Control of a Pneumatically Actuated, Fully Inflatable, Fabric-based, Humanoid Robot

Charles M. Best   Joshua P. Wilson   Marc D. Killpack

Abstract—Although humanoid robots take the form of humans, these robots often approach manipulating the world in a very different way than humans. For example, many humanoid robots require precise position control and geometric models to interact successfully with the world. Humanoid robots also often avoid making contact with the world unless the contact can be well modeled. In this work, we present preliminary results on soft robot platforms that can change the way humanoid robots interact with humans and human environments. We present preliminary control methods and testing on fully inflatable, pneumatically actuated, soft robots. We first show that model predictive control (MPC) and linear quadratic regulation (LQR) are sufficient for position control of a single joint with one degree of freedom. We also demonstrate MPC and LQR as methods of control for an inflatable humanoid robot on one arm using five degrees of freedom. Our initial development for multi-joint control is based on the methods developed for the single degree of freedom platform. Using the MPC controller with joint space commands, a task of picking up a board from a chair and placing it in a box was successful eight out of ten times. Our models and control methods will allow for a new type of humanoid robots that are well suited to interacting more safely and naturally in human environments.

I. INTRODUCTION

There has been the promise for many years of robots revolutionizing the way that humans accomplish work and interact with the world. Humanoid robots especially have been the recent subject of large-scale endeavors such as the DARPA robotics challenge. Although the results from past research have been impressive, in this paper, we present a fundamentally different robot platform and approach to controlling this platform than past humanoid robotics research. In particular, much of the current humanoid robot research uses position controlled robot arms for manipulation which requires accurate geometric and sometimes accurate models of contact dynamics to be successful. Using a soft, humanoid robot platform that is inherently more robust to contact with the world, humans, and even other robots is an approach that may be fruitful for advances in manipulation in unstructured environments.

The research in this paper demonstrates the first published work that we are aware of on high level control for a pneumatically actuated, fully inflatable, soft humanoid robot (shown in Figure 1). Pneumatic actuation means that joint torques are produced through the use of compressed air. Fully inflatable means that the system structure comes from pressurized chambers within a non-rigid medium such as fabrics or polymers. A robot of this type is naturally lightweight and compliant which can be desirable in many situations but does result in greater complexity and difficulty in sensing, actuation, and control [1].

II. RELATED WORKS

Our specific contributions include the following:

- Development of a state space impedance model that predicts future states of an antagonistic, pneumatically actuated joint.
- Development of a joint space Model Predictive Controller for soft humanoid robots which is shown to have comparable performance to the more conventionally used Linear Quadratic Controller while allowing realistic system constraints to be incorporated.
- Demonstration of control for a real inflatable, fabric based robots with pneumatic actuation that can be controlled to repeatedly accomplish useful tasks.

This paper is organized as follows. Section II discusses past work relevant to this research. A description of the systems for which we have implemented control is in Section III. In Section IV we then discuss the model and formulation for MPC and LQR controllers for a single degree of freedom joint along with the system response to the controllers. The use of a MPC and LQR controller on a multiple degree of freedom arm is then discussed in Section V.

Fig. 1: Inflatable Humanoid Robot named King Louie
more effective at interacting with humans or human environments. Robots currently have limited uses in homes, hospitals, schools, or other areas where safe interaction with people or the environment may be necessary. One reason why robots are not common in these places is because robots can be dangerous to people or property when there is incidental contact.

Lightweight robots such as our platform have less inertia and are less likely to cause harm because of lower contact forces and lower overall momentum when moving at varying speeds. Our research is motivated by a desire to take advantage of the positive characteristics of soft and inflatable robotic systems while maintaining a level of control that will allow them to be useful. Applications for a robot of this type include search and rescue, health care, living assistance, and space exploration.

Research has shown that contact forces from inflatable links can be controlled in [2] and [3] with cable driven actuators. While cable driven actuators are an effective means of actuation for inflatable structures, our work will focus on pneumatic actuation as this is more consistent with the design intent for inflatable structures. In [4], [5], and [6] it was found that planning is possible for fluid driven elastomer actuators using dynamic and constant curvature kinematic models which is a similar actuation method to ours. The fabric designs for our actuators were based from the designs for rotary, fabric-based, pneumatic actuated joints which were proposed in [7]. For rigid, traditional servo-pneumatic actuators, work in [8] characterized different models which made different constant temperature assumptions and in [9] these assumptions were used to control force and stiffness for a linear pneumatic actuator. Past research involving inflatable robots in general looks at the performance of an actuator or a series of actuators. We show that an entire system can be inflatable and controls can be developed for the system to effectively complete tasks normally done by a robot with rigid structure. The lack of literature on the control of inflatable structures where there is a wide range of applications suggests a viable new area for research.

Similar to the actuators on our robot, pneumatic artificial muscles called McKibben artificial muscles have been developed that use compressed air, and are used in [10] and [11] as orthotic devices. McKibben artificial muscles have also been used in robots, and control methods have been developed for these robots in [12] and [13]. Our robot differs from these in that McKibben muscles exert a tension force when pressurized whereas the actuators in our robot exhibit a torque on the joint from compressive forces. Probably more important is that prior researchers use actuators in parallel with a rigid-link system which is significantly different than using a fully inflatable system.

In terms of methods used to control our robot, Model predictive control (MPC) is an optimal control method that has long been used in the chemical processing industry [14]. Recent advances in computing power and dynamic optimization techniques such as those presented in [15], [16] have made MPC a viable control method in applications that require a high control rate. MPC has been demonstrated and shown promise in many areas outside of the chemical processing industry including control of Unmanned Aerial & Surface Vehicles [17], [18], [19] and even more recently in robotics [20], [21], [22], [23], [24].

Because MPC is not commonly used in robotics, the more conventional and widely accepted linear quadratic control is applied to the system to show that MPC is a viable control method. MPC and LQR control methods are very similar and overlap in some cases as seen in [25] where constraints were added to the LQR controller with a finite horizon making it MPC. It was seen in [26] that LQR methods are viable options for full body humanoid robots. LQR methods have also been used for humanoid tasks such as balance [27] and walking [28].

III. ROBOT PLATFORM DESCRIPTION

The platform used for this research is a fourteen degree of freedom humanoid robot called King Louie (Figure 1) and a single degree of freedom joint called a grub (Figure 2). Both were developed and built by Pneubotics, an affiliate of Otherlab. Besides internal electronics such as IMU and pressure sensors, these platforms are entirely made of ballistic nylon fabric with internal bladders to prevent air leakage. The structure of these robots comes from an inflatable bladder which is pressurized to between 1-2 PSI gage. At each joint, there are antagonistic actuators which can be filled to pressures between 0-25 PSI gage. For this research we use pressures between 0-17 PSI gage because of pinching effects in the main chamber at higher pressures. These actuators are similar to the designs mentioned in [7].

Fig. 2: This is a single degree of freedom platform that we call a “grub.”

Initial work for model development and control of our inflatable robotic systems was done with the grub. After we performed initial analysis and testing on the 1-DoF system (grub) we applied the same methods to the more complex 5-DoF arm on our humanoid system (King Louie). King Louie’s arm configuration, orientation, and motion can be approximated with Denavit-Hartenberg parameters using an assumption of rigid links and compliant joints. These parameters are shown in Table I with values for $\alpha$ and $d$ in meters, and $\theta$ and $\alpha$ in radians.


Joint angles are measured by using a motion capture system and calculating the resultant quaternion between link orientations for King Louie. The joint angle for the grub is measured with an inertial measurement unit IMU located on the link. King Louie has IMU sensors on each link and we are purposefully using motion capture data in a way such that it can be directly replaced by noisier IMU data in future work.

IV. SINGLE JOINT DYNAMICS AND CONTROL

The steady state joint angle response for a step input of pressures in both chambers on a single joint shows hysteresis and depends on the initial pressure and angle states. A sample mapping can be seen in Figure 3. This mapping was produced using the same initial conditions for each data point and varying the commanded pressure in each chamber. The shape of the map changes when there are different initial conditions but the trend stays the same. In addition, there are multiple inputs that map to a single position output which means there are an infinite number of pressure combinations to achieve a desired angle. Another complication is that each pressure combination can result in a steady state response that varies from the average response by as much as 80° due to hysteresis. In future work, we expect that MPC could take advantage of having two pressure inputs on each joint and modeling the hysteresis to have improved performance.

The equation

\[ \theta_d = \frac{s_2}{1 + e^{-(s_1(P - s_4))}} + s_3 \]  

is the general form of the equation we used to map pressure inputs to input angles for an impedance control model of the joint used with LQR and MPC. The coefficient \( s_1 \) determines the slope of the curve and was fit to match the slope in Figure 4. The coefficients \( s_2, s_3, \) and \( s_4 \) are determined by the joint limits. For this work \( q \) designates joint angles and \( \theta_d \) will represent system inputs. The system input \( \theta_d \) is a function of pressures that maps the pressure states to an angle which is necessary for the units in the simplified impedance model, discussed later, to be consistent.

We have developed this pressure model from data taken on the grub and have then used it directly on each individual joint of King Louie. Our results later in the paper demonstrate that this model can successfully be used as a simplified general model for pressure-angle mapping in this type of robotic system. Although this method simplifies the control input problem, the constraint on commanded pressure in the two joint chambers also limits system capabilities. The ability to use the full pressure range for each bladder could result in faster system response, better dynamic behaviour, and the ability to have variable stiffness at each joint.
A. Model

For this research we have made several rigid body assumptions to simplify the model for the soft robot system such as there being no lateral or torsional deflection along the links. Deviations from this approximate model will degrade overall system performance, but we currently treat them as disturbances on the system. Future work will include aspects within the model that capture dynamics unique to an inflatable robot. However, the results of LQR and MPC for the system presented here show that a rigid body model is sufficient for basic control of the system.

The differential equation that we use to describe the motion of a single link is

$$ I \ddot{q} + K_d \dot{q} + mg \frac{L}{2} \sin(q) = \tau_a $$

where $I$ is the moment of inertia of the link about its joint, $K_d$ is a damping coefficient, $m$ is the mass of the link, $g$ is the gravity constant, and $L$ is the length of the link. The applied torque for our impedance control model is

$$ \tau_a = K_s (\theta_d - q) $$

where $K_s$ is the stiffness coefficient and $\theta_d$ is the system input.

This impedance model for torque behaves like a torque spring where the deflection is the difference between $\theta_d$ and the joint angles $q$, $\theta_d$ is assumed to be a steady state response for a pressure input. Our actual low-level joint pressure control uses two pressure commands instead of a commanded angle $\theta_d$, so Equation (4) is needed to map the system input directly to a pressure combination using Equations (1) and (2).

Solving for $\ddot{q}$ and linearizing the system with a change of variable where $q_e$ and $\theta_e$ define the point about which we are linearizing such that

$$ \ddot{q} = q - q_e $$

and

$$ \ddot{\theta}_d = \theta_d - \theta_e $$

we can then represent the dynamics in state space form as follows:

$$ \begin{bmatrix} \ddot{q} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} A \\ B \end{bmatrix} \begin{bmatrix} \ddot{\theta}_d \\ \dot{\theta}_d \end{bmatrix} $$

$$ \dot{q} = C \begin{bmatrix} \ddot{q} \\ \dot{q} \end{bmatrix} $$

where $\dot{q}$ is the measurement of the joint angle and

$$ A = \begin{bmatrix} -\frac{K_d}{I} & \frac{mgL}{2I} \cos(q_e) - \frac{K_s}{I} \\ 1 & 0 \end{bmatrix} $$

$$ B = \begin{bmatrix} \frac{K_s}{I} \\ 0 \end{bmatrix} $$

and

$$ C = \begin{bmatrix} 0 \\ 1 \end{bmatrix} $$

The damping and stiffness terms were estimated directly by applying a step response to the real system and optimizing in terms of $\frac{K_d}{I}$, $\frac{K_s}{I}$, $m$. From this system identification, we saw that forces due to gravity could be modeled as having little effect on the system output due to the low overall weight of the platform such that:

$$ A \approx \begin{bmatrix} -\frac{K_d}{I} & -\frac{K_s}{I} \\ 1 & 0 \end{bmatrix} $$

was an adequate model.

This was found when $m$ was identified as being near zero and three orders of magnitude smaller than what our system identification found for $K_s$ and $K_d$. Adding a more significant link-side load to a single joint (or a multi-joint robot), will require re-examining this approximation in future work.

With $T_s$ as the time step, we transform the state space form from the continuous time-domain to the discrete domain using the matrix exponential method found in [29] where

$$ A_d = e^{A T_s} $$

and

$$ B_d = (A_d - I) A_d^{-1} B $$

so that the complete discrete state space equation is

$$ \begin{bmatrix} \ddot{q}(k + 1) \\ \dot{q}(k + 1) \end{bmatrix} = A_d \begin{bmatrix} \ddot{q}(k) \\ \dot{q}(k) \end{bmatrix} + B_d \dot{\theta}_d. $$

B. Sensing

A single IMU is placed on the link of the grub to estimate the joint angle. With the grub oriented upwards, the joint angle can be estimated from the measurements of two perpendicular accelerometers measuring the direction of gravity. We use a Kalman filter during actuation to produce a smoothed state estimate.

C. Control

For the single degree of freedom control development we used both an LQR and MPC controller for comparison. Both controllers use the same dynamic model as described previously in Section IV-A. The form of these two controllers is described in the next sections. A block diagram for the system can be seen in Figure 5. The controller sends inputs in the form of $\theta_d$ which are converted into pressure commands by solving for $P$ in Equation (4) and then $P_0$ and $P_1$ in Equations (1) and (2). A PID pressure controller then adjusts actuator pressures by manipulating electrical currents sent to spool valves which adjusts air flow. Pressures commanded to the actuators range from 0 to 17 PSI gage with a source pressure around 20 PSI gage. The PID gains were tuned for acceptable performance for the grub and for each joint on King Louie and can maintain the pressure within 0.1 PSI from commanded values for $P_0$ and $P_1$. Future hardware developments will increase the allowable max actuator and
source pressure which will result in better response time and payload capability for the system.

![Control system block diagram for complete system.](image)

Fig. 5: Control system block diagram for complete system.

1) LQR Control: For position control using a linear quadratic regulator (LQR), the discrete state space model from Equation (17) was used. A new discrete state $v(k)$, defined as

$$v(k + 1) = v(k) + q_{goal}(k + 1) - q(k + 1)$$

(18)

which is the discrete approximation of the integral of the error [29] and $q_{goal}$ is the commanded joint angle where the controller is driving the system. The new discrete state space model then becomes

$$
\begin{bmatrix}
\dot{q}(k + 1) \\
\dot{v}(k + 1)
\end{bmatrix} =
\begin{bmatrix}
A_i & 0 \\
-C_dA_i & 1
\end{bmatrix}
\begin{bmatrix}
\dot{q}(k) \\
\dot{v}(k)
\end{bmatrix} +
\begin{bmatrix}
B_i \\
-C_dB_i
\end{bmatrix} \Delta \theta_d
$$

(19)

where

$$A_i = \begin{bmatrix} A_d & 0 \\ -C_dA_d & 1 \end{bmatrix}$$

(20)

and

$$B_i = \begin{bmatrix} B_d \\ -C_dB_d \end{bmatrix}.$$  

(21)

In preliminary testing, we saw that without a limit on the input $\theta_d$, the controller would command large changes in inputs that would instantly saturate the pressure controller in both directions because the pressure dynamics are not accounted for in the model. In order to limit the system input, $\theta_d$ was made a state as defined by

$$\hat{\theta}_d(k + 1) = \hat{\theta}_d(k) + \Delta \hat{\theta}_d$$

(22)

where $\Delta \hat{\theta}_d$ was the new system input. The discrete state space model for the LQR controller then becomes

$$X(k + 1) = A_{lqr}X(k) + B_{lqr} \Delta \hat{\theta}_d$$

(23)

where

$$X = \begin{bmatrix} \hat{q} & \hat{q} & v & \hat{\theta}_d \end{bmatrix}^T$$

(24)

$$A_{lqr} = \begin{bmatrix} A_i & B_i \\ 0 & 1 \end{bmatrix}$$

(25)

and

$$B_{lqr} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T.$$  

(26)

The control input $\Delta \hat{\theta}_d$ was found at each step using the LQR formulation

$$\Delta \hat{\theta}_d = -KX$$

(27)

where $K$ is the solution to the algebraic Riccati equation using $A_{lqr}$, $B_{lqr}$, $Q$, and $R$ where $Q$ contains the weights on the states and $R$ is the weight on the input. Both $Q$ and $R$ were tuned manually so the joint could achieve the commanded angles.

2) Model Predictive Control: The model predictive controller uses the discrete model Equation (17) to predict future states across the horizon $T$. The system is simulated over the horizon where the inputs $\hat{\theta}_d$ are picked in such a way that the cost function is minimized. The cost function minimized across the horizon is

$$f(\hat{q}, \hat{\dot{q}}, \hat{\theta}_d) = \sum_{k=0}^{T} (\|\hat{\dot{q}}(k) - \hat{\dot{q}}_{goal}^2 + \|\hat{\dot{\theta}}(k)\|_R^2 + \|\hat{\dot{\theta}}_d(k)\|_R^2 + \|\hat{\dot{q}}(T) - \hat{\dot{q}}_{goal}^2 + \|\hat{\dot{\theta}}_d(k)\|_R^2 + \|\hat{\dot{q}}(T)\|_R^2)$$

(28)

subject to the system model as a constraint in Equation (17) and the other following constraints:

$$\hat{q}_{min} \leq \hat{q} \leq \hat{q}_{max}$$

(29)

$$\Delta \hat{\theta}_d(k) = \hat{\theta}_d(k) - \hat{\theta}_d(k - 1)$$

(30)

$$|\Delta \hat{\theta}_d| \leq |\Delta \hat{\theta}_{max}|.$$  

(31)

The values for $\hat{q}_{min}$ and $\hat{q}_{max}$ are determined by the current states and the joint limits as follows:

$$\hat{q}_{min} = q_{min} - q$$

(32)

$$\hat{q}_{max} = q_{max} - q.$$  

(33)

The limit $|\Delta \hat{\theta}_{max}|$ was determined by the slope of the system response to a step input. This limit is necessary, similar to LQR, because the pressure dynamics are not currently part of the system model.

This optimization can be solved at rates up to 200 Hz with a horizon of $T = 65$ time steps using C code generated by the online convex optimization code generation tool CVXGEN [30]. The first input from the optimized input trajectory is applied as control to the actual system and the optimization is then reformulated and runs again at the next time step with updated states from system measurements. In Equation
(28), $Q_1$ weights the error over the horizon, $Q_2$ weights the final error, $Q_3$ weights the final velocity, $R_1$ weights the inputs over the horizon, $R_2$ weights the change in input over the horizon and all were tuned manually for acceptable performance.

D. Results and Discussion for Single Joint

![MPC and LQR Response for Single Joint](image)

Fig. 6: Single Joint Control Response

The MPC response, as well as the LQR responses with and without an integrator can be seen in Figure 6 for a single joint. The MPC response shows less overshoot and better tracking of the goal than the LQR response which for some inputs relies on the integrator for accurate tracking. However the LQR response is faster and there is less oscillation around the goal. The LQR response shows that the system model can be used for acceptable performance. We expect that further tuning of the weights for either controller or a dynamic model that accounts for the pressure dynamics would result in better performance. Two advantages that MPC has over LQR is the ability to add constraints, and add different terms to the cost function without major changes to the system model.

V. Multiple Joint Dynamics and Control

As shown previously MPC and LQR are both acceptable preliminary position control methods for a single degree of freedom joint. Although, the performance between the two was fairly comparable, there are pressure saturation limits and pressure dynamics that are currently not included in our control formulation. This along with other constraints (such as joint limits) that are currently included in our MPC model lead us to expect that MPC will be a better long term solution than LQR. However we have used both MPC and LQR methods developed for a single joint to develop and implement MPC and LQR controllers for King Louie (a 5-DOF inflatable robot).

A. Model

The same rigid body model that was used for the grub was used for each joint on King Louie. Each joint is treated independently from the rest, even though inertia and gravity have a much greater effect than they did on a single joint. However, ignoring these effects in the model and treating them as disturbances does not prevent successful control of the arm. This is in part due to the low total weight of the arm which is estimated to be between 10 and 15 pounds. When compared to the actuator pressures which for this work have a maximum value of 17 PSI gage, the reaction forces between the opposing bladders are expected to be orders of magnitude greater than the forces due to mass and gravity so the system inputs will have a dominating effect on the system response.

B. Sensing

For King Louie we used motion capture sensors for controller development. We calculate the relative quaternion rotation between frames on each link. Assuming rigid joints and links, we decompose the quaternion describing the rotation between two links into the relevant Euler angles. The motion capture reflective markers used to measure the joint angles can be seen in Figure 8.

C. Control

Similar to control for the grub, we compared LQR and MPC for controlling five degrees of freedom simultaneously. We only control five degrees of freedom because we are treating the gripper as either opened or closed and we are not actively controlling the hip joints besides inflating both of them to maximum pressure to increase stability of the torso.

For both MPC and LQR, system parameters and weightings were tuned at each joint for acceptable performance. Each joint is controlled by a separate process and solves for MPC at 200 Hz for each joint. Because the MPC cost function can be minimized at 200 Hz both the MPC and LQR controllers were operating at 200 Hz. A higher level controller passes desired joint angles to each joint separately. For this work, only step commands are passed to the controllers. One benefit to this type of controller developed for a single joint is that MPC can run at faster rates with a longer time horizon. A single controller that accounts for all the joints and more of the system dynamics would have slower solve rates and necessarily shorter horizons because of computational limitations, severely limiting the low level controller bandwidth. However, accounting for the low level control response in a higher level controller for future work, where high bandwidth is not as important, may be beneficial for overall control.

D. Results and Discussion for King Louie

We were able to successfully implement MPC and LQR joint angle control for a multiple joint inflatable robot system. The system response to a step input sent to each joint can be seen in Figure 7. In Figures 8a and 8b, King Louie’s arm can be seen at the commanded angles. This orientation was chosen to show that King Louie can grab and manipulate objects in front of itself.
MPC and LQR tracked the commanded joint angles with similar trends as with one joint. MPC is slower to achieve the commanded angle but tracks well for all the joints with some oscillation. LQR has the faster response but does not track as well especially for Joint 2. This joint was fabricated differently than the other joints so that motion is restricted in one direction.

The slope of the sigmoid function in equation (4), could be manipulated through \( s_1 \) which was important for stability. Originally, the coefficients where chosen to match the profile in Figure 4, but were then tuned for each individual joint. A steeper slope was necessary for the joints at the wrist which is likely because these joints are much smaller than the shoulder joints, and are more sensitive to changes in pressure. A steeper slope forced the inputs towards the center of the pressure range for these joints. A more shallow slope was necessary for the shoulder joints where the full pressure range was needed to compensate for disturbances caused by the mass of the arm.

Using MPC, joint space commands were given in a sequence to perform a task. The task involved picking up a board from a chair and placing it in a box behind King Louie (See attached video or https://youtu.be/o044-KW921I). The board weighs 1.38 pounds (.63 kg), is 20.5 inches (52 cm) long, 3.5 inches (9 cm) wide, and 1.5 inches (4 cm) thick. The task was completed eight out of ten times successfully. For one of the failed attempts, the board bounced off the top of the box and fell out of the box. For the other failed attempt, some of the IR reflective markers were not identified by the motion capture system which caused our joint estimation to fail after already successfully retrieving the object but before placing it in the box.

VI. CONCLUSIONS

We have successfully demonstrated the viability of a Linear Quadratic Regulator and Model Predictive Control in controlling an inflatable humanoid robot. These systems show great potential for high levels of human-robot interaction due their innate compliance and low inertia. In the future we expect this type of robot will work together with humans to perform many tasks. The preliminary work demonstrated in this paper is a first step towards that goal and towards improved performance for soft robot control in general.

ACKNOWLEDGEMENT

This work was supported by an Early Career Faculty grant from NASAs Space Technology Research Grants Program. We also wish to acknowledge the hard work of Kevin Albert and his team at Pneubotics (a spinoff of Otherlab Company) for collaboration in regards to our soft robot hardware platform.

REFERENCES
