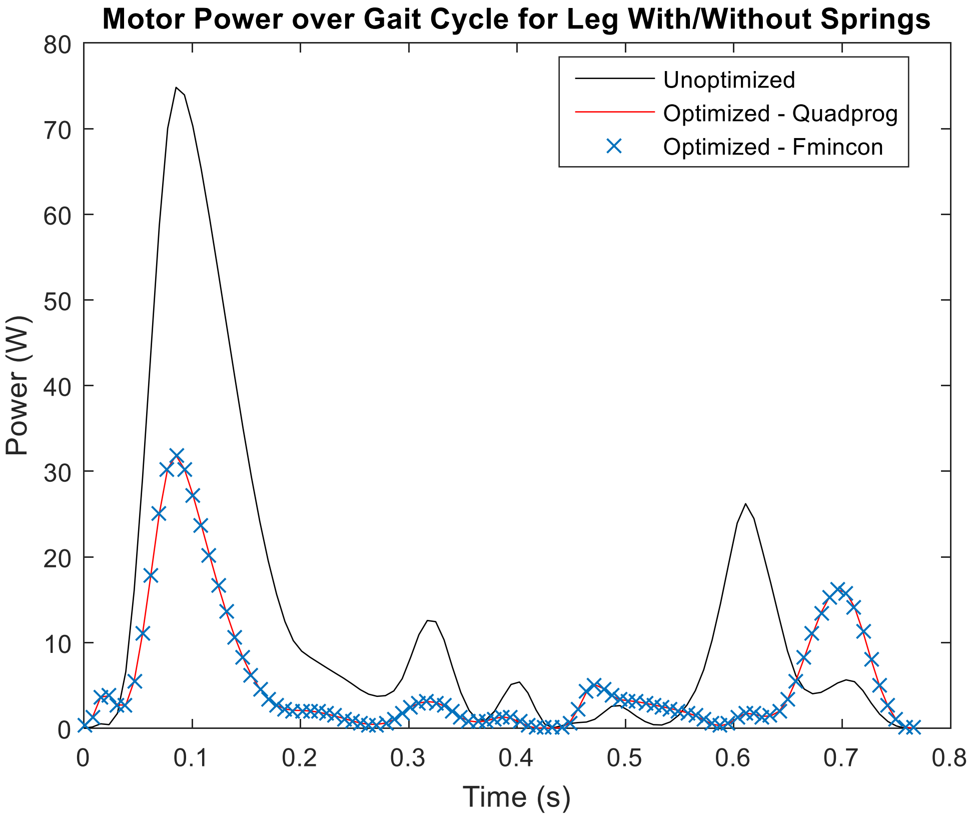


Figure 6. Total Motor Power Optimization Results

Table 3. Total Motor Power Optimization & Spring Stiffness Results

	Fmincon	Quadprog
Total Motor Power without Springs, $P_{motors,w/o\ springs}(W)$	1.2879E3	1.2879E3
Total Motor Power with Springs, $P_{motors,w/\ springs}(W)$	537.2587	537.2587
Spring Stiffness at Hip Joint, k_{11} (Nm/rad)	0.0339	0.0339
Spring Stiffness at Knee Joint, k_{33} (Nm/rad)	0.2349	0.2349
Spring Stiffness across the Hip and Knee, $k_{13}, k_{31}(Nm/rad)$	0.0520	0.0520
Offset Angle at Hip Joint, $\beta_1\,(rad)$	-0.0216	-0.0216
Offset Angle at Knee Joint, $\beta_3\,(rad)$	0.0956	0.0956

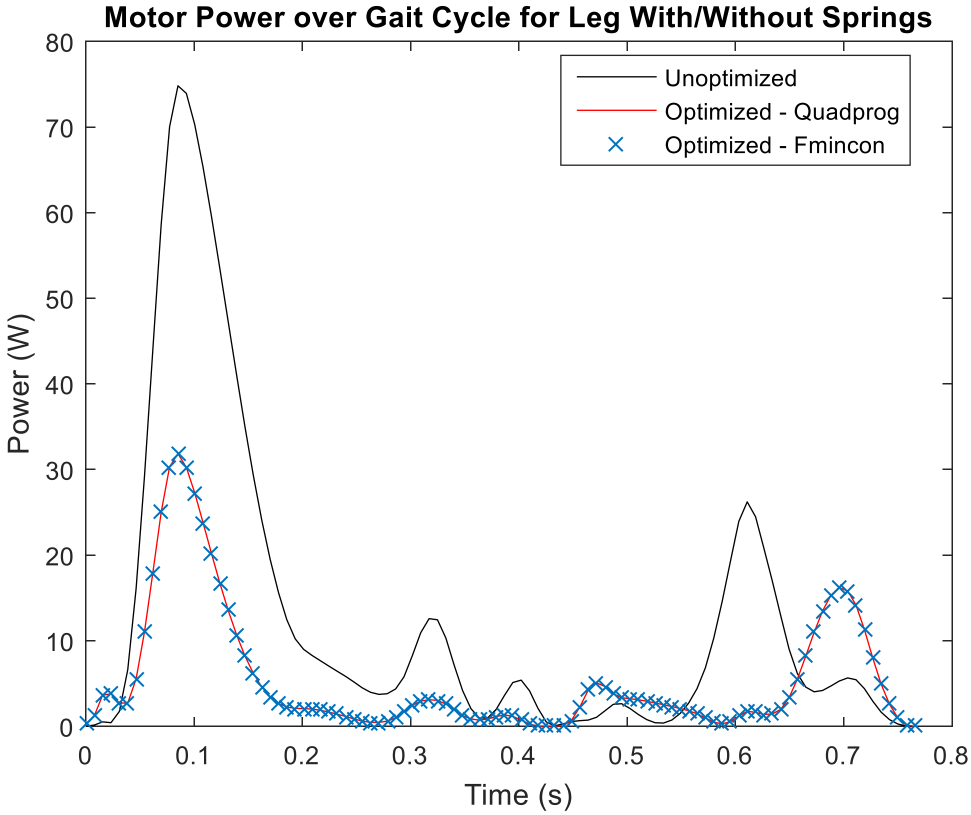


Table 3. Total Motor Power Optimization With Increase Gear Ratio

Table 3. Total Motor Power Optimization & Spring Stiffness Results With Increase Gear Ratio

Increase in Gear Ratio by 10	**Fmincon**	**Quadprog**
Total Motor Power without Springs, $P_{motors,w/o\ springs}(W)$	1.2879E3	1.2879E3
Total Motor Power with Springs, $P_{motors,w/\ springs}(W)$	537.2587	537.2587
Spring Stiffness at Hip Joint, k_{11} (Nm/rad)	0.0244	0.0244
Spring Stiffness at Knee Joint, k_{33} (Nm/rad)	0.1689	0.1689
Spring Stiffness across the Hip and Knee, $k_{13}, k_{31}(Nm/rad)$	0.0374	0.0374
Offset Angle at Hip Joint, β_1 (rad)	-0.0216	-0.0216
Offset Angle at Knee Joint, β_3 (rad)	0.0956	0.0956

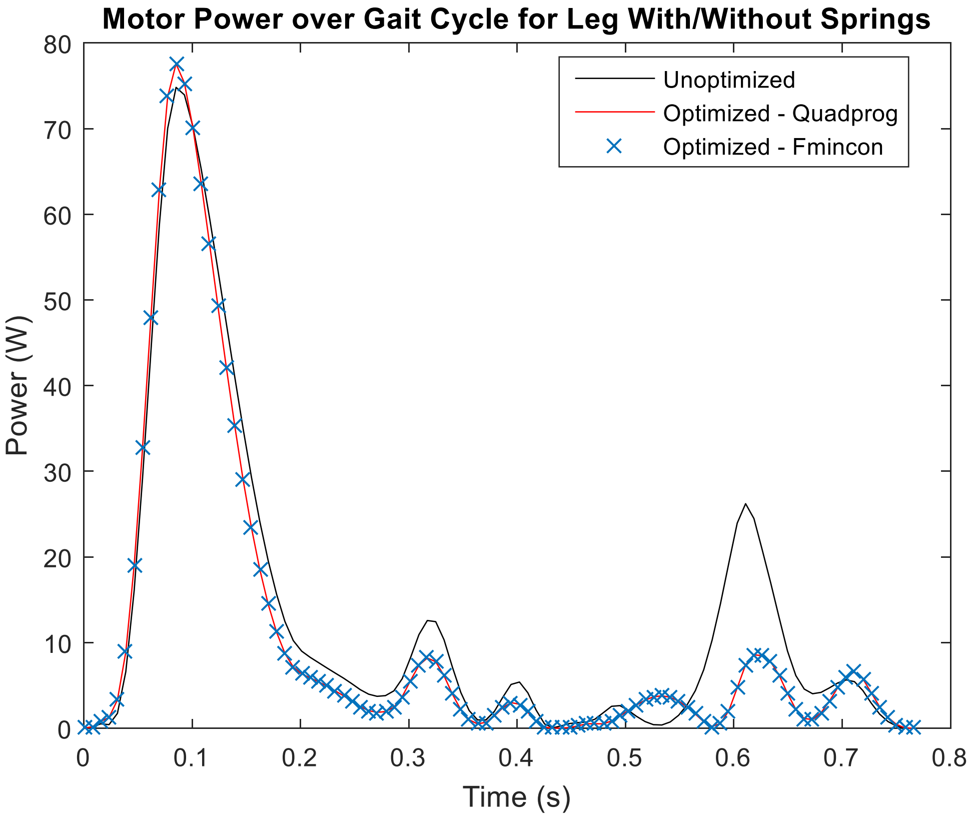


Table 4. Total Motor Power Optimization With Decrease Gear Ratio

Table 4. Total Motor Power Optimization & Spring Stiffness Results With Decrease Gear Ratio

Decrease in Gear Ratio by 10	**Fmincon**	**Quadprog**
Total Motor Power without Springs, $P_{motors,w/o\ springs}(W)$	1.2879E3	1.2879E3
Total Motor Power with Springs, $P_{motors,w/\ springs}(W)$	1.0597E3	1.0597E3
Spring Stiffness at Hip Joint, k_{11} (Nm/rad)	0.1363	0.1363
Spring Stiffness at Knee Joint, k_{33} (Nm/rad)	0.5105	0.5105
Spring Stiffness across the Hip and Knee, $k_{13}, k_{31}(Nm/rad)$	0.2179	0.2179
Offset Angle at Hip Joint, $\beta_1\,(rad)$	-0.1013	-0.1013
Offset Angle at Knee Joint, $\beta_3\,(rad)$	0.3680	0.3680

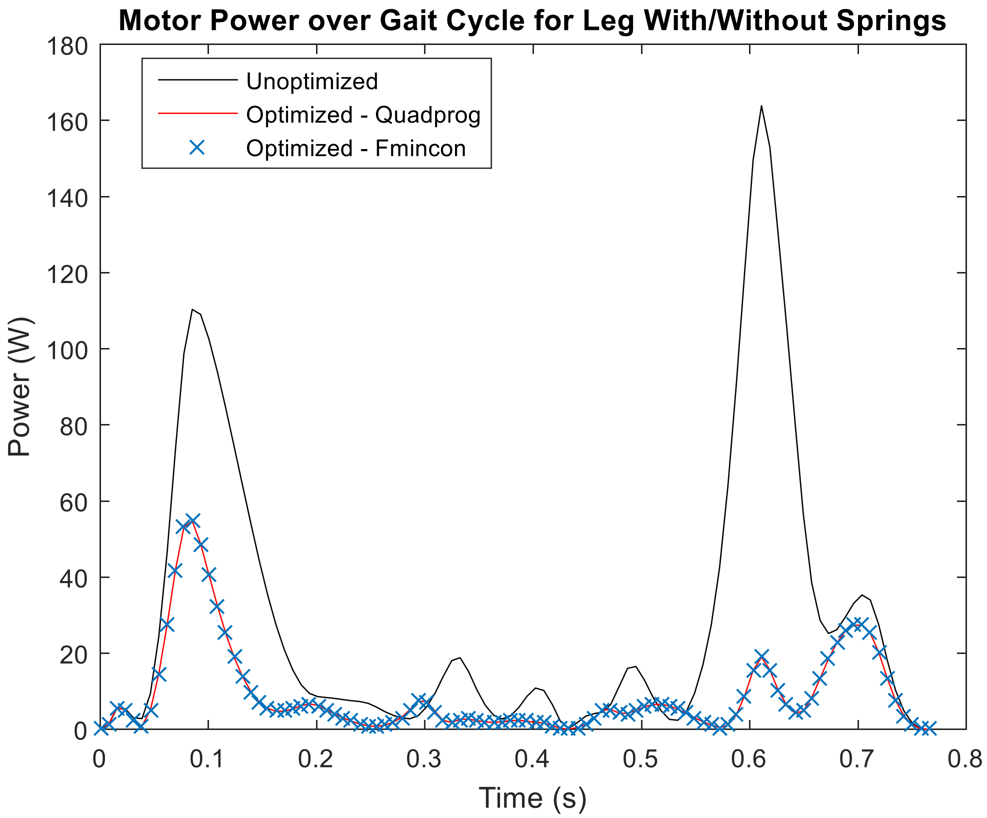


Table 5. Total Motor Power Optimization With Increase Moment of Inertia

Table 5. Total Motor Power Optimization & Spring Stiffness Results With Increase Moment of Inertia

Increase in Moment of Inertia by 0.04	**Fmincon**	**Quadprog**
Total Motor Power without Springs, $P_{motors,w/o\ springs}(W)$	2.9842E3	2.9842E3
Total Motor Power with Springs, $P_{motors,w/\ springs}(W)$	900.0445	900.0445
Spring Stiffness at Hip Joint, k_{11} (N/m)	0.0217	0.0217
Spring Stiffness at Knee Joint, k_{33} (N/m)	0.1560	0.1560
Spring Stiffness across the Hip and Knee, $k_{13}, k_{31}(N/m)$	0.2843	0.2843
Offset Angle at Hip Joint, $\beta_1\,(rad)$	0.0389	0.0389
Offset Angle at Knee Joint, $\beta_3\,(rad)$	0.0843	0.0843

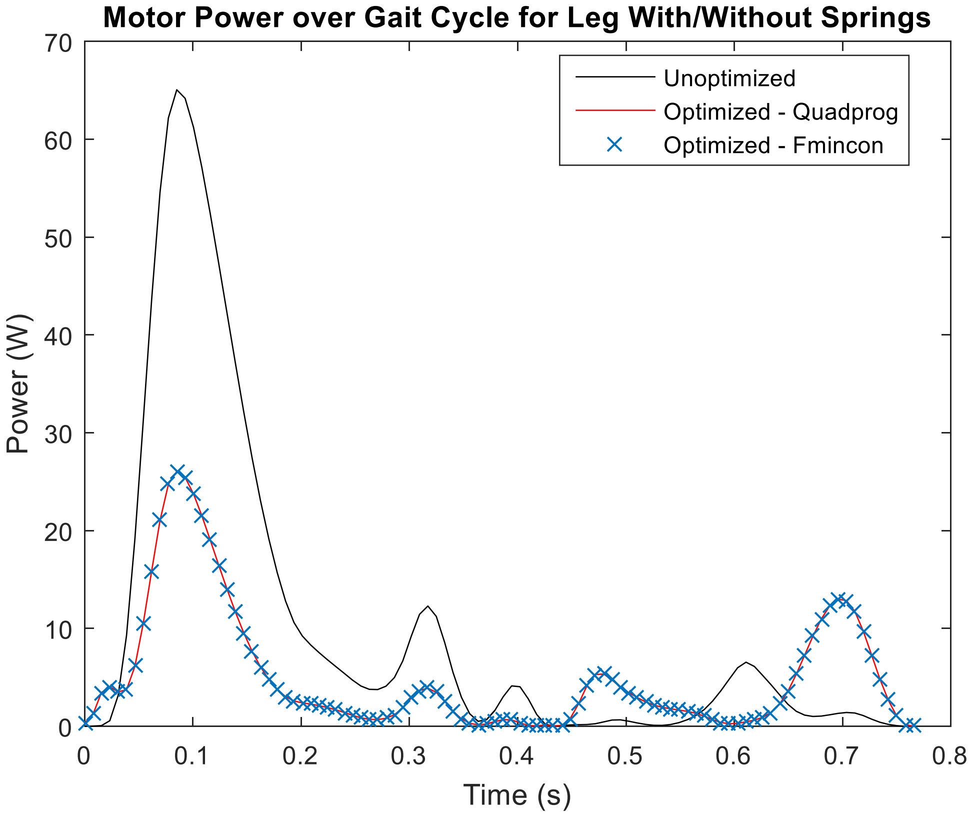


Figure 6. Total Motor Power Optimization Results With Decrease Moment of Inertia

Table 6. Total Motor Power Optimization & Spring Stiffness Results With Decrease Moment of Inertia

Decrease in Moment of Inertia by 0.01	**Fmincon**	**Quadprog**
Total Motor Power without Springs, $P_{motors,w/o\ springs}(W)$	1.0128E3	1.0128E3
Total Motor Power with Springs, $P_{motors,w/\ springs}(W)$	497.4822	497.4822
Spring Stiffness at Hip Joint, k_{11} (Nm/rad)	0.0374	0.0374
Spring Stiffness at Knee Joint, k_{33} (Nm/rad)	0.3182	0.3182
Spring Stiffness across the Hip and Knee, $k_{13}, k_{31}(Nm/rad)$	0.0318	0.0318
Offset Angle at Hip Joint, $\beta_1\,(rad)$	0.0130	0.0130
Offset Angle at Knee Joint, $\beta_3\,(rad)$	0.1698	0.1698

4 Subsystem – Motor Optimization (Peng Wei)

4.1 Problem Statement

In this robot, a high torque is required at a relatively low shaft speed, thus a gear system is needed to help get the job done. Gears can increase or decrease the speed of rotation (decrease or increase the torque on the other hand) and can easily be used to reverse the direction of rotation. The goal of this subsystem is to minimize the input power of motors by changing the gear ratios and ensure the motor running in continuous operation range. An optimized gear ratio allows the motor to run at an efficient condition when the gait pattern is fixed and that high efficiency is beneficial to the total energy cost of this robot. However, the gear ratio is restricted by some constraints. For example, the maximum speed is fixed for some mechanical reasons and the motor should run under continuous operating conditions to satisfy the thermal limits. At the end of the day, this will become an optimization problem where minimizing the input power is the objective function and those mechanical and thermal limits can be regarded as the constraints. To accomplish that, the knowledge gained in the class would be used. Moreover, the MATLAB build-in function - ‘fmincon’ will be utilized to solve this optimization problem.

4.2 Nomenclature

1) g, Gear ratio
2) η [%], Efficiency of the motor
3) M_e [mNm], Torque after gear
4) $\acute{\theta}_e$ [rad/s], Velocity after gear
5) M [mNm], Motor torques
6) $\acute{\theta}$ [rad/s], Motor velocity
7) $\acute{\theta}$ [rad/s^2], Motor acceleration
8) I [mA], Current
9) U [V], Voltage
10) K_m [mNm/A], Torque constant
11) K_n [rpm/V], Speed constant
12) I_{max} [A], Nominal current (This is the maximum continuous current for each motor)
13) I_0 [A], No load current
14) M_{max} [mNm], Nominal torque (This is the maximum continuous torque for each motor)

15) n_{max} [rpm], Maximum speed of the motor

16) J [gcm^2], Rotor Inertia

17) R [Ω], Terminal resistance

18) P [W], Power

4.3 Mathematical Models

For a fixed gait pattern, the function for this subsystem should run the torque/velocity after gear for each motor and output the gear ratios which minimize the total electrical power. The constraints come from the mechanical and thermal limits for the specific motor model. Those gear ratios will be what the optimizer changes.

Objective function

The gear ratio is defined as the input speed relative to the output speed (or output torque to the input torque):

$$P_{in} = P_{out}$$

$$M_m \cdot \dot{\theta}_m = M_e \cdot \dot{\theta}_e$$

$$\frac{M_e}{M_m} = \frac{\dot{\theta}_m}{\dot{\theta}_e} = \ g \ (gear\ ratio)$$

To derive the expression of objective function, let’s start from the power balance equation for the motor system. The electrical motor convert electrical power P_{el} (current I_{mot} and voltage U_{mot}) into mechanical power P_{mech} (speed n and torque M). The losses that arise are divided into frictional losses, attributable to P_{mech} and in Joule power losses P_J of the winding (resistance R). The power balance can therefore be formulated as:

$$P_{el} = P_{mech} + P_J$$

where

$$P_{el} = U \cdot I$$

$$P_{mech} = \frac{\pi}{30{,}000} M \cdot \dot{\theta}$$

$$P_J = R \cdot {I_{mot}}^2$$

Here, both current and voltage are functions of torque and speed, the relations are given as:

$$I = \frac{M + \ddot{\theta} \cdot J}{K_m}$$

$$U = I \cdot R + \frac{\dot{\theta}}{K_n}$$

Where K_m is the torque constant, K_n is the speed constant, J is the rotor inertia and R is the resistance. The acceleration term can be estimated using finite difference method. Also applying the gear ratio on the motor velocities and torques:

$$M = \frac{M_e}{g}$$

$$\dot{\theta} = g\dot{\theta}_e$$

$$\ddot{\theta} = \frac{d\dot{\theta}}{dt} = g\frac{d\dot{\theta}_e}{dt} = g\frac{\dot{\theta}_{e,i-1} - 2\dot{\theta}_{e,i} + \dot{\theta}_{e,i+1}}{(dt)^2}$$

Combine all the relations above, the equation can be rewritten as:

$$P = (I \cdot R + \frac{\dot{\theta}}{K_n}) \cdot (\frac{M + \ddot{\theta} \cdot J}{K_m})$$

$$P_{el} = (\frac{M + \ddot{\theta} \cdot J}{K_m} \cdot R + \frac{\dot{\theta}}{K_n}) \cdot \frac{(\frac{M_e}{g} + \frac{d\dot{\theta}}{dt} \cdot J)}{K_m}$$

$$P_{el} = (\frac{\frac{M_e}{g} + \frac{d\dot{\theta}}{dt} \cdot J}{K_m} \cdot R + \frac{g\dot{\theta}_e}{K_n}) \cdot \frac{(\frac{M_e}{g} + \frac{d\dot{\theta}}{dt} \cdot J)}{K_m}$$

$$P_{el} = (\frac{M_e R}{gK_m} + g\frac{d\dot{\theta}_e}{dt} \cdot \frac{JR}{K_m} + g\frac{\dot{\theta}_e}{K_n}) \cdot (\frac{M_e}{gK_m} + g\frac{d\dot{\theta}_e}{dt} \cdot \frac{J}{K_m})$$

$$P_{el} = \frac{d\dot{\theta}_e}{dt} \cdot \left(\frac{d\dot{\theta}_e}{dt} \cdot \frac{J^2 R}{{K_m}^2} + \frac{J\dot{\theta}_e}{K_n \square_m}\right) \cdot g^2 + \frac{{M_e}^2 R}{{K_m}^2} \cdot \frac{1}{g^2} + \frac{M_e}{K_m} \cdot (2\frac{d\dot{\theta}_e}{dt} \cdot \frac{JR}{K_m} + \frac{\dot{\theta}_e}{K_n})$$

This is the expression of electrical power at a given time step (a given torque and velocity). Since the torque and velocity will change within the gait pattern, and the gear ratio is constant during the process, the total power should be the sum of the input power calculated at each time step for all three motors, which is:

$$P_{el} = \sum \left(P_{el,1} + P_{el,2} + P_{el,3}\right)$$

Where the index 1,2,3 represent three motors. This the final expression of objective function. In summary, the goal is to minimize the input electrical power by choosing the best gear ratios under a given gait pattern and geometry. The mathematic expression is:

Objective function: $min_{gear\ ratio}\ (Electrical\ Power)$

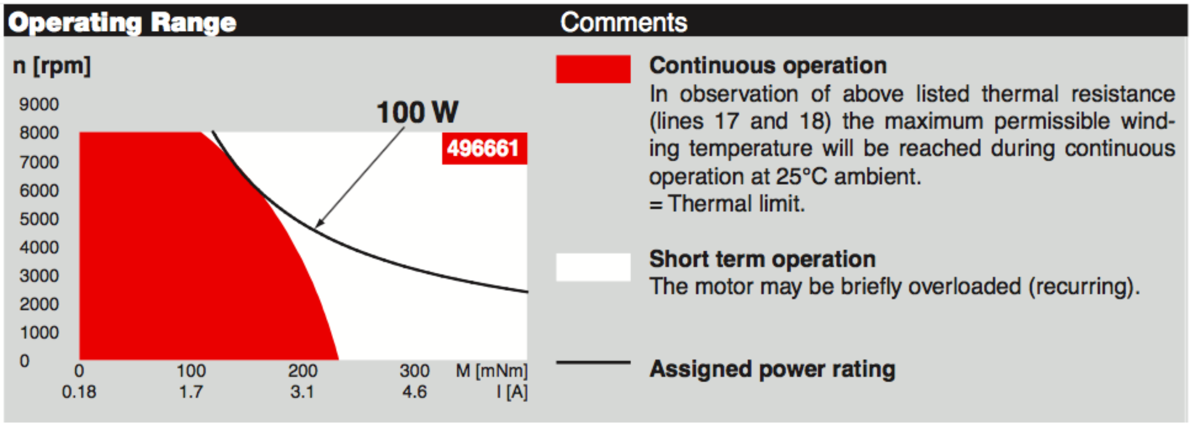
Electrical Power = function (gear ratio, torque and velocity from gait pattern)

Constraints

Table 2.3.3.2.1: Detailed Motor Information

Nominal voltage	36	V
No load speed	4550	rpm
No load current	176	mA
Nominal speed	3950	rpm
Nominal torque (max. continuous torque)	217	mNm
Nominal current (max. continuous current)	2.84	A
Stall torque	3520	mNm
Stall current	46.9	A
Max. efficiency	88	%
Terminal resistance phase to phase	0.767	Ω
Terminal inductance phase to phase	0.675	mH
Torque constant	74.9	mNm/A
Speed constant	127	Rpm/V
Speed/torque gradient	1.31	Rpm/mNm
Mechanical time constant	0.601	ms
Rotor inertia	44	gcm^2
Thermal resistance housing-ambient	7.17	K/W
Thermal resistance winding-housing	1.35	K/W
Thermal time constant winding	23.1	s
Thermal time constant motor	1400	s

Ambient temperature	-40 ~ +100	℃
Max. winding temperature	+155	℃
Max. speed	8000	rpm



🟥 **Continuous operation**
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.

□ **Short term operation**
The motor may be briefly overloaded (recurring).

— **Assigned power rating**

Figure 2.3.3.2.1: Operating Range of the Selected Motor

Based on the table and graph, to ensure the motor run in continuous operating range (the red region), these conditions should be satisfied:

1) The speed of each motor should be smaller than Max. speed n_{max}

$$0 \leq \acute{\theta}_1 \leq n_{max}[rpm]$$

$$0 \leq \acute{\theta}_2 \leq n_{max}[rpm]$$

$$0 \leq \acute{\theta}_3 \leq n_{max}[rpm]$$

2) The torque for each motor should be smaller than the max. continuous torque M_{max}

$$0 \leq M_1 \leq M_{max}[mNm]$$

$$0 \leq M_2 \leq M_{max}[mNm]$$

$$0 \leq M_3 \leq M_{max}[mNm]$$

3) As we can see, the max. continuous torque is not a straight line and the value varies with different motor speeds. The reason is that the motor is restricted by some thermal limitations. The Joule power losses heat up the winding, the heating produced must be able to dissipate and the maximum rotor temperature should not be exceeded.

$$T_w - T_U = \Delta T_w = (R_{th1} + R_{th2}) \cdot P_J \leq Max.Temperature\ Difference$$

This results in a maximum continuous current (torque), at which the maximum winding temperature is attained under standard condition. Higher motor currents cause excessive winding temperatures. With EC motors, eddy current losses increase in the return as speed increases and produce additional heat. Thus the maximum permissible continuous current (torque) decreases at faster speeds accordingly.

The relation is provided by manufacturer and not easy to calculate by hands, thus some approximations are used here. By observation, the curve has a parabolic shape, thus if knowing the coordinates of three points on that curve, the corresponding equation can be derived. Here, the three points are chosen randomly based on the graph:

point 1= (108,8000)

point 2 = (145, 6450)

point 3 = (217,0)

and use the second order 'polyfit' function in MATLAB, the equation becomes:

n = -0.4375 M^2+68.8048 M+5672.5080, when $108 \leq M \leq 217$

Apply the gear ratio, simplify the equation:

$$g\acute{\theta}_e \leq -0.4375\left(\frac{M_e}{g}\right)^2 + 68.8048\frac{M_e}{g} + 5672.5080, \text{ when } 108 \leq \frac{M_e}{g} \leq 217$$

$$\frac{M_e}{g} \leq \frac{-68.8048 - \sqrt{68.8048^2 - 4 * (-0.4375) * \left(5672.5080 - g\acute{\theta}_e\right)}}{2 * (-0.4375)}$$

$$14660.9895 - 1.75g\acute{\theta}_e$$

$$\frac{M_e}{g} \leq \frac{68.8048 + \sqrt{14660.9895 - 1.75g\acute{\theta}_e}}{0.875}$$

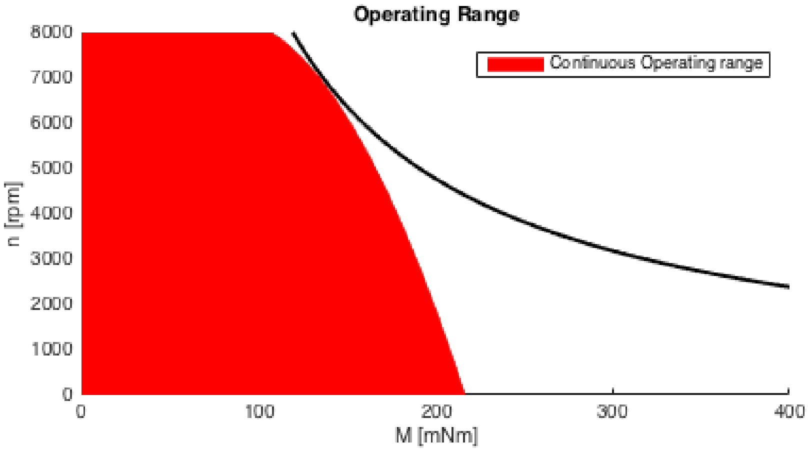


Figure 2.3.3.2.2: Operating ranges plotted in MATLAB

The figure above shows how the region looks like in MATLAB. Compare with the original graph, this estimate is good enough. Considering the gear ratio, the constraints for this problem will become:

1) $-g \leq 0$
2) $g\acute{\theta}_e \leq n_{max}$
3) $\frac{M_e}{g} \leq \frac{68.8048+\sqrt{14660.9895-1.75g\acute{\theta}_e}}{0.875}$

Design variables and parameters

Table 2.3.3.3.1: Sample Torques and Velocities Input from gait pattern

T [s]	0.0077	0.0155	…
$M_{e,1}$ [Nm]	0.1550	0.9049	…
$M_{e,2}$ [Nm]	0.0071	-2.2971	…
$M_{e,3}$ [Nm]	0.2321	-0.3686	…
$\acute{\theta}_{e,1}$ [rad/s]	0.8498	2.9000	…
$\acute{\theta}_{e,2}$ [rad/s]	-0.0996	-0.9923	…
$\acute{\theta}_{e,3}$ [rad/s]	1.4761	2.8117	…

Table 2.3.3.3.2: Parameter Values for the selected Motor

Name	Symbol	Value
Rotor Inertia	J	44 gcm^2
Torque Constant	K_m	74.9 mNm/A
Speed Constant	K_n	127 rpm/V
Resistance	R	0.767 Ω
Max. Speed	n_{max}	8000 rpm
Max. continuous Torque	M_{max}	217 mNm

The design variables for this subsystem are the gear ratio $\acute{g}$ for each motor (g1, g2, g3). Torque M_e and velocity $\acute{\theta}_e$ will be loaded as the input, they are 3xN matrices from the gait pattern and the time for this motion will also be provided. The parameters for the motor are the rotor inertia J, torque constant K_m, speed constant K_n, resistance R, maximum continuous torque M_{max} and the maximum speed n_{max}.

Summary Model

For a fixed gait pattern, the function for this subsystem is to run the torque/velocity after gear and output the gear ratio which minimize the total electrical power. Those gear ratios will be what the optimizer changes. The constraints come from the mechanical and thermal limits for the specific motor model.

<u>Objective function:</u>

Min: Total Input Power = function (g1, g2, g3)

$$P_{total} = \sum_{timespan} (P_{el,1} + P_{el,2} + P_{el,3})$$

where $P_{el,t} = \frac{d\acute{\theta}_{e,t}}{dt} \cdot \left(\frac{d\acute{\theta}_{e,t}}{dt} \cdot \frac{J^2R}{K_m{}^2} + \frac{J\acute{\theta}_{e,t}}{K_nK_m}\right) \cdot g^2 + \frac{M_{e,t}{}^2R}{K_m{}^2} \cdot \frac{1}{g^2} + \frac{M_{e,t}}{K_m} \cdot \left(2 \cdot \frac{d\acute{\theta}_{e,t}}{dt} \cdot \frac{JR}{K_m} + \frac{\acute{\theta}_{e,t}}{K_n}\right)$

Constraints:

1) $-g \le 0$
2) $g\acute{\theta}_{e,i} \le 8000$
3) $\frac{M_{e,i}}{g} \le \frac{68.8048+\sqrt{14660.9895-1.75g\acute{\theta}_{e,i}}}{0.875}$

2.3.4 Model Analysis

Base on the graph of operating range (constraint region), this problem is obviously bounded and will have an optimal result in that well-bounded region. By observation, the gear ratios for three motors are independent to each other, thus it can be solved one by one. For each motor, let's use a KKT condition to find the optimal analytical solution.

KKT Conditions:

$$P_{el,t} = \frac{d\acute{\theta}_{e,t}}{dt} \cdot \left(\frac{d\acute{\theta}_{e,t}}{dt} \cdot \frac{J^2R}{{K_m}^2} + \frac{J\acute{\theta}_{e,t}}{K_nK_m}\right) \cdot g^2 + \frac{{M_{e,t}}^2R}{{K_m}^2} \cdot \frac{1}{g^2} + \frac{M_{e,t}}{K_m} \cdot \left(2\frac{d\acute{\theta}_{e,t}}{dt} \cdot \frac{JR}{K_m} + \frac{\acute{\theta}_{e,t}}{K_n}\right)$$

Let's define:

$$A_t = \frac{d\acute{\theta}_{e,t}}{dt} \cdot \left(\frac{d\acute{\theta}_{e,t}}{dt} \cdot \frac{J^2R}{{K_m}^2} + \frac{J\acute{\theta}_{e,t}}{K_nK_m}\right)$$

$$B_t = \frac{{M_{e,t}}^2R}{{K_m}^2}$$

$$C_t = \frac{M_{e,t}}{K_m} \cdot \left(2\frac{d\acute{\theta}_{e,t}}{dt} \cdot \frac{JR}{K_m} + \frac{\acute{\theta}_{e,t}}{K_n}\right)$$

then

$$P_{el,t} = A_t \cdot g^2 + B_t \cdot \frac{1}{g^2} + C_t$$

$$P_{el,total} = \sum P_{el,t} = \sum A_t \cdot g^2 + \sum B_t \cdot \frac{1}{g^2} + \sum C_t$$

To simplify the problem, the third constraint will be replaced by a straight line instead of the quadratic curve.

$$\frac{M_{e,i}}{g} \leq -0.013625 \cdot g\acute{\theta}_{e,i} + 217$$

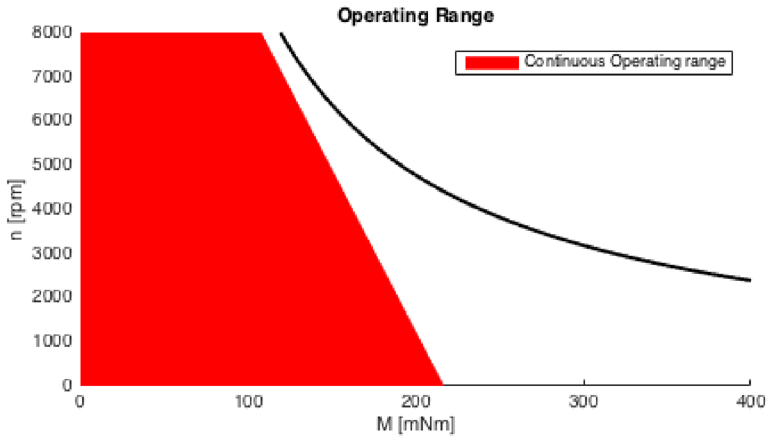


Figure 2.3.4.1: Simplified Constraint Region

Besides, the second constraint can be rewritten as:

$$g\dot{\theta}_{e,max} \leq 8000$$

The Lagrangian becomes:

$$L = \sum A_t \cdot g^2 + \sum B_t \cdot \frac{1}{g^2} + \sum C_t + \mu_1(-g) + \mu_2\left(g\dot{\theta}_{e,max} - 8000\right)$$
$$+ \sum \mu_t \left(\frac{M_{e,t}}{g} + 0.013625 \cdot g\dot{\theta}_{e,t} - 217\right)$$

Then the KKT condition is:

$$\frac{\partial L_t}{\partial g} = 2\sum A_t \cdot g - 2\sum B_t \cdot \frac{1}{g^3} - \mu_1 + \mu_2\dot{\theta}_{e,max} + \sum \mu_t \left(-\frac{M_{e,t}}{g^2} + 0.013625 \cdot \dot{\theta}_{e,t}\right) = 0$$

$$\mu_1 \geq 0,\ \ \mu_2 \geq 0,\ \ \mu_t \geq 0$$

$$\mu_1 \cdot (-g) = 0$$

$$\mu_2 \cdot \left(g\dot{\theta}_{e,max} - 8000\right) = 0$$

$$\mu_t \cdot \left(\frac{M_{e,t}}{g} + 0.013625 \cdot g\dot{\theta}_{e,t} - 217\right) = 0$$

Since the gear ratio should be positive in practice , the first constraint will not be active, which means $\mu_1 = 0$; Define $D_1 = \sum A_t$, $D_2 = \sum B_t$, the equation can be simplified as:

$$2D_1 \cdot g^4 - 2D_2 + (\mu_2 \cdot \dot{\theta}_{e,max}) \cdot g^3 + \sum \mu_t\big(-M_{e,t}g + 0.013625 \cdot g^3\dot{\theta}_{e,t}\big) =0$$

Thus,

$$2D_1 \cdot g^4 + (\mu_2 \cdot \dot{\theta}_{e,max} + 0.013625 \sum \mu_t\dot{\theta}_{e,t}) \cdot g^3 + \sum \mu_t(-M_{e,t})g = 2D_2$$

$$\mu_2 \cdot \big(g\dot{\theta}_{e,max} - 8000\big) = 0$$

$$\mu_t \cdot \left(\frac{M_{e,t}}{g} + 0.013625 \cdot g\dot{\theta}_{e,t} - 217\right) = 0$$

$$\mu_2 \geq 0,\ \ \mu_t \geq 0$$

where

$$D_1 = \sum A_t = \sum \frac{d\dot{\theta}_{e,t}}{dt} \cdot \left(\frac{d\dot{\theta}_{e,t}}{dt} \cdot \frac{J^2R}{{K_m}^2} + \frac{J\dot{\theta}_{e,t}}{K_nK_m}\right)$$

$$D_2 = \sum B_t = \sum \frac{{M_{e,t}}^2R}{{K_m}^2}$$

The analytical equation is difficult to solve here, the value will be calculated in MATLAB with some root finding methods and will compare with the results coming from ‘fmincon’.

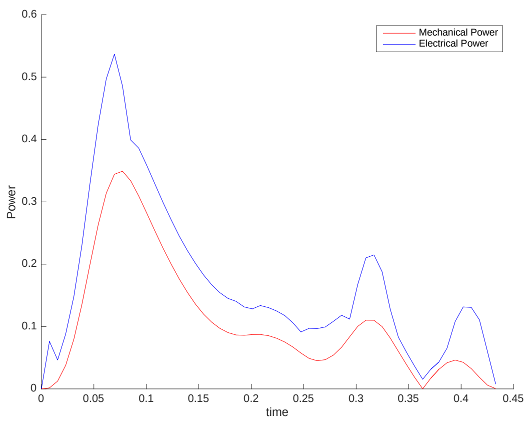
2.3.5 Optimization Study and Discussions

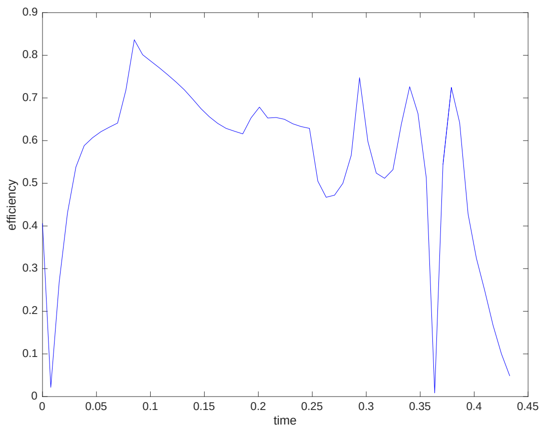
Matlab function ‘fmincon’ are used to implement the optimization for this problem. The main file for this subsystem will initialize parameters and functions to optimize. Then it will call the optimization tool (fmincon) and finally plot the results. The optimized results from this subsystem will be used to improve the whole performance of the robot.

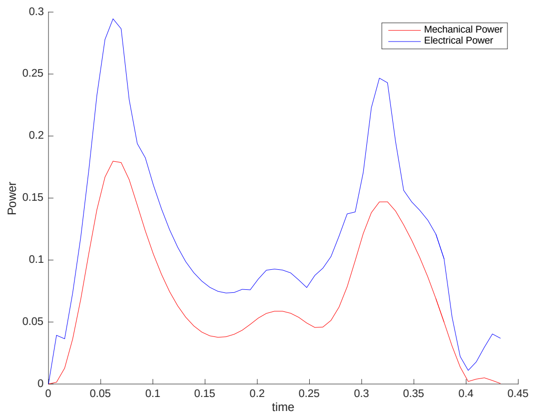
The initialization includes the detailed information about the motor, the torque and velocity from the gait pattern and the corresponding moving time. All the values are read into MATLAB. In ‘fmincon’, the ‘active-set’ algorithm are used, the tolerance of the function is 1e-5, the maximum iteration number is 10,000, the maximum function evaluation is 10,000. After running the optimization, the active constraints can be found from ‘active-set’ algorithm, and with that information, the KKT conditions is simplified and solved. Here is a test result from ‘fmincon’, the input torque and velocity are some real data from earlier experiment (see in Appendix).

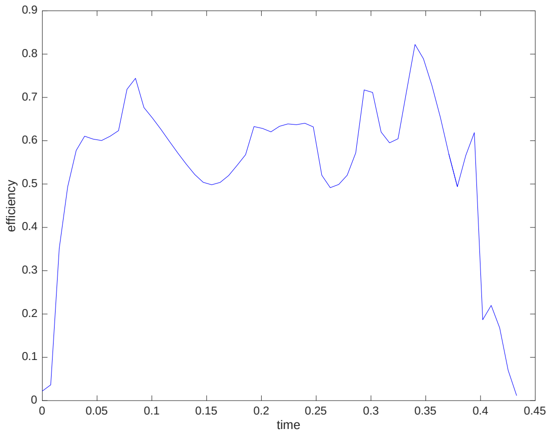
Table 2.3.5.1: Results from fmincon

	Initial guess	Gear ratio	Fval	Iteration Number
Motor 1	10	9.8020	9.7473	4
Motor 2	10	17.3499	6.7784	7
Motor 3	10	6.0229	3.9252	9









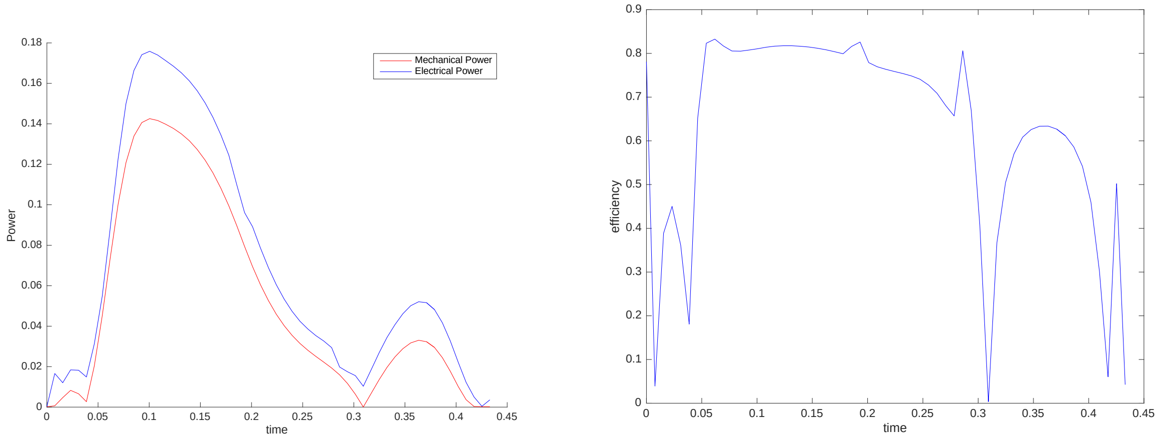


Figure 2.3.5.1: Mechanical and Electrical Power versus Time and Efficiency versus Time

The total length of test data is N=57. From the table, it can be found that 'fmincon' reached some optimal values within a small number of iterations. The gear ratios are all positive and larger than one which is true in reality. Also, the mechanical and electrical power are plotted on the same figure. As can be seen, the efficiency = mechanical / electrical power is always smaller than 1, which make sense. Based on 'fmincon' result, all the inequality constraints are inactive, which means all the μ_t are zero, thus the KKT is simplified as:

$$D_1 \cdot g^4 = D_2$$

$$g = \sqrt[4]{\frac{D_2}{D_1}}$$

Table 2.3.5.2: Comparison of fmincon and KKT condition results

	fmincon		KKT Conditions	
	Gear ratio	Power (W)	Gear ratio	Power (W)
Motor 1	9.8020	9.7473	9.8020	9.7473
Motor 2	17.3499	6.7784	17.3502	6.7784
Motor 3	6.0229	3.9252	6.0229	3.9252
Total:		20.4509		20.4509

Then the solution from KKT conditions can be calculated and values are given here. As can be seen, MATLAB fmincon gave the same answer as KKT conditions, which validates the previous optimization results. Note that here the input are some test data, the values can be changed in the future and the constraints are not always inactive. Once getting some active constraints, the μ_t for the those active constraints should have some positive values. In this case, the KKT condition will become more complex. The g value can be then calculated using the ‘root’ function in MATLAB. In conclusion, the motor optimization model works well in this subsystem.

5 Subsystem – Gait Pattern Generation (Hope Yao and Nathan)

5.1 Problem Statement

The goal of this subsystem is to find an ideal gait pattern that is both stable and minimizes the power expenditure of the actuators. We have defined the dynamic equations of motion for a simplified version of the robot. We will simulate the robot's dynamic motion using a simulator created in Matlab's Simulink program. This simulated robot can be commanded with a gait pattern defined by several continuously adjustable parameters. Using this gait pattern and some initial conditions the simulation can be run and the state information of the robot can be extracted across the span of a step. This state information is used to assess the stability of the robot at the end of its step. The stability of the robot is a constraint of the optimization problem. The power output of the gait pattern may also be extracted from the results of the dynamic simulation, this will be the function we minimize with our optimization routine.

5.2 Nomenclature:

A "gait pattern" is defined in this paper as a set of curves that define the way a walking robot or animal will move during a single cycle of its gait. These curves include the positions of the leg, and forces required during gait.

Gait percent is a parameter that varies linearly with time from 0 to 100% and repeats.

The length of time taken to reach 100 percent is the "step time". With step time defined the velocity of the leg can also be defined from the gait pattern.

5.3 Mathematical Model

For this project it is very important to simplify the problem as much as possible. In order to do this, the robot has been simplified to a 2d model. The model consists of a body and two legs. Furthermore the degrees of freedom of each leg have been reduced down to 2: a length and an angle. Figure 1 shows a visualization of the simplified model. This model was built in a dynamic simulator created by SpringActive inc. After the dynamic equations were generated and plugged into the model, the next task was to select a method to parameterize the gait.

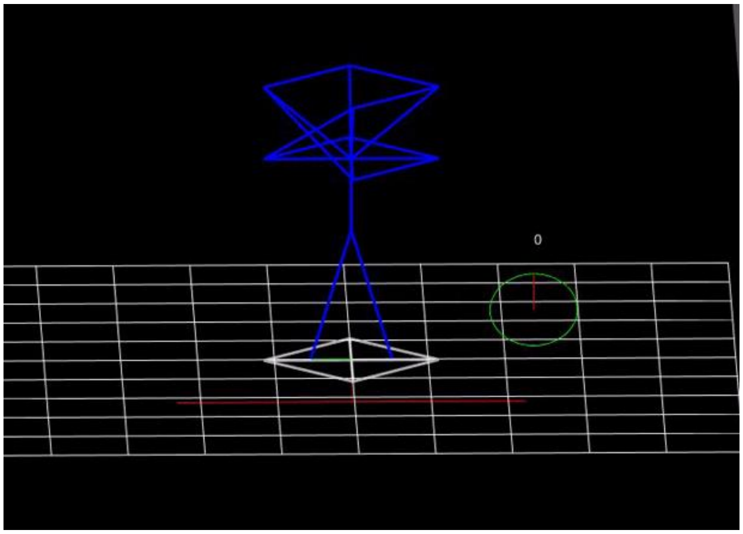


Figure 1: Simplified robot model

Objective Function

The optimization routine attempts to minimize actuator power while a tune parameters that define a robot’s gait cycle (See section entitled: “Modeling the System”). Every time the objective function is called the dynamic simulator runs through the robot’s gait cycle once, starting at a given initial robot state. The dynamic simulator outputs the robot’s state information and the simplified actuator forces required to achieve the gait pattern. From the actuator forces the power output may be calculated. The objective is be to minimize this power. It should be noted that this assumes that there exists some gait pattern that allows stable steady state walking for the robot. In reality a more advanced walking control may be needed to keep a real robot balanced. These advanced walking controllers do not have a set pattern but some set of rules that balance the robot (see: zero moment point control for example). As stated above we are not using advanced walking control logic, but a simple, repeated, open-loop pattern. In the end, this approach was not suitable to converge to a truly stable gait pattern.

Constraints

The main constraint for this subsystem is that the gait pattern selected needs to be a stable gait pattern. Stability, however, can be a tricky thing to assess in a walking robot. There are several ways to look at stability. The simplest way is to enforce some stability criteria upon every point of the gait cycle. This is how zero moment point controllers define stability (keeping the “zero moment point” inside the footprint of the robot at all times. By this definition, most biological gaits are *not* stable. A potentially better stability criteria would be cyclic stability. This looks at the stability of the gait cycle as a whole. One way to assess this is to see if the gait of the robot will converge to an exact repetition of a closed trajectory [1]. This means that a constraint for our robot could be that the state at the end of the step must be exactly the same as the state at the beginning.

Design variables and parameters

The hip angle was modeled using a second order Fourier series. The hips was then forced to be 180 degrees out of phase from each other. In all only 5 parameters are needed to fully define the hip pattern. The equation for this is as follows:

$$\theta = X_1 sin(\theta) + X_2 cos(\theta) + X_3 sin(2\theta) + X_4 cos(2\theta) + X_5$$

Where X is the array of parameters that the optimization tunes, and θ is a vector of length n that increases linearly from 0 to 2π. The length n for our study was 100 which was a sufficient number of data points to create a lookup table for the open loop controller.

While using a Fourier series on the hip angle seemed to work very well using a Fourier series on the leg length produced a set of patterns that didn't seem "gait-like" at least intuitively speaking. A parameterization was selected that allowed a long constant section in the beginning of the gait pattern, and then a second section that is a sum of sinewaves. This allows the leg to be a constant length while the foot is on the ground and then lifted up for the swing portion, while only using 4 more parameters. See figures for an example gait pattern. The exact equation for the leg length is as follows:

$$\theta = linspace(0,2\pi,m)$$

$$GP = [0linspace(X_9, 100, □)];$$

$$l = \left[1X_6 sin(\theta) + X_7 sin\left(\frac{\theta}{2}\right)\right] X_8$$

Where m is the number of data points in the swing portion of the curve, X is the parameter that the optimization tunes, l is the length of the leg, and $GP$is the gait percent. After the gait pattern is generated with these equations, a spline function is used to smooth it out. This results in the gait patterns you see in the following figure.

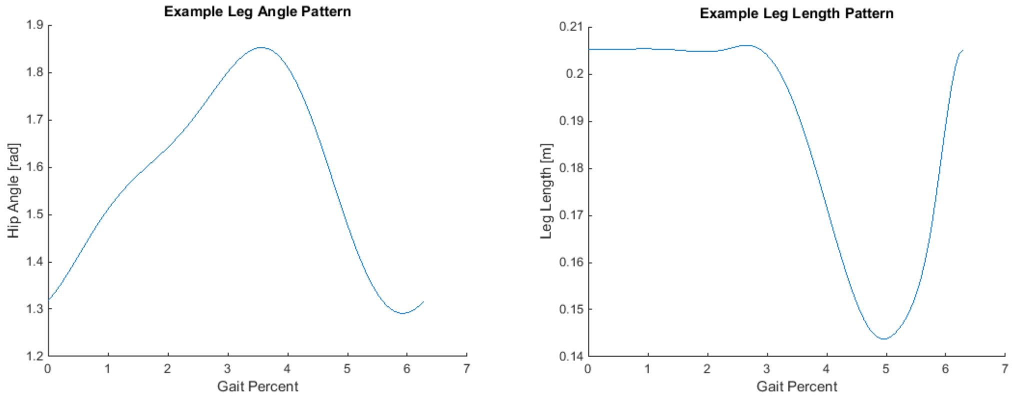


Figure 2:Example gait pattern

In addition to the parameters needed to generate the gait pattern, a one other parameter was made "tunable" for the optimization. This parameter was the "step time" which is the duration of time that passes during one step.

5.4 Optimization Study

Initial attempts to find a stable, low power gait patterns used gradient methods such as fmincon. There was a problem with this kind of optimization routine because of the oscillatory nature of the system. Gradient information obtained from partial difference functions does not find a good search direction. Two methods were used to bypass this problem. One method forced the partial difference function to use a larger perturbation for the system. This allowed the gradient method to converge on a solution. Another solution to this problem was using CMA-ES. This multi-start, non-gradient method searched a larger portion of the parameter space and found solutions with lower function values than the modified gradient method. Figure 3 shows an example of CMA-ES converging to a solution. The algorithm was able to reduce the objective function by an order of magnitude. Several example gaits are attached in the form of animations. It is difficult to quantify how "good" these gait patterns are or not, as there is not much to compare them to. They could be compared to human gait, but this kind of comparison has not been made for the purpose of this paper. In addition this makes the assumption that human gait is going to be an optimal solution to this problem. This is a bad assumption because human gait is optimized around the actuation system available to it: its muscular-skeleton system. In the following section the results of a parametric study are presented, which give an idea of the solution space that is being explored here.

5.5 Parametric Study

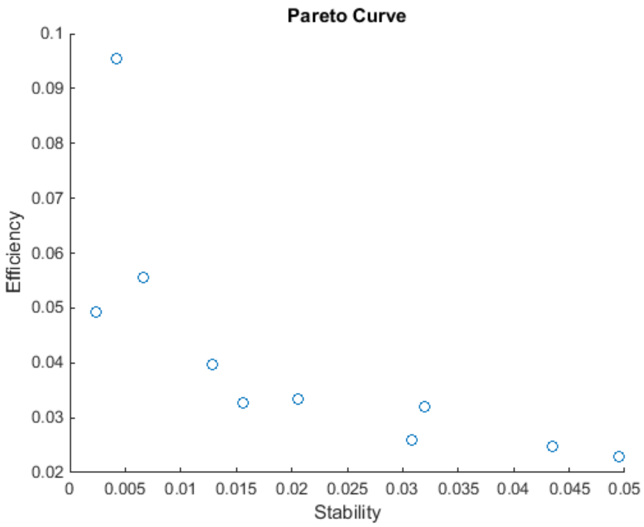


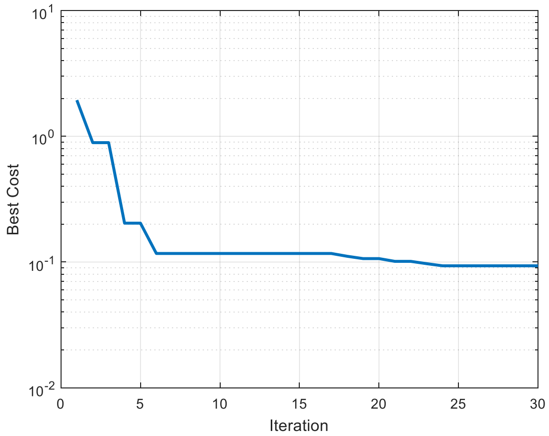
Figure 3: Pareto Curve

A parametric study has been completed on this subsystem. This study entails optimizing a cost function with various weighting between two or more conflicting goals. In this case the goals are: one, the stability of the gait and, two, the efficiency of the gait. Figure 3 shows the Pareto curve. This surface represents optimization cost function that is a sum of the two cost functions with weighting varied from a 1 to 10 ratio to a 10 to 1 ratio. There are ten points in all. Each of which represents the output of CMA ES algorithm which runs a dynamic simulation hundreds or thousands of times. This kind of study is very time and processing intensive. In the future it could be useful to add more points to this curve but the general shape is pretty obvious. Also, a video is included that shows the corresponding gait of all of these 10 points on the surface. The videos are in order from most stable to most power efficient.

5.6 Discussion of Results

This subsystem ended up being the most challenging subsystem. The main obstacle to overcome arose when attempting to constrain the solutions to “stable gaits”. This is due in part because it is difficult to define “stable gait” mathematically. Many methods were attempted including minimizing change in kinetic and potential energy, minimizing change in center of mass height, and minimizing change of state throughout the gait cycle. All of these methods failed for various reasons including failure to converge at all, and converging to a low function value that did not represent a real gait. To give one specific example a problem with minimizing change in potential and kinetic energy is that during a free fall there is no net change in the sum of these energies, so the gait that was found was much more of a fall than a step. In the end it was comparing the change of initial and final state of the step that returned stable solutions consistently. In future research it may be desirable to develop a tunable control strategy that always results in stable gaits. This would reduce the number of iterations of an optimization routine dramatically by not having to iterate through unstable gait patterns.

Another limitation of this optimization study is in the choice of initial state of the robot. This choice was basically an arbitrary one, all the gait patterns selected are only stable given this initial state. The choice was made to use this simplification for a number of reasons. A future goal is to find some efficient way to select an optimal gait pattern given an arbitrary state input. This will allow some kind of extrapolation of the results to a more general case.

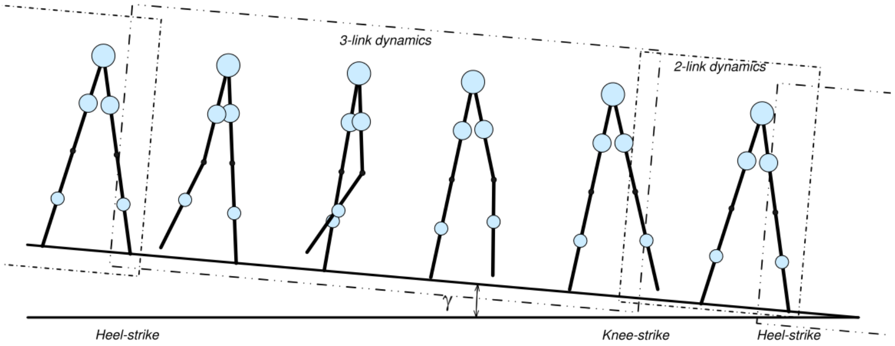


5.5 Passive dynamic walking (Hope Yao)

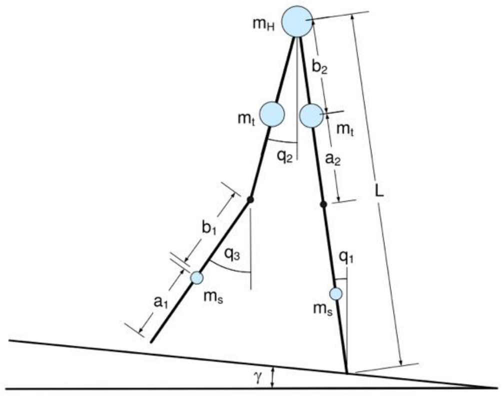
In this part, I will investigates another simplified walking, namely passive dynamic walking.

Numerical model

Here passive dynamic walking is chosen as the problem to be optimized. Passive dynamic walking is a specific simplified walking scheme that models human walking down a slope. The structure of the robot is simplified into a linked chain and all masses are lumped to the gravity center of different parts of the leg. By passive it means that there is no control input to the system at all, the robot walk on its own merely based on gravity pull. There are two different walking state during one walking cycle. The first one is called 3-linked dynamics, which corresponds when the bipedal has three chains swinging at the point walking just started. The stance leg won't bind at this state, so there only exists three swinging chains: no-stance thigh, non-stance thigh and stance leg. 3-linked chain state will evolving into 2-linked chain state after non-stance knee and non-stance thigh rotate to the same angle and impact. This is called knee strike event. After the knee strike event, the non-stance knee and non-stance thigh will bind together and rotate as whole. This is when the so called 2-lined dynamics starts. Because there is no control input, all potential energy is dispersed at the knee strike and heel strike event.



Only three variable are needed to define the state of this model. Here I choose the orientation angle of different non-stance thigh, non-stance shank and stance leg as the state variables. Masses are lumped and the location of them are also design variables. The swing state is governed by non-linear second order structure dynamics and the impact event could be described by momentum conservation. Detailed formula could be find in [].



Integration block in Scipy package is used here to simulate this problem.

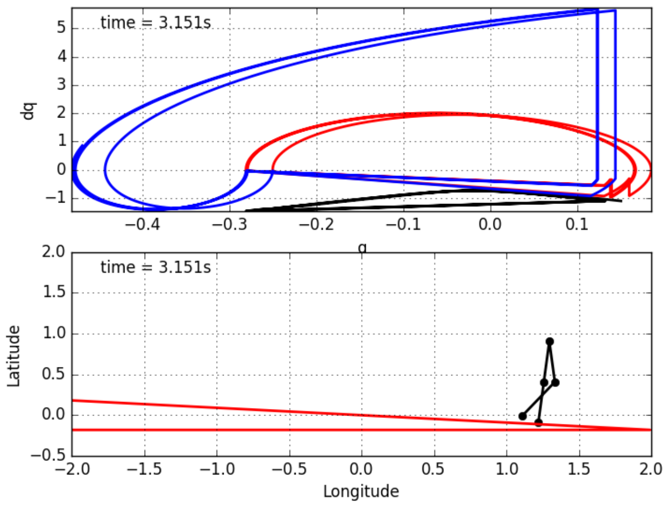
Optimization

There are two objective functions, the first one is the speed of the robot and the second one is its stability. Speed of the bipedal is measured by the distance of one step cycle and the stability is measured by the difference of state variables at the beginning of a step cycle and the end of a step cycle. There are seven variables to be optimized. Three for initial state and four for the location of lumped mass. The initial state variables could be viewed as control variables while the mass location could be viewed as structure variables. So both design and control variables are going to be optimized. Constrains here are set as q1 in (0.15, 0.2), q2 in (-0.3,-0.2), q3 in (-0.3, 0.2), a1 and b2 in (0.2, 0.4), a2 and b1 in (0.1, 0.26). Besides, there are extra constrains on the total length of the leg, it is set in the range of (0.9, 1).

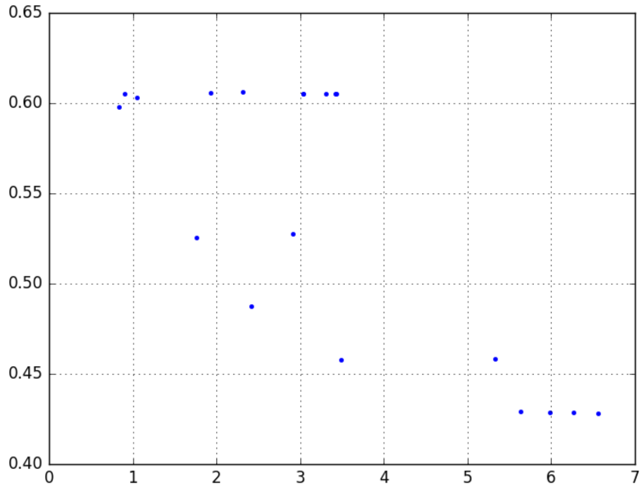
Optimization block in Scipy package is used here. More specifically, L-BFGS-B method is used to find the minimum solution.

Results and discussions

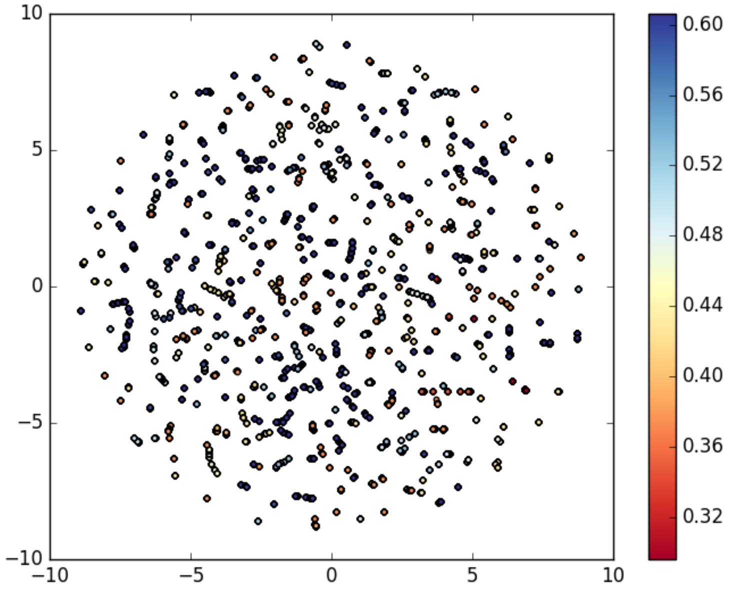
Following figure is a screen shot of the bipedal animation. The first subplot is the periodic orbit in state space. Three different lines stands for three different state variables. It is obvious from the figure that, after some steps, the orbit is approaching a limit cycle.



The Pareto surface of the two objectives are shown in bellow. X axis is the stability, which is the reciprocal of the difference between start and end state of a walking cycle. Y axis is the speed. It is strange that the points are not strictly monotonous decreasing as it should be in theoretical. The reason I think might be the nonlinearity of the passive dynamic walking. It is very difficult to converge to global optimum using the gradient based method for this kind of highly non-linear system (especially with gradient approximation using finite difference). I'm still working on combining global search method like CMA-ES with this simulator and hopefully it will get better result.



During the computation of Pareto surface, the optimizer ran a total of 13540 simulations. And all these simulation results have been recorded. Hoping to find some structure in those data, T-SNE is further incorporated to visualize them. Following figure is a visualization for speed, darker color indicates higher speed. However, many points are so clustered together that they merged into single points after dimension reduction. Because the merge effect, not too much pattern could be find. I think better results could be obtained if global search algorithms like CMA-ES could be used to gather those data.



Future directions

Control signals could be added to the system very easily. If this system becomes controllable, I can use human walking data to make it walk in a bio-mimic way. Moreover, I can even do non-parametric reinforcement learning on it for different terrains.

6 System Integration

For this project it is possible to integrate all of the optimization tools into one optimization routine. This is aided by the fact that each subsystem changes an independent set of parameters of the system. The tool would look something like this:

Step 1: The gait pattern subsystem would select a candidate gait pattern and check whether it is stable.

Step 2: The stable gait pattern would be sent to the geometry optimization. The geometry optimization would select a candidate geometry that satisfies the geometry constraints (R-Conditioning number, and length bounds).

Step 3: This candidate gait pattern and geometry set would then go through the stiffness optimization. This optimization would find the global best spring stiffness matrix using quadratic programming and then output a new, reduced force pattern for the motors.

Step 4: This force pattern, along with the candidate geometry and gait pattern, would go to the gear ratio subsystem. The gear ratio subsystem would find the optimal gear ratio using the KKT conditions, and it would output the motor power. This motor power is the cost function for steps 1 and 2.

Repeat Steps 2-4 until the geometry subsystem has converged. Return result (motor power) to Step 1.

This all in one method would take an incredibly long time to converge, despite the fact that steps 3 and 4 are very fast with their quadratic programming and KKT conditions results. Another method for system integration could be a decomposition approach such as:

Step 1: The gait pattern subsystem would use mechanical power (as opposed to electrical power) as its cost function and ignore the possible improvements from geometry, stiffness, and gear ratio while it is running.

Step 2: The result of the gait pattern subsystem could be sent to the geometry optimization. The geometry optimization would ignore the possible benefits from springs and gear ratios and by using mechanical power (as opposed to electrical power) as its cost function.

Step 3: The result of the geometry and gait pattern optimization could then go through the stiffness optimization. This optimization would find the global best spring stiffness matrix using quadratic programming and then output a new, reduced force profile for the motor subsystem.

Step 4: The results of all of the above optimization routines (gait pattern, geometry and updated force profile.), would then go to the gear ratio subsystem. The gear ratio subsystem would find the optimal gear ratio using the KKT conditions, and it would output the motor power. This motor power is the cost function for steps 1 and 2.

This approach would be much more efficient but it may ignore any synergistic effects that might be present in the AIO method. These synergistic effects result from running the optimizations in parallel. For example, some gait patterns may be more “optimizable” than others for the geometry, compliance, gear ratio subsystems. Similarly some geometries may gain more benefit than others from the spring and gear ratio subsystems. In future work either of these system level optimization approaches could be used to further optimize the design of the leg.

7. ACKNOWLEDGMENTS

This work is the dissertation topic of group member Nathan Cahill. He has been developing the robot for the last 1.5 years with funding from the National Science Foundation Graduate Research Fellowship Program.

8. APPENDIX, KINEMATIC SOLUTION FOR THE LEG MECHANISM

Spring Optimization Mathematical Model Derivations

Equation Derivation:	Comments:	Eq.#:
$F_{total} = F_{springs} + F_{motors}$	Total force at the end effector is composed of the force harnessed by the springs and force generated by the motor	(1)
$F_{motors} = F_{total,motor} - F_{springs,motor}$	Eventually we'll want to obtain the Power of the motors and thus we'll need to rewrite the equation in respect to the motors coordinate system	(2)
$F_{springs,joint} = -K(\theta - \beta) = K\beta - K\theta$	Offset angle (β) will be considered, and is subject to change. Note that spring force is current in the joints coordinate system	(3)
$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_3 \end{bmatrix}$	Gait pattern shows no abduction and adduction at the hip joint and thus $\theta_1$can be set to zero	(4)
$\beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} \beta_1 \\ \beta_3 \end{bmatrix}$	Gait pattern shows no abduction and adduction at the hip joint and thus $\beta_2$can	(5)

be set to zero

$$K = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} = \begin{bmatrix} k_{11} & k_{13} \\ k_{13} & k_{33} \end{bmatrix}$$

Associated hip abduction/adduction springs with can be disregarded since $\theta_2 = 0$. Note that K is a symmetric matrix, and thus $k_{13} = k_{31}$ (6)

$$\begin{aligned} F_{springs,ankle} &= J_{leg}^T(K\beta - K\theta) \\ &= J_{leg}^T K\beta - J_{leg}^T K\theta \end{aligned}$$

First, we are going to map the sprig force to the end effector coordinate system, so that we can couple it with the end effector velocity ($V_{total,ankle}$), which is empirically provided from the gait pattern (7)

$$\begin{aligned} &F_{springs,ankle} \\ &= J_{leg}^T * K\beta - \begin{bmatrix} J_{leg}^T\theta_1 & J_{leg}^T\theta_3 \end{bmatrix} * vec(K) \\ &= J_{leg}^T * K\beta - A_t * vec(K) \\ &= \begin{bmatrix} J_{leg}^T & J_{leg}^T\theta_1 & J_{leg}^T\theta_3 \end{bmatrix} \begin{bmatrix} K\beta \\ -vec(K) \end{bmatrix} = Ax \end{aligned}$$

Utilizing matrix manipulation, we are able to decuple the variables from the parameters by setting $A = \begin{bmatrix} J_{leg}^T & J_{leg}^T\theta_1 & J_{leg}^T\theta_3 \end{bmatrix}$ and $x = \begin{bmatrix} K\beta & vec(K) \end{bmatrix}^T$, where $vec(K)$ is the vectorization of the Kmatrix (8)

$$\begin{aligned} F_{motors,ankle} &= F_{total,ankle} - F_{springs,ankle} \\ &= B - Ax \end{aligned}$$

End effector force (at the ankle) is empirical provided by the gait pattern and is denoted as B here (9)

$$\begin{aligned} F_{motors} &= (J^T)^{-1} F_{motors,ankle} \\ &= (J^T)^{-1}(B - Ax) \\ &= (J^T)^{-1}B - (J^T)^{-1}Ax \\ &= B_m - A_m x \end{aligned}$$

We now can map the coordinate system from the end effector to the motor coordinate system, where B and A are now in motor coordinates, B_m and A_m (10)

$$V_{motors} = (J^T)^{-1} V_{total,ankle} = C_m$$

We denote motor velocity as C_m (11)

$$\begin{aligned} P_{motors} &= F_{motors}^T \cdot V_{motors} \\ &= (B_m - A_m x)^T C_m \end{aligned}$$

Now we can calculate for the motor power (12)

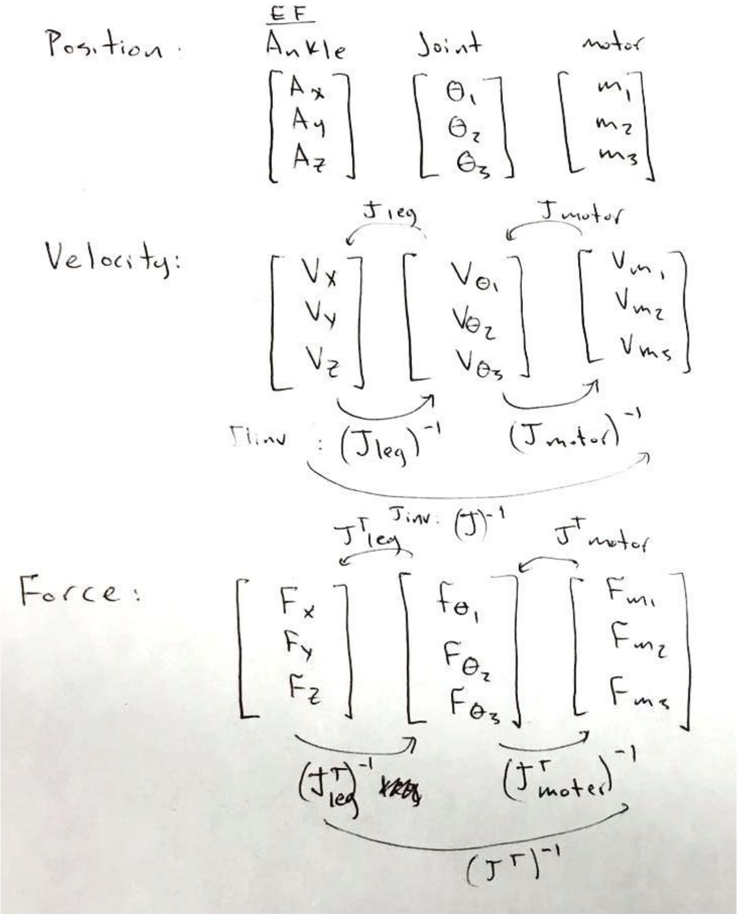
$$\begin{aligned} P_{motors} &= diag(C_m) * (B_m - A_m x) \\ &= C_{md} * (B_m - A_m x) \\ &= C_{md} B_m - C_{md} A_m x \end{aligned}$$

We can rewrite power calculation in the following manner, where $diag(C_m)$ is the diagonal of vector C_m (13)

$$\begin{aligned} P_{motors}^2 &= (C_{md} B_m - C_{md} A_m x)^T (C_{md} B_m \\ &\quad - C_{md} A_m x) \\ &= (B_m^T C_{md} \\ &\quad - x^T A_m^T C_{md})(C_{md} B_m \\ &\quad - C_{md} A_m x) \\ &= B_m^T C_{md}^2 B_m - B_m^T C_{md}^2 A_m x \\ &\quad - x^T A_m^T C_{md}^2 B_m \\ &\quad + x^T A_m^T C_{md}^2 A_m x \\ &= \frac{1}{2} x^T H x + f^T x + c \end{aligned}$$

In order to consider the influence of the total force (B_m), we'll take the power squared and get it the form of a quadratic: (14)

$\frac{1}{2} x^T H x + f^T x + c$, where $H = 2 A_m^T C_{md}^2 A_m$, $f = -2 B_m^T C_{md}^2 A_m$, and $c = B_m^T C_{md}^2 B_m$



The first step taken towards designing this specialized mechanical limb was selecting the structure of the system. We chose to start with a biologically inspired three segment leg with four degrees of freedom (DOF). The joints include a two DOF hip, and a one DOF knee and one DOF ankle (see Figure 6). Figure 4 is an annotated sketch of a side view of a simplified version of the leg. It also includes the angles used to define the position of the leg. This figure only shows a planar simplification of the system and does not show hip abduction/adduction. Figure 6 shows the spatial leg mechanism, but a planar version of the mechanism is evaluated first.

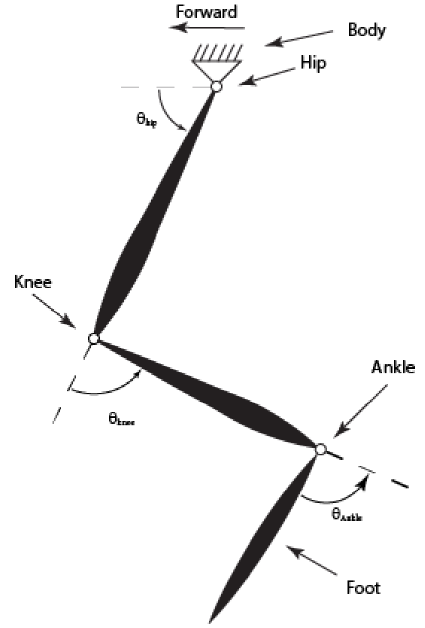


Figure 4. Simplified Planar Serial Leg Structure

This leg structure is often actuated in serial by placing one actuator across each DOF. This means that the heavy actuators are fixed to the leg segments that they actuate, which adds to the inertial losses in the system. This problem is compounded by the fact that the only parameters to adjust in an attempt to optimize this system are the length of the leg segments and the actuators themselves. Our approach allows more flexibility to reduce inertia and place the actuators on the body. We actuate the joints of the leg with motor driven two-link RSS mechanisms which are attached between the body and each leg segment. The R joints of these linkages are fixed to the body through small motors which are used to control the leg. This removes from the leg the motor mass, thereby decreasing the inertial cost of tasks which involve rapidly swinging the mass of the limb back and forth. Figure 4 is an example of a planar version of the mechanism with two motor linkages attached. The red and blue arrows marked τ_1 and τ_2 represent torques applied by each motor. The dotted red and blue arrows represent the torque that each joint would have to produce to provide a force at the end effector equivalent to what would be produced by τ_1 and τ_2. If a serial chain system was designed, a larger motor and torque would be needed at the hip as compared to the parallel actuation.

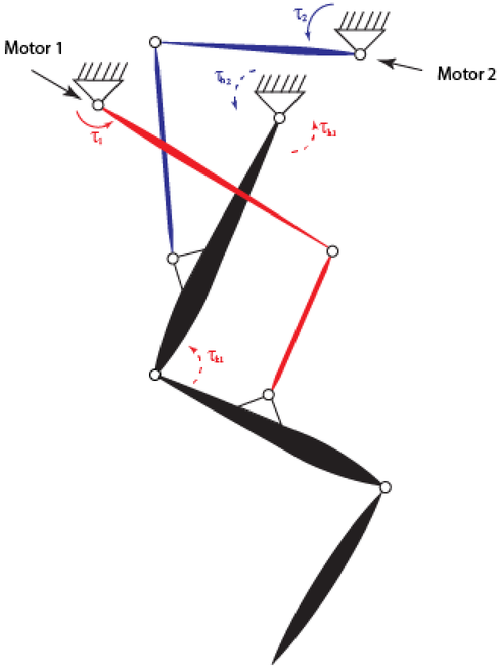


Figure 5. Annotated Drawing of the Planar Leg Structure with Parallel Actuators.

The actual leg is not a planar mechanism as the ones shown above; it has a third DOF allowing hip adduction/abduction. This will allow the quadruped extra maneuverability and balance. This will give the robot added capabilities such as side-stepping, (see Figure 6).

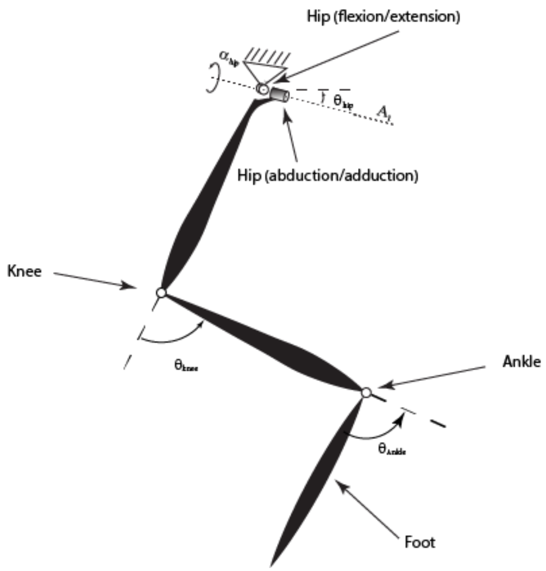


Figure 6. Annotated Drawing of the Serial Leg Structure with All Joints Included.

Figure 6 shows the serial portion of leg structure as it was used in this application. Notice the extra revolute joint at the hip allowing abduction/adduction.

Another feature of this leg design is its ankle joint. This allows the ankle to switch between a spring joint during stance and a passively controlled four bar parallelogram mechanism during swing. Using digitized high speed data of a cheetah running we made a mechanism that would lift the foot during swing in the same way as the cheetah. When force is applied to the foot, however, an elastic member which plays a similar role to the Achilles tendon of humans is engaged and makes the ankle a passive spring joint. This allows the entire leg to be controlled with only three actuators.

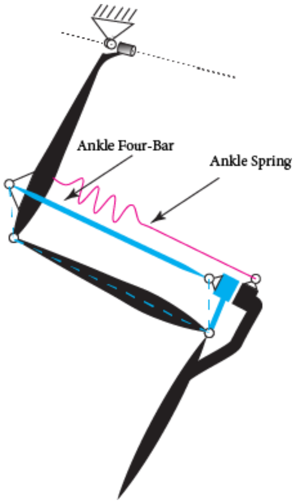


Figure 7. Annotated Sketch of the Ankle Mechanism. When a force is applied to the foot and it rotates clockwise, the ankle joint becomes a spring loaded joint, but in the absence of the force, the four-bar mechanism engages the foot and controls its position during the swing phase of gait.

Configuring and implementing these actuators in a workable design presented several engineering problems. The first was that there are several possible configurations to select from. These are detailed in Tables 1-3: Configurations 1 and 2. These various configurations will be outlined in the following text.

Table 1. LIST OF ALL POSSIBLE ACTUATOR CONFIGURATIONS

	Thigh	Shank
Config 1	2 Actuators	1 Actuator
Config 2	1 Actuator	2 Actuators
Config 3	0 Actuators	3 Actuators

The authors did not want to create an over actuated system, as this adds unnecessary complication to the system at this stage of the project. For example, one can choose to connect zero, one, or two actuators to the thigh but not all three. This is because the hip joint only allows two DOF: flexion/extension and abduction/adduction. If three actuators are connected to this segment one of them will be dependent on the position of the other two. Additionally, the knee joint would be entirely unactuated (Or a fourth motor could be added to the system to actuate the knee, but this would create an over-actuated system: 3 degrees of freedom and 4 actuators).

In Configuration 1, two actuators are connected to the thigh of the robot and one is connected the shank. Table 2 describes how each motor affects the leg joints in this configuration, and Figure 6 is an annotated drawing of the configuration.

Table 2: BREAKDOWN OF CONFIGURATION 1

	Actuator 3 - Thigh	Actuator 1 - Thigh	Actuator 2 - Shank
Hip Ad/ Ab	Actuated	Actuated	Actuated
Hip Flex/ Ext	Actuated	Actuated	Actuated
Knee Flex/ Ext	Un-actuated	Un-actuated	Actuated
Ankle Flex/ Ext	Un-actuated	Un-actuated	Un-actuated

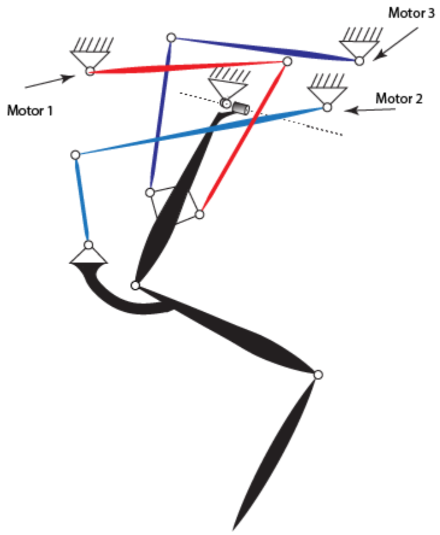


Figure 8. Annotated Sketch of Configuration 1 in three dimensions.

In Configuration 2 (Figure 9), one motor is connected to the thigh and two motors are connected to the shank. Table 3 describes how each motor affects the leg joints in this configuration, and Figure 7 is an annotated drawing of the configuration.

Table 3: Breakdown of Configuration 2

	Actuator 3 - Thigh	Actuator 1 - Shank	Actuator 2 - Shank
Hip Ad/ Ab	Actuated	Actuated	Actuated
Hip Flex/	Actuated	Actuated	Actuated

Ext			
Knee Flex/ Ext	Un-actuated	Actuated	Actuated
Ankle Flex/ Ext	Un-actuated	Un-actuated	Un-actuated

After some analysis Configuration 2 was shown to be significantly better than Configuration 1 when it came to increasing the conditioning number of the leg Jacobian (see design of physical prototype section).

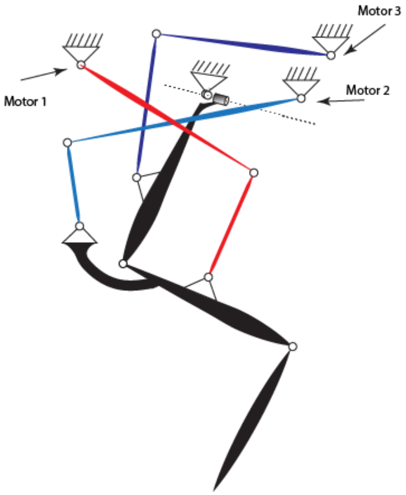


Figure 9. Annotated sketch of Configuration 2 in three dimensions. This configuration allowed a much greater usable workspace.

Kinematics

Another engineering problem faced in the design and implementation process, was the selection of the geometric properties of the linkages. The kinematics of the leg mechanism would have to be solved to gain a scientific understanding of the problem. In fully serial robot limbs the forward kinematics are commonly used and tend to be a simple problem to solve.

The inverse kinematics for a serial mechanism are not as simple. However, in parallel robots the inverse kinematics problem tends to be simpler, and the forward kinematics more complex. Our design is a combination of the two; therefore, challenges arise when solving either forward or inverse kinematics. In the end the inverse kinematic equations were solved. A detailed description of the solution to the inverse kinematic equations is given.

The first step in this method of solving the equations for this robot was to break the problem up into two parts: the serial portion of the leg and the parallel portion of the leg.

Serial Kinematics. The serial portion of the robot limb is shown in Figure 10. Notice that the foot is left out of this diagram. This greatly simplifies the kinematic solution. In fact if the foot and ankle angle are left in the kinematics, there are infinite solutions to the inverse kinematic equations. This also would make the Jacobian matrix noninvertible. Recall that for this system, the ankle is controlled by a spring during stance and a mechanism during swing. For stance this means that with a desired force output the ankle joint will need to be at some known angle. For swing, the position of the foot will be defined by the thigh and shank leg segments. In both cases controlling the ankle joint's position will be enough to fully define the leg. Figure 10 shows the thigh and the shank, along with all of the angles needed to fully define their position in space.

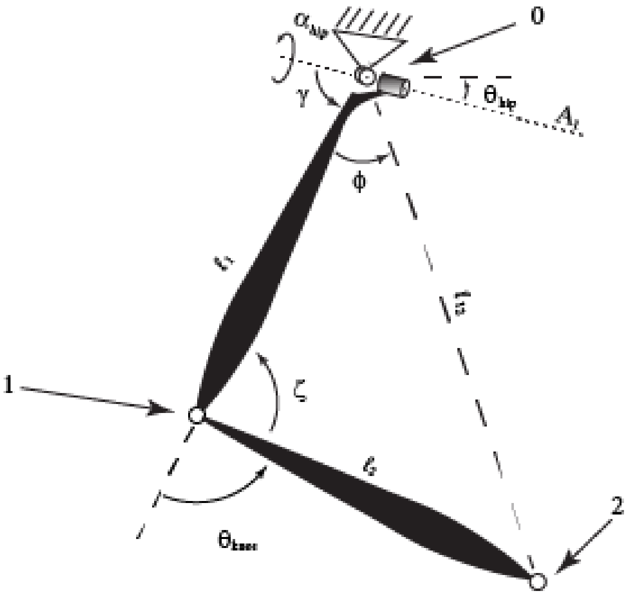


Figure 10. Serial Limb Annotated Sketch

From Figure 10 it can be seen how the two degree of freedom hip connects the thigh to the body. The first rotation $\square_{Hip}$ controls hip flexion and extension, the second rotation, $\square_{Hip}$, controls hip abduction and adduction. The third rotation is $\square_{knee}$, it controls knee flexion and extension. The goal is to determine these angles given a three dimensional ankle joint position and the geometric properties of the leg. The first step is to find the hip and knee angles. We know that the axis$\acute{A}_1$ lies in the plane that contains the thigh and the shank sections of the leg. We can define axis one as a function of $\square_{Hip}$:

$$\acute{A}_1 = \begin{bmatrix} cos(\theta_{Hip}) \\ -sin(\theta_{Hip}) \\ 0 \end{bmatrix} \qquad (1)$$

The dot product of $\acute{A}_1$ and the vector between the hip and the ankle $(\acute{O}2)$is defined:

$$\acute{O}2 \cdot \acute{A}_1 = |\acute{O}2||\acute{A}_1|cos(\gamma + \phi) \qquad (2)$$

Note that γ and ϕ are angles shown in Figure 8. The angle γ is a constant geometric property of the leg and ϕ can be easily found using the Law of Cosines since the three lengths of the triangle that contain it are already known. Since $\acute{A}_1$is a function of $\square_{Hip}$, this equation can be solved symbolically for $\square_{Hip}$ (using symbolic math software such as Matlab).

Finding $\square_{Knee}$ is just a matter of solving the triangle formed by points 0, 1, and 2 using law of cosines to find the inner angle and subtracting that from pi to determine $\square_{Knee}$.

The angle $\square_{Hip}$ is the arctangent of the perpendicular distance from $\acute{A}_1$ to the ankle over the height of the ankle coordinate:

$$\alpha_{HIP} = atan2\left(\frac{-y_2}{x_2 sin\theta_{Hip} + z_2 cos\theta_{Hip}}\right) \qquad (3)$$

where, $x_2, y_{2,}$ and $z_{2,}$, are the *x*, *y*, and *z* coordinates of the ankle joint (joint 2 in Figure 8).

Parallel Kinematics. The next part of solving the inverse kinematics of the leg is solving the parallel portion of the leg. This means, given the leg angles found in part 1 of the kinematics, find the motor angles. To understand the solution to this problem, the two link mechanism needs to be defined more specifically. It is an "RSS" linkage. The joint fixed to the leg segment is a spherical joint. The middle joint in the linkage (joint A_0) is also a spherical joint. Joint A is the motor. Link BA_0 will be called the connecting rod and link AA_0 will be called the control arm. The control arm rotates about the motor which is fixed to the body. Joint A_0 is therefore confined to rotate in a circle centered at point A and perpendicular to the motor axis U. Point B is fixed to the leg. The angles of which are known from the analysis of the serial kinematics. This means the position of B can be found by using a rotation matrix. Therefore the control arm is confined to rotate about a known point (B). The position of point A_0 must be on a sphere about Point B whose radius is the length of the connecting rod. This means that A_0 is located at the intersection of the sphere centered at point B and the circle centered at point A. (See Figure 11)

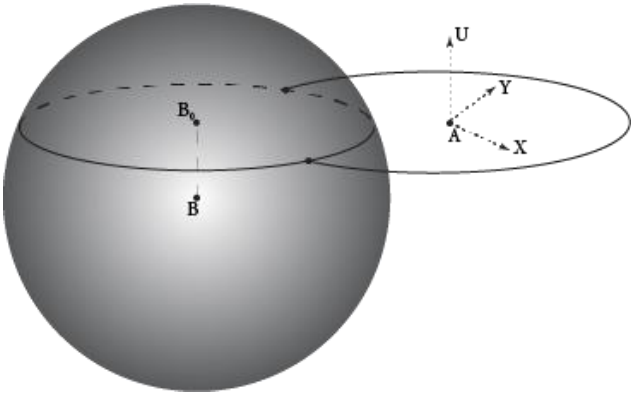


Figure 11: Defining the circle centered at B_0. This is the intersection of the plane containing circle A and sphere B.

A_0 is confined to a known circle, and therefore, a known plane. The intersection of this plane with Sphere B is a circle (see Figure 11). (Also, the intersection could be a point or it could not exist at all, in this case the link BA_0 would be too short.) This intersecting circle can be defined by the known geometric information. The center of the intersecting circle (B_0) makes a vector with the center of the sphere: $B\acute{B}_0$. This vector is parallel with the vector $\acute{U}$, because B_0 is just the projection of B onto the plane containing circle A. The vector $B\acute{B}_0$ can be defined as the projection of vector $\acute{B}A$ on to vector U:

$$B\acute{B}_0 = \acute{B}A \cdot \acute{U} \qquad (4)$$

The center (B_0) of the intersecting circle is now known and finding the radius will fully define the circle. This is accomplished by solving for the third length of a right triangle, see Figure 12.

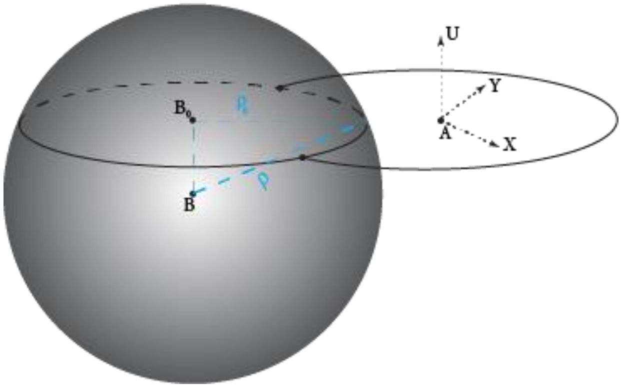


Figure 12. Finding the radius of the intersecting circle at B_0.

The equation for the radius follows:

$$\rho_0 = \sqrt{\rho^2 - \left|B\acute{B}_0\right|^2} \qquad (5)$$

where ρ_0 is the radius of the intersecting circle centered at B$_0$. The goal is to find θ_m. We can write θ_m as a function of γ and ϕ (see Figure 13). The angles γ and ϕ are used separately in serial and parallel kinematics. So the angles γ and ϕ of Figures 8 and 11 are different.

The next step towards solving for θ_m involves solving the triangle formed by the points B$_0$, A$_0$, and A. The three lengths of this triangle are known: $\left|B_0\acute{A}_0\right|$ is ρ_0, and $\left|A_0\acute{A}\right|$ is the length of the control arm (r), and the last length is the distance between two known points. The inner angle of this triangle γ is necessary to find the motor angle. This angle can be found using law of cosines. Next, the angle phi needs to be determined. This is the angle between the vector $\left|A\acute{B}_0\right|$ and the unit vector Y. This vector Y is an arbitrarily defined vector that represents the direction the control arm would be pointing when θ_m is zero. The angle ϕ can be found using the dot product of the unit vector and the vector $\left|A\acute{B}_0\right|$. Finally, the equation for θ_m is:

$$\theta_m = \phi \pm \gamma \qquad (6)$$

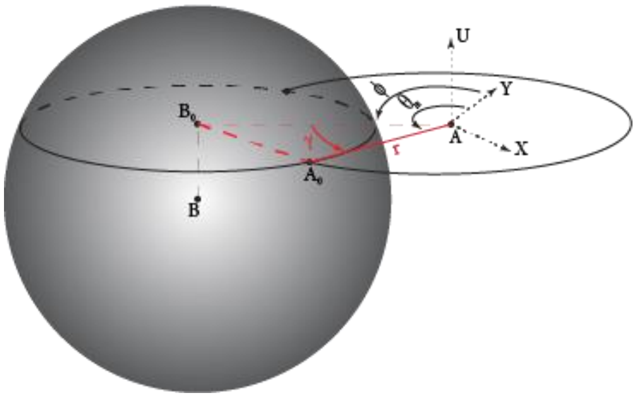


Figure 13: Final steps to solving the inverse kinematics. The kinematics are solved for θ_m.

There are two solutions to this problem, . “elbow in” and the other, “elbow out”. As can be seen in figures 3-5, some of the linkages are set in one configuration and others are set in the other. Figures 11-12 show an annotated sketch of both configurations.

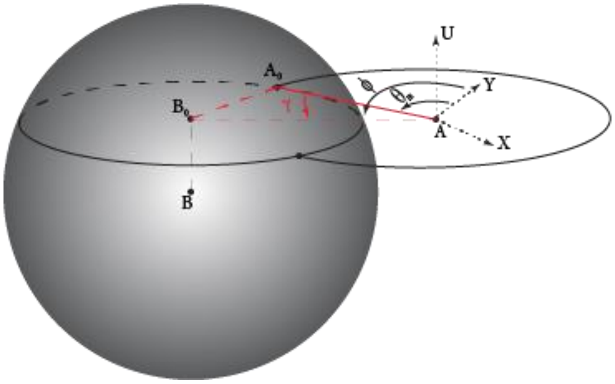


Figure 14: Annotated sketch of alternative solutions to the inverse kinematics problem.

Jacobian

Knowing the Jacobian of a robot limb is very important for determining the function of the motor. Each column of the Jacobian contains a vector that describes what a unit velocity input at the corresponding motor would output at the end effector. For example the first column is the vector that points in the direction that the leg will move if the first motor is moved. The Jacobian can also be used as a tool to look at torque/force relationships. Taking the transpose of the Jacobian gives a matrix that when multiplied with end effector forces gives motor torques. This kind of information is useful to know when designing a structure of this complexity.

For any limb structure calculating the Jacobian is as simple as taking the partial derivatives of the kinematics function. For this leg, there are multiple ways to find the Jacobian. The approach that we take is to individually differentiate the serial leg kinematics and the parallel kinematics. This gives two Jacobian matrices. We will call them J_S and J_p for the serial Jacobian and the parallel Jacobian, respectively. Defining three input and output vectors will used to define the two Jacobians:

$$\acute{X}_2 = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} \qquad (7)$$

The vector $\acute{X}_2$ is the x, y, z position of the robot foot relative to the hip joint in body coordinates.

$$\acute{\theta}_l = \begin{bmatrix} \theta_{Hip} \\ \alpha_{Hip} \\ \theta_{Knee} \end{bmatrix} \qquad (8)$$

The vector $\acute{\theta}_l$ is the vector of leg angles that are controlled by the actuators. The ankle joint is being ignored here.

$$\acute{\theta}_m = \begin{bmatrix} \theta_{m1} \\ \theta_{m2} \\ \theta_{m3} \end{bmatrix} \qquad (9)$$

The vector $\acute{\theta}_m$ contains the position of each motor on the leg. Now the Jacobian matrices can be defined.

$$\acute{\acute{X}}_2 = J_s \acute{\acute{\theta}}_l \qquad (10)$$

The matrix J_s is the Jacobian of the serial leg. We found this matrix by taking partial derivatives of the inverse kinematics and finding ${J_s}^{-1}$ analytically. The numerical inverse of this matrix can then be taken to determine J_s. When multiplied with the derivative of the leg angles this Jacobian gives a vector that represents the time derivative of the ankle position. Next the parallel kinematics can be used to find another Jacobian matrix.

$$\acute{\acute{\theta}}_l = J_p \acute{\acute{\theta}}_m \qquad (11)$$

The matrix J_p is the Jacobian of the parallel mechanism. It is found in the same manner as above. The partial derivatives of the inverse kinematics are used to find ${J_p}^{-1}$ and then the numerical inverse of this matrix produces J_p. This matrix establishes velocity relationships between motor speeds and joint speeds. It can be useful for visualizing the effect a given motor has on each joint. The whole leg Jacobian is the product of these two matrices. Substituting $\acute{\acute{\theta}}_l$ from (11) into (10) yields:

$$\acute{\acute{X}}_2 = J_s J_p \acute{\acute{\theta}}_m = J \acute{\acute{\theta}}_m \qquad (12)$$

where J is the classic Jacobian matrix. When multiplied with actuator speeds, it produces the end effector velocity vector. The Jacobian matrix was used heavily in the design process. It helped to create tools that gave numerical design goals, and simplified a large design space into a smaller one.

Design of physical Prototype

Selecting Geometric Properties. After selecting an actuation method, solving the kinematics equations, and defining the Jacobian matrices, the next step towards a functional leg prototype was selecting values for all the geometric properties of the leg. We used a gait pattern digitized from high speed footage of a cheetah running as a starting place. This presented an ambitious goal in terms of range of motion. For example, it required over 150 degrees of total hip flexion/extension.

Originally the authors chose to actuate the system with Configuration 1, but in the process of analysis it was discovered that this configuration was not ideal for the application. The 2-norm condition number of the Jacobian was plotted throughout the workspace of the robot arm, and in almost every position it was ill conditioned. Therefore, the robot was not controllable in 3D space by the motors. Even with an optimization tool, we could not find a solution with usable properties.

Switching to Configuration 2 solved the ill-conditioning problem. The optimization tool we developed was able to find a solution that had more than enough “usable workspace”. For this application usable workspace will be defined as any point in 3D space when the end effector is in that position its 2-norm condition number is less than 20. Figure 15 shows the improvement in usable workspace. The plot shows the surface that is the boundary between usable workspace and unusable workspace. The robot leg is shown in red for scale. It is obvious that selection of configuration can make a big difference.

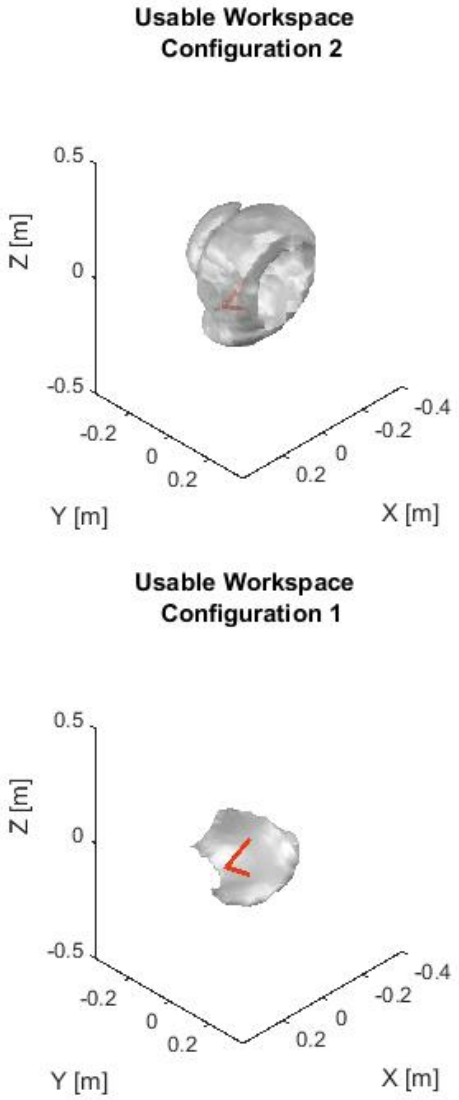


Figure 15. Improvement of volume of usable workspace. Upper figure shows workspace of the configuration 2 while the lower figure shows the workspace of configuration 1.

Building a Prototype. After selecting the leg's geometry, a physical model was designed in CAD software and printed via a 3D printer. The actuator we chose to use is a Dynamixel RX-24F actuator. This actuator has a small electric motor coupled with a 193:1 gear reduction. The on board electronics were replaced by a more versatile off-board multi-axis robot controller developed by SpringActive, Inc.

Testing on a Treadmill. To validate this approach, we set up a treadmill test. We fabricated a stand over the treadmill to which the leg prototype could be clamped. A gait profile was then created using an ankle joint path and the inverse kinematic solution. This path was commanded to the motors in a simple PID negative feedback loop. See Figure 16 for a series of photos of the leg as it moves through its gait pattern. The test validated the inverse kinematics and gave insight into the next stage of development of the leg. More rigorous testing will commence with the leg moving through different gait patterns for walking and running.

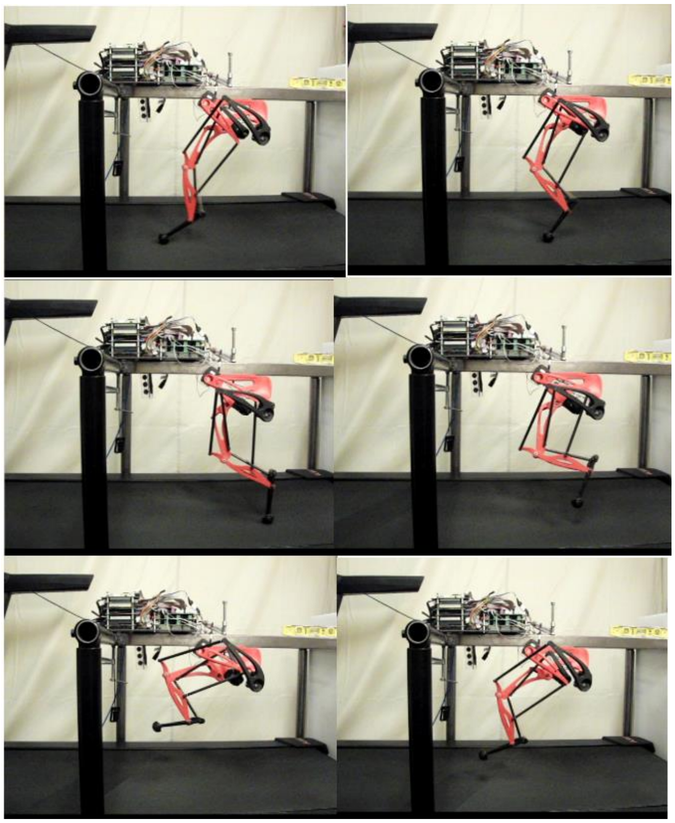


Figure 16: The leg moving through a gait pattern on a treadmill.

Conclusions

This paper outlines the design effort of a novel actuation approach from conceptualization to the first stages of testing. It details the choice of configuration, which turned out to be important, and somewhat counter intuitive. It also walks through the kinematic solutions, showing relatively simple solutions to challenging problems. The goal of the work is to use multiple small motors in parallel to actuate the hip and the knee. In this way, during the stance phase of gait, multiple motors can be used in parallel to provide a powerful burst for push-off. Our two-link parallel structure allows the motors to cross multiple joints and therefore can be used to actuate several joints at once. Also, by mounting motors at the base, the inertia of the leg is greatly reduced. A first treadmill test was performed to validate the work.

REFERENCES

[1] M. Raibert, K. Blankespoor, G. Nelson, R. Playter, and others, *"Bigdog, the rough-terrain quadruped robot,"* in Proceedings of the 17th World Congress, 2008, pp. 10823–10825.

[2] D. J. Hyun, S. Seok, J. Lee, and S. Kim, *"High speed trot-running: Implementation of a hierarchical controller using proprioceptive impedance control on the MIT Cheetah,"* Int. J. Robot. Res., vol. 33, no. 11, pp. 1417–1445, 2014.

[3] J. Pratt and B. Krupp, *"Design of a bipedal walking robot,"* 2008, vol. 6962, p. 69621F–69621F–13.

[4] T. Boaventura, C. Semini, J. Buchli, M. Frigerio, M. Focchi, and D. G. Caldwell, *"Dynamic torque control of a hydraulic quadruped robot,"* in 2012 IEEE International Conference on Robotics and Automation (ICRA), 2012, pp. 1889–1894.

[5] D. Koepl and J. Hurst, *"Impulse Control for Planar Spring-Mass Running,"* J. Intell. Robot. Syst., vol. 74, no. 3–4, pp. 589–603, Jun. 2014.

[6] J. Bhounsule, J. Cortell, and A. Ruina, *"Design and control of Ranger: an energy-efficient, dynamic walking robot,"* in Proc. CLAWAR, 2012, pp. 441–448.

[7] J. A. Grimes and J. W. Hurst, *"The design of ATRIAS 1.0 a unique monoped, hopping robot,"* in Proceedings of the 2012 International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, 2012, pp. 548–554.

[8] *"ASME DC | Journal of Dynamic Systems, Measurement, and Control | Performance Analysis and Feedback Control of ATRIAS, A Three-Dimensional Bipedal Robot."* [Online]. Available: http://dynamicsystems.asmedigitalcollection.asme.org.ezproxy1.lib.asu.edu/article.aspx?articleid=1760199. [Accessed: 02-Sep-2015].

[9] *"Maxon DC motor and Maxon EC motor - Key information"*

http://www.maxonmotor.com/medias/sys_master/root/8815460712478/DC-EC-Key-Information-14-EN-42-50.pdf?attachment=true

[10] *"Motor data and operating ranges of maxon DC motors"*, maxon academy

http://www.maxonmotorusa.com/medias/sys_master/8798985682974/maxonMotorData-Handouts.pdf?attachment=true