Performing Explosive motions using a multi-joint arm actuated by pneumatic muscles with quasi-DDP optimal control

G. Kumar Hari Shankar Lal Das\textsuperscript{1,2}, B. Tondu\textsuperscript{1,2}, F. Forget\textsuperscript{1}, J. Manhes\textsuperscript{1}, O. Stasse\textsuperscript{1}, P. Soueres\textsuperscript{1}

Abstract—New actuators are expected to endow robots with the capability to execute explosive movements such as throwing, kicking or jumping. In the present paper, we present a ball throwing and kicking tasks performed by an anthropomorphic arm with each joint actuated by an agonist-antagonistic pair of Mckibben artificial muscles. Pneumatic actuators have inherent compliance and hence they are very interesting for applications involving interaction with environment or human. However, the control of such systems is known to be difficult. The paper presents an implementation of iterative Linear Quadratic Regulator (iLQR) based optimal control for such actuators. The method is applied to positioning tasks and generation of explosive throwing of a ball to a maximum distance and kicking a ball. It is then compared to traditional control strategies to justify that optimal control is effective in controlling the position in highly non-linear pneumatic systems. The algorithm validation is reported here by several simulations and hardware experiments in which the shoulder and elbow flexion are controlled simultaneously.

I. INTRODUCTION

With the emerging robotic technology, robots are expected to perform high performance task in the presence of human. So safety has become a major concern. In order to address this issue while executing high performance tasks, compliance at the actuator level would be required. Pneumatic actuators are inherently compliant and have very high power to weight ratio. These factors motivate researchers to use pneumatic actuation for exoskeleton, prosthesis, rehabilitation devices or even for walking robots and human robot interaction. The goal of this paper is to use an arm actuated by McKibben muscles to perform explosive tasks such as throwing or kicking a ball. Recent works [1] advocate other kind of actuation systems which are easier to control but come with a high price tag. Our goal was to demonstrate a kind of actuation systems which are easier to control but come with a high price tag.

The method is applied to positioning tasks and generation of explosive throwing of a ball to a maximum distance and kicking a ball. It is then compared to traditional control strategies to justify that optimal control is effective in controlling the position in highly non-linear pneumatic systems. The algorithm validation is reported here by several simulations and hardware experiments in which the shoulder and elbow flexion are controlled simultaneously.

II. DYNAMIC MODEL OF A JOINT ACTUATED BY McKIBBEN MUSCLES

There have been several attempts to model the McKibben artificial muscle [4], [7], [8], [9], [10]. These attempts were made to capture an accurate model to a combination of complex phenomena during static and dynamic contraction. When we want to include such a physical model into the closed-loop control of a McKibben muscle actuator, a compromise must be found between the complexity of an accurate model and the time required for its computation. The need for an efficient but not too complex dynamic muscle model is all the more important in the case of artificial muscles for robot arms. Keeping this in mind, the simplified model of the joints actuated by a pair of agonist-antagonist McKibben muscles. The capability of the non-linear iLQR control to execute simple tasks such as reaching a point, maximizing the velocity of joint or end effector, kicking or throwing a ball are demonstrated on this platform.

The paper is organised as follows: The dynamical model of joints actuated by McKibben muscles and experiments in which the shoulder and elbow flexion are briefly described. In the present paper, we present a ball throwing or kicking a ball. It is then compared to traditional control strategies to justify that optimal control is effective in controlling the position in highly non-linear pneumatic systems. The algorithm validation is reported here by several simulations and hardware experiments.
where,

- \( l \) being the current muscle length and \( l_o, r_o, \alpha_o, \varepsilon, a, b \) are the geometric parameters of the muscle. Please refer to the figure Fig. 1 and see [7] and [11] for more details. The parameter \( k \), slightly greater than 1, is empirically chosen for taking into account the conic shape at muscle tips. The viscous coefficient \( f_v \) is also empirically estimated from a mean linear second-order identification of the McKibben muscle contraction in response to given pressure steps. Control pressure in a muscle is provided by an Intensity-Pressure (I/P) converter which translates a current value into a desired pressure value that has to be fed to the muscle. In literature, the model of pressure variation are highly non-linear and eventually become too cumbersome for devising control strategies [12] [13]. We propose a somewhat simpler empirical model to cover the dynamics of the pressure generation from the I/P converter. It is derived from SAMSON I/P converters reported in [14]. Following this model the instantaneous pressure \( p \) inside a muscle is represented by a damped second order differential function with a control pressure as input as follows:

\[
\ddot{p} + 2w_n \dot{p} + w_n^2 p = w_n^2 p_{des}
\]

Note that the input of the intensity converter is a current value which is then scaled to the corresponding pressure control input \( p_{des} \) in the pressure unit. The above equation is also non-linear as the natural frequency \( w_n \) has been empirically identified as a function of the volume of the muscle, \( V \) which in turn depends on the geometry of the muscles [15].

\[
w_n = 2\pi f_v(1/V)
\]

\[
V = \frac{\pi r_o^2 l \sin^2(\alpha_o)}{1 - \cos^2(\alpha_o)(1 - \varepsilon)^2}
\]

For a second order system, the rise time is inversely proportional to its natural frequency and in the case of our pressure dynamic model, the natural frequency is inversely proportional to the volume. It implies that the bigger the muscle, the larger the rise time. Moreover a constant \( w_n \) could be chosen if the contraction is small (i.e small joint operational range).

A. Agonistic-antagonistic joint actuator

A pair of artificial muscles can be set up in antagonistic fashion to drive a chained wheel of radius \( R \). According to Fig. 2, the resulting actuator torque \( T \) can be written as follows:

\[
T = R[F_1(\varepsilon_1, P_1) - F_2(\varepsilon_2, P_2)]
\]

where \( F_1 \) and \( F_2 \) are the forces of muscles 1 and 2 respectively, defining the antagonist muscle pair.

As illustrated in Fig. 2, we formulate a generic expression for the actuator torque in terms of joint angle and control input by using Eq.(1), and neglecting the terms \( \varepsilon_1^2 \) and \( \varepsilon_2^2 \) as

\[
T = K_1 \Delta P_1 - K_2 \Delta P_2 - K_3 (P_1 + P_2) \theta - K_4 \dot{\theta} + K_5,
\]

where,

\[
K_1 = (\pi r_o^2)R[a(1 - 2k\varepsilon_{10}) - b],
K_2 = (\pi r_o^2)R[a(1 - 2k\varepsilon_{20}) - b],
K_3 = 2(\pi r_o^2)R^2k_o/l_o,
K_4 = (\pi r_o^2)R^2 f_v/l_o,
K_5 = (\pi r_o^2)R[(a - b)(P_{10} - P_{20}) - 2k_o(P_{10}\varepsilon_{10} - P_{20}\varepsilon_{20})].
\]

If \( P_{10} = P_{20} \) and \( \varepsilon_{10} = \varepsilon_{20} \), \( K_5 \) is equal to zero. The details of our robot-arm including the dimensions of muscles are given in [2]. The antagonist muscle actuator is now considered as a MIMO-system whose inputs are the control pressures \( \Delta P_1, \Delta P_2 \), and outputs are both the \( \theta \) and the actuator stiffness which is defined as the instantaneous ratio between the current torque variation and the current angular position variation (see Eq.(8)). When no gravity effect is considered, the static equilibrium position of the actuator can be directly derived from Eq.(6) with zero angular velocity.

\[
\theta_{equ} = (K_1 \Delta P_1 - K_2 \Delta P_2 + K_3) / (K_3 (P_1 + P_2))
\]

With associated stiffness \( \sigma_{equ} \) expressed as

\[
\sigma_{equ} = -\frac{\partial T}{\partial \theta} = K_3 (P_1 + P_2)
\]

From Eq.(7) and Eq.(8), it is possible to remark that the equilibrium position can be changed while keeping the same stiffness by modulating \( \Delta P_1 \) and \( \Delta P_2 \) with a constant \( \Delta P_1 + \Delta P_2 \). In the case of a symmetrical pressure variation in both

![Fig. 1. Muscle geometric parameters: l₀ initial length, r₀ radius and α₀ braid angle](image)

![Fig. 2. Pulley-chain driven antagonist muscle actuator made of two identical McKibben artificial muscles. It has the possibility to adapt initial torque at initial zero θ-position by means of P₁₀, P₂₀, ε₁₀ and ε₂₀](image)
muscles: $\Delta P_1 = -\Delta P_2 = \Delta P$, the actuator becomes a SISO-system whose corresponding torque $T_{SISO}$, is now given by the following relationship:

$$T = (K_1 + K_2)\Delta P - K_3(P_{10} + P_{20})\dot{\theta} - K_4\dot{\theta} + K_5$$

(9)

where stiffness at equilibrium position is now constant and equal to $K_3(P_{10} + P_{20})$.

We can now extend the model to a generic multi joint robot.

B. Robot dynamics

This section presents the robot model formally including the rigid body model of the robot with its pressure dynamics. Let us consider a $n$ degrees of freedom robot with generalized joint angle coordinates $q \in \mathbb{R}^n$. Each joint is actuated by 2 pneumatic muscles, so there will be $2n$ muscles, each one with a pressure $P_i \in \mathbb{R}^{2n}$. The robot dynamics can be represented in standard form as below:

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) = T(q, P)$$

(10)

$$\dot{P} + 2C_p\dot{P} + G_pP = G_pP_{des}$$

(11)

$M(q) \in \mathbb{R}^{n \times n}$ is the mass inertia matrix of the robot, $C(q, \dot{q}) \in \mathbb{R}^{n \times 4}$ is the coriolis and centrifugal terms, $G(q) \in \mathbb{R}^n$ is the gravity related terms, $T(q, P) \in \mathbb{R}^n$ is the torque generated by pneumatic muscles as described in Eq.(5). $C_p = diag[w_{n1}, w_{n2}, \ldots, w_{n2n}]$ and $G_p = diag[w_1^2, w_{2}^2, \ldots, w_{2n}^2]$ are the collection of coefficients of the pressure dynamics of each muscle and $w_{ni}$ is the natural frequency of the pressure dynamics at the $i^{th}$ muscle. $P = [P_1, P_2, \ldots, P_{2n}]^T \in \mathbb{R}^{2n}$ is the vector of current muscle pressure and $P_{des} = [P_{des1}, P_{des2}, \ldots, P_{des2n}]^T \in \mathbb{R}^{2n}$, is the vector of desired pressure $P_{desi}$, the control input in pressure unit given to the I/P converter of the $i^{th}$ muscle. The operating range of the joints are limited by the maximum contraction of the spanning muscles. Apart from this, due to hardware limitation of I/P converter, the control input $P_{des}$ is bounded as follows:

$$U = P_{des} \in \mathbb{R}^{2n} : P_{des} \in [P_{min}, P_{max}]$$

(12)

Where, $P_{min}, P_{max}$ are the lower and upper bound on control input $P_{des}$.

III. OPTIMAL CONTROL FORMULATION

This section presents the optimal computational framework used to find the control sequences to perform a desired task. The optimal control problem is the minimization or maximization of a performance criterion with respect to the control under a set of constraints that arise from the physical limitation of the control action and from the plant dynamics.

A. State space representation

Let us represent the dynamics stated in Eq.(10) and Eq.(11) in state space form considering the state vector as $x = [q, \dot{q}, P, \dot{P}]^T$.

$$\dot{x} = f(x, u) = M^{-1}[-C(q, \dot{q}) - G(q) + T(q, P)]$$

(13)

$$\dot{P} = M^{-1}[-C(q, \dot{q}) - G(q) + T(q, P) - C_p\dot{P} - G_pP + G_pP_{des}]$$

where, $f$ is the non-linear function given by Eq.(13) in state $x$ and control $u$ that satisfies Eq.(10) and Eq.(11). In the present work, we consider the constraints on the state in the optimal control formulation. There exists a mechanical limit to each degree of freedom of the arm. For the operation to be safe, we have introduced these limits as constraints on the state space inside the cost function of the optimal control formulation. The chosen cost function for the state space constraints is expressed by the following equation:

$$C_s = \lambda_s^{max} + \lambda_s^{min}$$

(14)

where,

$$max = 1 - \lambda(x_{max} - x),$$

$$min = 1 - \lambda(x - x_{min}),$$

$x_{min}, x_{max}$ are lower and upper limits on the state and $\lambda$ is a constant. The above consideration for the cost function will ensure that the cost near the limits will be very high as evident in the Fig. 3 and hence the optimal solution will keep the system within the operating limits. Also, the control action has to be admissible, i.e $u \in U = [P_{min}, P_{max}]$. The control problem formulation is then expressed as determining an open-loop control input $u = u(t, x) \in U$ which can minimize or maximize a cost function along a given time interval $t \in [0, T]$ and with initial state $x(0) = x_0$. For a non-linear dynamics Eq.(13) and non-quadratic cost Eq.(17), optimal control solutions can be obtained using full DDP. However, as DDP is computationally expensive, an iterative LQR (iLQR) approach is considered [16]. The iLQR method relies on linearizing the dynamics and approximating the cost function to quadratic form along the trajectory. This control approach is briefly summarized in the next section.

B. Iterative Linear Quadratic Regulator (iLQR)

iLQR is initialized with a nominal control sequence and the corresponding state trajectory $(x_0, u_0)$. The dynamical

![Fig. 3. Cost function of the state constraints](image-url)
system is then linearized as in Eq.(15) and the cost function is approximated by the quadratic form Eq.(16) and a local LQR problem is then solved. Using this solution, the states and the control sequence are improved iteratively.

\[
\Delta \dot{x} = A \delta \dot{x} + B \delta u + c_x \delta \dot{x} + c_u \delta u
\]

\[
\Delta J = h^T \delta x(T) + \frac{1}{2} \delta x^T(T) h \delta x(T) + \int_0^T c_x^T \delta x + c_u^T \delta u + \delta x^T c_x \delta u + \delta u^T c_u \delta u \tag{15}
\]

where \( A = \frac{\partial f}{\partial x} \) and \( B = \frac{\partial f}{\partial u} \). In Eq.(15) and Eq.(16) subscript \( x \) and \( u \) indicate that the function is partially derivated with respect to \( x \) and \( u \). At every iteration, Eq.(15) and Eq.(16) are solved and \( (\delta x, \delta u) \) are deduced from the resolution of a modified Ricatti-type system. Then the new improved sequence is generated by \( x \leftarrow x + \delta x \) and \( u \leftarrow u + \delta u \). When \( \Delta J \approx 0 \), the iLQR converges and gives an optimal control sequence \( u^* \) ∈ \( U \) and the corresponding optimal state trajectory \( x^* \).

### IV. Simulation and Experimental results

The experimental set-up considered here is the manipulator of LAAS CNRS which is a seven degrees of freedom (DoF) anthropomorphic arm, where each joint is pneumatically actuated by a pair of Mckibben muscles. In the experiments presented in this paper, only two joints are controlled. These joints are the flexion \( \theta_1 \) at the shoulder and the flexion \( \theta_2 \) at the elbow. (see Fig. 4). So, for the experiments, the robot can be viewed as a 2 DoF manipulator with state defined by \( x = [\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2]^T \). Each muscle is pressurized by an I/P converter which converts a current command to a reference pressure value. The objective of the experiments is to evaluate the performance of the iLQR control to achieve the following tasks with our multi-link pneumatic robot.

1) Final position control: We aim to compare the quality of the positioning control with the iLQR approach and with the feed-forward proportional control. Simulations and experiments are done in the case when the sum of the pressure in agonist-antagonist pair of muscles at each joint is kept constant, i.e. \( P_1 + P_2 = \text{Constant} \). For this task, we also analyze the stiffness modulation and we show that the optimal control approach enhances the smoothness of the motion by simultaneous modulation of position and stiffness.

2) Capability to execute explosive movements by throwing a ball to a maximum distance. For this second task we compare simulation results with the results of experiment executed with the real robot. In all of the experiments, the most generic expression for the cost function can be written as follows:

\[
J(x_0) = C_f + C_r + C_s
\]

where \( C_f = h(x(T)) \) is the final cost, \( C_r = \int_0^T c(x(t),u(t,x(t)))dt \) is the integral of the running cost \( c(x,u) \) which encapsulates the task objectives and \( C_s \) is the cost value given by Eq.(14) imposed by the constraints.

#### A. Task 1: Position control

The manipulator is given a final position with a load mass of \( m_l = 0.1 \text{kg} \) at the end effector. iLQR is used here to find the optimal path to reach the final goal \((\theta_1, \theta_2) = (28.8, 57.6)\text{degrees}\).

The following cost function is considered for this task.

\[
C_f = Q_f (x_{ref}(T) - x(t))^2,
\]

\[
C_r = Q \int_0^T ((x_{ref}(T) - x(t))^2 + u^2(t))dt,
\]

where, \( x_{ref} \) is the final position for the two joints. The iLQR control uses Eq.(13) and Eq.(18) to solve for optimal control sequence \( u^* \). This optimal control sequence in forward application to Eq.(13) will yield the needed trajectory. To compare the effectiveness of the iLQR approach, the task is executed using a Proportional-Integral controller with a feed-forward term. The feed-forward term gives a desired pressure for each muscle at the joint needed to maintain a desired joint angle. Thus, the control action of the feed-forward PI controller can be defined as follows:

\[
u(t) = P_{feed} + K_p (e(t)) + K_i \int_0^t (e(t))dt \tag{19}
\]

where, \( e(t) = x_{ref} - x(t) \). \( K_p \) and \( K_i \) are proportional and integral gains respectively. The simulation results are shown in Fig. 5. Response of feed-forward PI controller is shown in black dashed line is compared with the responses of optimal position control in Case-I when \( P_1 + P_2 = \text{Constant} \) (blue lines).

From the position plots in Fig. 5, it appears that the iLQR approach gives a good compromise between keeping the stiffness low and minimizing the oscillations which results into a smooth motion. However, the feed-forward PI controller makes joint 1 very stiff and joint 2 very flexible leading to an overshoot and oscillations.
B. Task 2: Maximizing the link speed

The objective is to execute some explosive motions with the aim to perform in the future tasks such as ball throwing, kicking or hammering a nail. Such motions would require either maximizing the joint/link speed or end-effector speed [17], [18]. So the task considered here is maximizing the angular speed of the elbow joint. The task is first simulated and then executed on the real robot. A comparison between simulation and the experimental results are shown in Fig. 6.

For maximizing the elbow joint’s angular speed at final time $T = 1$, the task requires only the terminal cost, $C_f$. But to minimize the control effort, a running cost $C_r$ involving only control pressures is used.

$$C_f = -Q_f(x_4(T))^2,$$

$$C_r = Q_a \int_0^T u^2(t)dt,$$  \hspace{1cm} (20)

where, $x_4 = \dot{\theta}_2$ is the angular velocity of the elbow joint which constitutes the fourth state variable. $Q_f$ and $Q_a$ are weights for the terminal cost and the running cost. The simulation results are shown in Fig. 6.

In simulation, the maximum joint velocity reached at terminal time is around -15 rad/sec. The robot response and the simulation response in position and speed show a good match. However, some discrepancies can be observed.

C. Task 3: Ball throwing and kicking

The objective is to throw a ball to a maximum distance. The distance to be thrown is related to the kinematics of the arm and hence is a function of the state of the arm. The cost function, thus, comprises the terminal cost $C_f$ involving the distance and the running cost $C_r$ involving only control efforts. Also the final time of the motion is fixed at $T = 4$ seconds.

$$C_f = d = kf(x),$$

$$C_r = Q_a \int_0^T u^2(t)dt,$$  \hspace{1cm} (21)

where, $kf(x(t))$ is the function which computes the maximum distance to which the ball will be thrown. This function relates the motion of the ball with the robot kinematics. We used the same cost function in a second experiment for the task of kicking a ball stationed at a point. Snapshots of the video of both the experiments are presented in Fig. 10 and Fig. 11.

Simulations are done for the case where the sum of pressures in the agonist-antagonistic pair is kept equal to 4 bar at the joint 1 to 4 bar and 5 bar at the joint 2. We present the simulation results when there are no constraints on the state of the robot (Fig. 8). However, the robot has limits at its joints which is typically between $[-28.5, 114.6]$ degrees. It is evident from Fig. 8 that it is necessary to include the constraints in order to make the real robot execute the task.

Plots in Fig. 9 compare the response of the real robot with the simulated response of the model when optimal control inputs are applied in the open-loop. It is interesting to note that the optimal solution given by the iLQR is somewhat intuitive to the human behaviour. It takes few swings to attain the maximum speed of the end effector before launching the ball. The maximum distance achieved by the ball was 2.3m against 3m predicted by the simulation.
The observed discrepancies between simulated and the real robot response can be justified by the fact that the system is controlled in open-loop without feedback. Also we have no model of the various effects of hysteresis and dry friction. Moreover, model of the gripper is not available so the exact time of opening of the gripper is not known. It has been roughly estimated by observation for the ball throwing experiment. Since modeling all the aspects of pneumatic muscles actuators seem to be difficult, it will be interesting in the future to rely on the close loop control.

The hardware set up used for the experimentation is the robot mentioned in [2] whose control modules have been upgraded recently. The important components of the current set up are I/P converters, encoders, NI data acquisition devices and the development computer running real time control software. 1) I/P converter : It is a Samson I/P 6111 and produces output pressure in range 0 to 5 bar. There are 7 joints and a gripper. Each joint and the gripper are actuated by a pair of Mckibben muscles and each muscle is controlled by one I/P. 2) Encoders/Potentiometers: There are seven potentiometers, one at each Dof. 3) The data Acquisition device: CompactRIO from National Instruments is used for this purpose. It has NI9205 module which is an Analog-to-Digital-Converter (ADC) with up to 32 channels. This device samples the seven potentiometers. There is a NI9265 card which has Digital-to-Analog-Converters (DAC) and capable to provide analog current outputs in the 0 to 20mA range.

4) The development computer: It runs the high level control algorithm for the robot. It communicates via the NI module using standard UDP protocol at a frequency of 1kHz in real time. The computer has linux 14.04 with Xenomai real time kernel running in parallel. For the present experimentation, control software are running at 200Hz frequency.

V. CONCLUSION AND FUTURE WORK

We addressed the problem of modeling and controlling a robot manipulator actuated by Mckibben pneumatic muscles in an optimal control framework. Our first contribution was to propose a model which encapsulates the pressure dynamics in an efficient way. It is done by using a second order differential function whose bandwidth is dependent on the instantaneous volume of the muscles. Even though, the model does not include the effect of dry friction and hysteresis, it covers most of the static and the dynamic behavior. Using this model, along with the robot manipulator model, our second contribution was to show that the implementation of iLQR allows to perform efficient position control and also preserves the inherent compliance of the Mckibben muscles with respect to the more conventional control approaches. Having embedded the constraints on the state in the optimal control formulation, we have used this model to perform various tasks such as positioning control and maximizing the link speed. We also have shown the capability of the robot arm to perform explosive motions by throwing a ball. The results were reported in simulation first and then validated on the hardware platform. In future work, we plan to use this control approach to perform nail hammering with this pneumatic arm. This will be done in real time optimal control in close loop. A detailed analysis of stiffness variation and possibilities of stiffness control will then be possible.

VI. ACKNOWLEDGEMENTS

This work was partially supported by the European FP7 Project EUROC under Grant Agreement 608849 and the ERC Advanced Grant Actanthrope. We also thank Kevin Giraud Esclaide for his help in software development.

REFERENCES


