

|  |  |
| --- | --- |
| Milwaukee School of Engineering | Mechanical Engineering Senior Design |

***Advisor***: Dr. Luis A. Rodriguez

***Team Members***:

Logan Beaver

Justin Campbell

Tyler Paddock

Ronald Shipman

**February 19, 2015**

Development of an Agile Educational Robot

Team A.R.C. - Executive Summary

# Executive Summary

The Milwaukee School of Engineering (MSOE) participates in Science Technology Engineering and Mathematics (STEM) outreach events for prospective students. The school will benefit greatly from having a sophisticated robotic control system to build excitement about STEM as well as sparking interest in fluid power, automation, and the controls fields. An agile pneumatic robot is not only a complicated control system that can be used to get young people excited about STEM, but it will also increase the prestige of MSOE knowing that a group of seniors attending the school were able to design and build the system from the ground up. In addition it also provides an exciting opportunity for future groups to iterate on the design and integrate new and exciting features.

To fulfill the needs of the project existing robot designs were researched to help determine the initial objectives and constraints for the project. Existing walking robots such as Boston Dynamics Big Dog and Little Dog, the Swiss Federal Institute of Technology (EPFL) Cheetah Cub, and various robots from the Massachusetts Institute of Technology Computer Science and Artificial Intelligence (CSAIL) laboratory were examined. These robots were used as a baseline comparison for the design specifications and constraints. From the robots the following constraints and project goals were identified for this project’s design.

* A maximum weight of 35 Kg for portability
* Robot fitting within a 0.75 m x 0.75 m x 1.0 m box for portability
* Custom debug panel creation to facilitate controls troubleshooting
* MATLAB and Simulink model support to allow mechanical engineering students to update control algorithms without knowledge of C/C++
* Electronic fuses and shielding to protect the robot and operator during use and maintenance
* Mechanical protection to reduce the risk of pinching and self-collision damage to the robot
* An easy to access emergency stop to quickly depower the robot
* A pressure relief valve to reduce the risk of overloading and damaging pneumatic components

The work done on this project is a continuation of the work done by Kevin Lee during the Research Experience for Undergraduates (REU) at MSOE. His work involved deriving a dynamic model for a simplified quadruped robot. This work is continued by the agile robotics controls team in deriving a full dynamic model for the physical robot and integrating it with control algorithms to manipulate the robot. This resulting robot design will be implemented in actual hardware toward the end of the project.

Pneumatic power was chosen over electronic and hydraulic power for a variety of reasons. Pneumatics were chosen over hydraulics due to the weight and maintenance needs associated with hydraulic systems. Hydraulic systems are also dirtier than pneumatic systems and pneumatic working fluid is freely available. Pneumatics were chosen over electronic systems due to their higher power density. Electrical systems have lower power density from the inefficiencies in converting electrical energy to work. In addition fluid power systems are compliant, meaning that if a large force is applied to the pneumatic actuators the fluid can compress whereas electronic actuators will experience an increased stress.

The robot locomotion utilizes a quadruped design. Four legs were selected because of the inherent static stability of a four legged design coupled with the decreased control complexity compared to robots with additional legs over four. This will allow the robot to initially actuate a slow statically stable gait as the software architecture is developed and will eventually lead to more sophisticated gaits being developed without the need for additional hardware.

The controls implementation will initially be done using a software simulation. This allows rapid updating of the main codebase as the mechanical and electronic designs iterate. Eventually the controls will be implemented into a main microcontroller which will take user input through a human machine interface (HMI) and relay the commands to the pneumatic actuators.

Four design alternatives were drafted to fulfill the design requirements. The design alternatives were named *Arachnia*, *Hexabox*, *Boxxy*, and *DogeBot*. After scoring each robot with a design matrix, *DogeBot* was chosen as the design to be continued, with a score of **96.19** out of 100.

## Dynamics

To determine the internal forces felt in the joints and the required torques for locomotion a dynamic mathematical model of the robot was constructed. From the specifications the robot was known to have four legs with two links each all attached to a main chassis. Summing the force and torque around each link of the robot resulted in 27 simultaneous equations used to calculate the state of the robot. To simplify the calculations it was assumed that the foot forces were known and that the robot exhibited purely planar motion. A diagram of the mathematical notation and free body diagrams are shown in Figures 4 to 6 in the appendix.

After summing the forces and torques for the robot joints the equations were split into two systems of equations. Eight equations were put into matrix form to solve for each of the hip and knee torques required at each joint, and sixteen equations were put into matrix form to solve for the reactionary forces in the hip and knee joints in the x and y direction. The sum of the force and torque equations about the body were not used in these calculations because the body’s state can be calculated entirely based on the hip reaction forces and torques. The 24 developed equations were all in terms of the kinematic and inertial state of the robot.

An initial Solidworks model of the robot was developed to determine the inertial state of the robot. The geometry of the robot was based off of the final design chosen after the fall phase of the project, and iterative changes were made based off of dynamic simulations, mechanical failure analyses, and motion studies. A diagram of the robot is shown in Figure 3 in the appendix.

With a known geometry for the legs a walking simulation was run to determine the kinematic state of the robot. The simulation was used to find the required angular accelerations of each joint to satisfy the minimum speed specification of 0.5 m/s. A half ellipse was selected as the foot path to test due to the ease of changing the step length and height as well as its similarity to the reversed Pear-Shaped Quadratic, which closely resembles paths found in animal steps. An image of the step simulation is shown in Figure 7 in the appendix.

After calculating the inertial and kinematic states of the robot for an elliptical step gait was selected for the dynamic simulation. The three gaits to be implemented on the robot are the drag, creep, and walk gaits. The drag gait is the simplest and involves one leg moving and dragging the rest of the robot along the ground. This gait is the simplest to implement and the most stable due to only a single leg actuating. Dragging is the least efficient gait.

The creep gait involves leaning the robot forward then actuating one leg at a time to center the body over the legs. The creep gait is more efficient than the drag because extra energy is not required to overcome the friction of dragging the robot. The creep gait is still relatively stable and simple compared to the drag, but there is extra complexity from leaning the body forward and actuating all four legs.

The walk gait involves actuating all four legs at once. Two diagonal legs move forward while the opposite push backward, moving the robot forward. Walking is the most complicated gait that will be attempted due to actuating all four legs simultaneously. It is also the fastest gait because the robot is constantly moving forward. A diagram of actuation for each gait is given in Figure 8 in the appendix.

The inertial and kinematic data was then used in the dynamic simulation to calculate the maximum torques required on the joints as well as the internal forces felt by the joints. The initial simulated gait used was the drag gait due to the simplicity of the gait to simulate and the extra forces felt from dragging the robot chassis. It was assumed that the three static legs were completely vertical and the feet felt a third of the total robot weight as force in the Y direction, and no force in the X. These simplifying assumptions are mostly accurate, and the forces felt by the stationary legs were so incredibly small in comparison to the swinging leg that the error was negligible.

The torques and forces were calculated for every instant of a 0.5 m drag step lasting one second. The states at the maximum torque and maximum total force were saved and used to determine cylinder specifications and mechanical stress on the leg. Because the simulation assumed pure torque applied to each joint intermittent calculations were required based on the position of the legs and piston attachment points to calculate the equivalent moment arm, and therefore force exerted by the piston on the leg. The maximum forces and torques were found to be during the swing phase of the step, which is intuitive because the legs are undergoing rapid acceleration requiring large inertial forces to compensate.

## Mechanical Design

The major mechanical components of the robot are the legs and the chassis. The legs are further broken down into the thigh and the shank. The constraints placed on the mechanical design include the overall dimensions, weight, and carrying capacity of the robot.

The chassis of the robot is designed to support the legs while in motion, as well as serve as the housing for the pneumatic and electrical components. The choice to use 6105 –T5 T-Slotted Aluminum framing for the chassis was due to the simplicity of construction. With a number of connection plates and brackets available, a simple chassis would be easy to construct. In addition to its simplicity, the Aluminum framing is lightweight, fulfilling the robot’s weight constraint, and strong enough to support the forces exerted by the pneumatics during operation. Custom hip joints will need to be designed to be able to easily attach the legs to the outside of the frame. Additionally, mounts for the pistons will need to be created in a similar fashion.

The design of the legs is based on the anatomy of quadruped mammals, more specifically dogs. In order to reduce the robots complexity, the number of joints in the legs was reduced from three to two. In order to compensate for the loss in range of motion, the thigh was designed with a bend in it. The bend allows for a shorter stroke length for the pneumatic cylinder. This in turn, means that the cylinder can be attached closer to the hip joint on both the body and the thigh. The bend in the thigh is also beneficial in that it prevents the piston controlling the rotation of the knee joint from reaching a singularity point and possibly seizing. The design of the shank is much simpler. One factor that helped to determine the lengths of the thigh and shank was the desired step length. Similar to the body, the legs will most likely be constructed using an Aluminum alloy. A specific material is yet to be determined and will be based on the required strength and machinability of the design.

As a means to ensure the design would be handle the forces due to the weight and pistons, an initial FEA analysis was run. Each component was tested separately, using the worst case scenario as an upper limit test of the structures: Trying to move at maximum velocity using the slowest gait with the joints seized. Under these conditions, the components hold up, meaning the designs should be more than capable of handling normal operating conditions.

An important design consideration focused on in this phase of the project was the design of the robot’s feet. The foot will provide the necessary friction required to prevent the robot from slipping on its walking surface. Options for feet of the robot include rubber, either spheres or rectangular sleeves, in which the lower shank is inserted and liquid rubber that can be applied to the bottom of the shank and allowed to cure. The rubber sphere feet allow the foot to contact the ground at any orientation, while still providing the necessary friction, however if the shank cannot be securely inserted, the effectiveness of the foot during the leg’s motion becomes a major concern. Rubber sleeves face the same considerations as rubber spheres in that the shank must be inserted securely, so for both options methods for ensuring a sound connection must be looked into. If appropriate sizes of the rubber spheres or sleeves are not available, custom making the appropriate feet is then required. When considering the need of custom manufacturing the feet of the robot, ultimately the use of a liquid rubber substance, such as urethane, that can be cured into the necessary shape, as an option for the feet of the robot becomes the most attractive. By cutting a hole into the bottom of the shank and dipping the shank into the liquid rubber, the appropriate shape of the leg is formed and, as liquid rubber flows through the hole in the shank and cures, a connection point between the foot and the shank is formed. By repeating this dipping process, an appropriate layer of rubber can be applied to the lower portion of the shank, generating an effective foot for the robot. Initial testing of the chosen liquid rubber substance will be conducted to confirm desired mechanical properties.

## Pneumatic Component Specifications

After the dynamic simulation of the robot’s legs was conducted and the necessary forces required to generate the torques in the legs were determined, initial air cylinder specifications could be made. These initial specifications were iterated based on appropriate bore sizes, related to leg dimensions, pressures required in cylinders to produce necessary forces, utilizing Equation 1 and the necessary volumetric flow rates of the cylinders, utilizing Equation 2, to produce the required forces. These values can then be used to specify compressor pressure and storage required. If the pressures necessary in the air cylinders was too high based on the achievable output pressure of the air compressor, a larger bore diameter was chosen to lower this necessary pressure. If the required volumetric flow rate of the cylinders was too large, options to lower this includes reducing the stroke length, decreasing the bore diameter of the cylinder, or reducing the required pressure in the air cylinder. After specifying air cylinders and the air compressor, the components that need to be specified include the relieve valve and the directional control valves that allows the necessary volumetric flow to the air cylinders.

(1)

where F is the required force to generate the required torque at one of the joints of the robot, Acap is the area of the cap end of the piston and Arod is the area of the rod end of the piston.

(2)

where Q is the required volumetric flow rate, Ap is the surface area differential of the cap end and rod end of the piston, DT is the total displacement of the piston in 1 cycle, N is the number of cycles per minute and C is the compression ratio comparing the working pressure in the air cylinder to atmospheric pressure. A Sample calculation of cylinder pressure and necessary volumetric flow rate can be found at the end of the appendix.

The electronics of the robot are broken up into two major subsystems, the motherboard and the debug panel. The motherboard was designed to contain the auxiliary electronics and signal conditioning components needed for the robot. The debug panel contains all necessary electronics to display battery levels and other statuses of the robot.

The motherboard contains the signal conditioning for each of the pneumatics cylinders. Each pneumatic cylinder is controlled by a signal analog direct current voltage. However, the signal driving this analog voltage is a pulse-width-modulated (PWM) output on the microcontroller. To convert a PWM into an analog signal an active low pass filter is used. After the low pass filter an opto-isolator is used to separate the microcontroller circuit from the pneumatic actuator circuit. An opto-isolator works by converting an electrical signal into an optical signal by using a diode. The optical signal is recaptures within the device and output onto another circuit as a current signal. At the output of the opto-isolator a trans-impedance amplifier is used to convert the output current signal to a voltage signal for the solenoid. To handle the feedback signal from the pneumatic actuator another opto-isolator is used to separate the two power circuits then the signal is amplified before being read by the microcontroller’s built in analog to digital converters (ADCs).

The debug panel subsystem contains a physical panel with light emitting diodes (LED) and connections for banana plug cables. The LEDs are used to show battery levels and the status of the robot. The banana plug connectors are used to interface to Milwaukee School of Engineering’s test equipment in the labs. Banana plugs are used because they are standard on test equipment. A USB slot is also included on the debug panel to assist in programming the microcontroller while leaving it in the robot.

The control algorithms are software implementations on the microcontroller. The software was not written by the team, but is instead code generated from Mathwork’s Simulink models. The control algorithm a proportional integral derivative controller using one input signal. This input signal is the position setpoint minus the feedback signal from the cylinder. Figure 1 shows the Simulink model for a whole leg using an upper and lower cylinder PID.

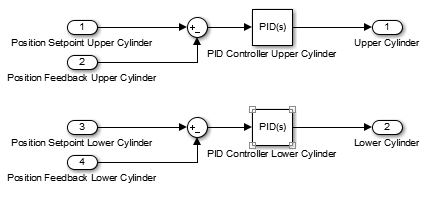


Figure 1: PID Model of a Leg

## State Machine

It was determined that the behavior of the robot would be dictated utilizing a state machine. Using a state machine, based on the physical condition of the robot and the user input, the robot moves from different states and behaves occurring to how those states are defined within the state machine. Currently an initial state machine has been developed that has two states. The first state is a stand-by state in which the robot is determined to be functioning correctly and is awaiting user input. The second state is a stop state in which the robot, after some input which forces the robot into this state, completely stops ongoing motion in the robot and moves the robot into a stable position. Figure X, found in Appendix X, shows a state machine flowchart in which various states are represented. States include forward and backward motion states in the robot, turning states and a stop state.



Figure 2: Flowchart representation of state machine architecture

# Appendix

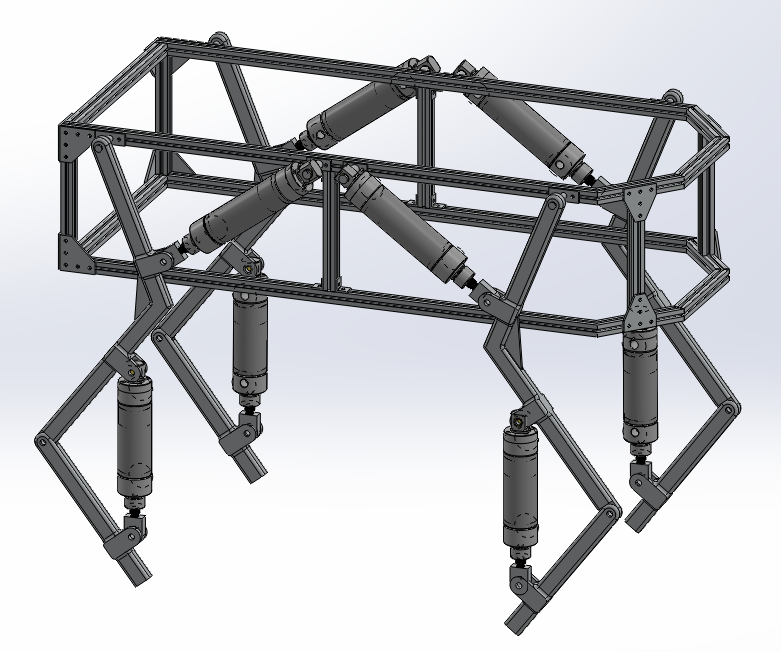


Figure 3: Mechanical design of DogeBot

ΘShank

Θthigh

Y

X

Wbody

F3

F4

F1

F2

4

3

2

1



Figure 4: Overall robot free body diagram with leg angle definitions

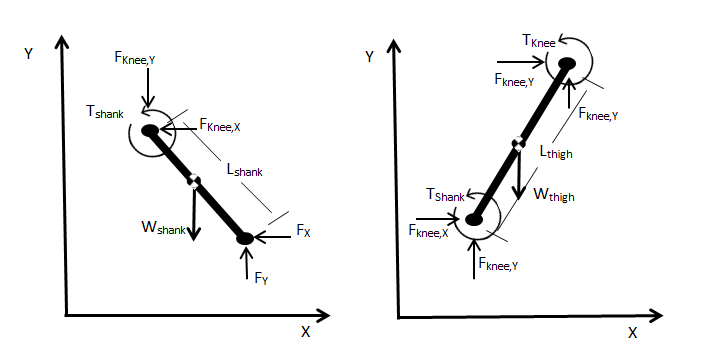


Figure 5: Leg free body diagrams

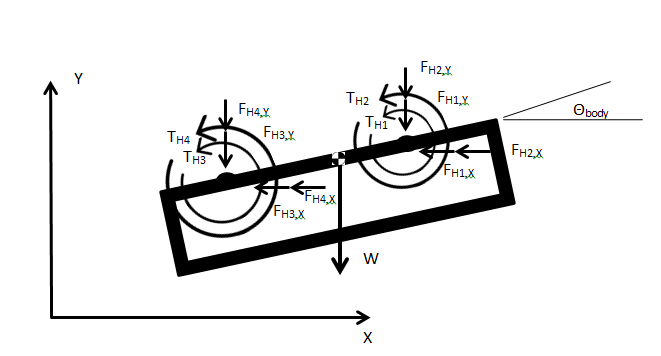


Figure 6: Body free body diagram with body angle definition



Figure 7: Kinematic model animation of a single leg following a semi ellipse path

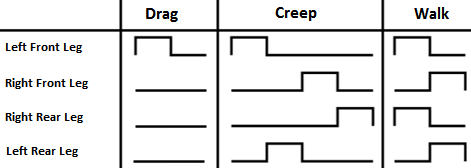


Figure 8: Leg actuation pattern of different walking gaits

Suppose a necessary force of 178 lbf is required to produce necessary torques in the hip joints of the robot. Given a bore diameter of 2 inches, the necessary pressure in the cylinder can be determined as follows

Assuming this pressure is the average within a thigh cylinder for a minute of operation, 60 total cycles for all thigh cylinders and a total piston travel of 6 inches per cycle, the total volumetric flow required by all thigh cylinders is calculated as follows

2.98 CFM (cubic feet per minute)