ILQR implementation on non-linear pneumatic arm actuated by agonist-antagonist pair of Mckibben muscles

Ganesh Kumar
Gepetto
LAAS-CNRS
Toulouse, France
Email: gkharish@laas.fr

Olivier Stasse
Gepetto
LAAS-CNRS
Toulouse, France
Email: stasse@laas.fr

Bertrand Tondu
Gepetto
LAAS-CNRS
Toulouse, France
Email: tondu@laas.fr

***Abstract*—The paper presents an implementation of ILQR method to control a joint actuated by agonist-antagonistic pair of Mckibben artificial muscles. The method is applied to the elbow joint of an anthropomorphic 7dofs arm at LAAS-CNRS. It is then compared to the traditional non-linear control method to justify that ILQR is effective in controlling highly non-linear dynamical system even in the absence of good non-linear model which is anyway hard to model in the case of Mckibben muscles.**

***Keywords*—Mckibben Muscle, ILQR, non-linear control, Anthropomorphic Pneumatic arm.**

I. Introduction

Pneumatic actuators based on Mckibben artificial muscle is known for their non-linearities and hence pose a great control challenge. The sources for non-linearities varies from hysteresis, saturation and internal friction between fabrics. However, having inherent compliance and very high power to weight ratio have made us to chose it as a research platform for the study compliance control and exploiting this to perform explosive works like hammering or throwing balls. Modelling the non linearities and getting a good model for the joint actuated by artificial muscles is not an easy task. The aim of the paper is to demonstrate that ILQR is effective in controlling joint actuated by Artificial Mckibben muscle even in the absence of good non-linear mode by considering an approximate linear identified model. The proposed method is then compared to the more traditional non-linear control method adopted for controlling the mckibben muscles like sliding mode controller. Introduction of the experimental setup and the dynamical model of our joint actuated by Mckibben muscles is presented in section II. Section III will describe the identified linear model which will be used as an approximation of the non-linear model. ILQR algorithm developd by Li and Todorov [1] is presented in section IV. Application of ILQR on the Mckibben model is then explained. Experimental results and future work will be discussed in section V and VI.

II. Dynamic model of a joint actuated by Mckibben muscles

Mckibben muscle has inner tube which inflates under pressure P and braided shell surrounding it translates the inflation into axial contraction force F.

$$F(\epsilon, P) = (\pi r_0^2) P[a(1 - k\epsilon)^2 - b] \tag{1}$$

Where,

$$\epsilon = \frac{l_o - l}{l_o}$$
$$a = \frac{3}{\tan^2 \alpha_o}$$
$$b = \frac{1}{\sin^2(\alpha_o)}$$

The model in equation (8) assumes that the inner tube is initially cylindrical with length l_o, radius r_o and textile weave of initial braid angle α_o. l is the instantaneous length of the muscle. So any Mckibben muscle can be characterized by these three parameters. The empirical parameter k is introduced to overcome the error due to the cylindrical inner tube assumption. Following the human arm model, a pair of artificial muscles can be setup in antagonistic fashion to drive a chained wheel of radius R. The antagonistic model used for our pneumatic device is based on the model presented in [2] which gives the expression for generated torque, equilibrium position and restoring force as follows:

$$T = R[F_1(\epsilon_1, P_1) - F_2(\epsilon_2, P_2)] \tag{2}$$

Using equation (8), it can be expressed as

$$T = K_1(P_1 - P_2) - K_2(P_1 + P_2)\theta \tag{3}$$

Where, K_1 and K_2 depends on parameters l_o, r_o and α_o. θ is the angular position of the joint or the angular position of the chained wheel. Setting equation (10) to zero, equilibrium position can be found.

$$\theta_{eq} = \frac{K_1(P_1 - P_2)}{K_2(P_1 + P_2)} \tag{4}$$

and the restoring torque can be expressed as

$$T_r = -K_2(P_1 + P_2)\theta \tag{5}$$

From the above model, it is evident that the system is double input (P_1, P_2) and single output (θ). However, the objective is to explore the possibility of controlling the stiffness which could be expressed as

$$\sigma = -\frac{T_r}{\partial\theta} = K_2(P_1 + P_2) \tag{6}$$

So, the system could be considered as MIMO (double input (P_1, P_2) and double output (θ, σ). In the present paper, however, a simplified SISO system model is considered by the following assumptions. $P_m = P_1 + P_2$ and $\delta P = P_1 - P_2$. The optimal control capabilities would be explored for the applications like achieving a final position with maximum speed or in minimum time or energy.

III. Approximate linear model identification

The Pneumatic arm is highly non-linear and its very difficult to model these non-linearities. Also some mechanical parameters like friction coefficients and inertias of links are not available. A system identification is carried out to find an approximate linear model using MATLAB system identification toolbox. A set of input is given to the system in open loop and its transient behaviour is recorded to create input-output pair. The input is the pressure variation δP and the output is elbow's angular position. Few such sets have been collected. Using system identification toolbox, a linear model has been chosen. It has three poles and no zeros. Its transfer function can be represented as in equation ([**?**]). The following figure compares the step response of the real system with the identified model. The muscles are first initialised with some pressure before sending a step command that justifies the non-zero initial position in the plot. The transfer function of the identified model is given by the following equation.

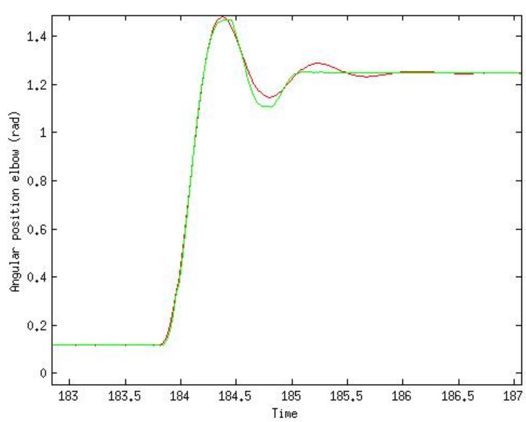

Fig. 1. Step response of elbow joint (green) and identified model (red)

$$H(s) = \frac{207.1}{s^3 + 10.73s^2 + 84.23s + 366} \tag{7}$$

$$\theta(s) = H(s) * (\delta P(s)) \tag{8}$$

The experiment is repeated at several operating point (x_i, u_i). At each operating point both the muscles is initialized with some pressure, however, sum of the pressure in muscle1 and muscle2 is kept constant at 4.2 bar which is just below the maximum capacity of our intensity pressure converter. We have adopted a transfer function model to find a linear model at every operating point. In order to fit to a transfer function model, which assumes zero initial state conditions, input-output data pair is created by offsetting the initial values at $t = 0$.

$$output(t) = x_i(t) - x_i(0); input(t) = u_i(t) - u_i(0) \tag{9}$$

This *inputoutput* time domain data pair is used in identification toolbox to find a transfer function of linear model. To simulate the identified model response and compare with the real system response, lsim function in matlab is used as follows:

$$Ir_i(t) = x_i(0) + lsim(Imodel_i, input(t), output(t); \tag{10}$$

Where, Ir is the identified response of the identified model I_{model} at i operating point.

In this way, we have a linear identified model at each operating point. In our experiments we have ten operating points. It is observed that for lower initial pressure, the system behaves linear since one single transfer function is able to fit the system response at several operating points. However, at higher initial pressure the system seems to be strictly non-linear.

A. Iterative Linear Quadratic Regulator (iLQR)

Let us consider the discretized version of the system dynamics of Eq. (**??**):

(11)

with, the cost function is defined as,

$$J_0 = \frac{1}{2}\left(\mathbf{x}_N - \mathbf{x}^*\right)^T Q_f(\mathbf{x}_N - \mathbf{x}^*) + \frac{1}{2}\sum_{k=0}^{N-1}\left(\mathbf{x}_k^T Q \mathbf{x}_k + \mathbf{u}_k^T R \mathbf{u}_k\right) \tag{12}$$

where,

- $\mathbf{x}_N$ describes the final state after each execution of the input $\mathbf{u}_k$.
- $\mathbf{x}^*$ is the given target state.
- Q and Q_f are the state cost weighting matrices.
- R is the control-cost weighting matrix.

Using the the equations (11) and (12), the iLQR approach proceeds iteratively by obtaining a nominal open loop trajectory $\mathbf{x}_k$ by applying an input $\mathbf{u}_k$. With an initial input sequence $\mathbf{u}_k = 0$, each iteration produces an improved $\mathbf{u}_k$ by linearizing the system dynamics around the sequence $(\mathbf{x}_k, \mathbf{u}_k)$ and solving a modified LQR problem. The iterations continue until a cost convergence is achieved.

Algorithm 1 ILQR (uk, xi, xf, Qf, Q, R, dt, nIter)

$cost_{curr} \leftarrow$ `ILQRCost` $(\mathbf{u}_k, \mathbf{x}_i, \mathbf{x}_f, Q_f, Q, R, dt)$
for $j = 1, \ldots n_{\text{Iter}}$ **do**
  $\delta\mathbf{u} \leftarrow$ `ILQRIterate` $(\mathbf{u}_k, \mathbf{x}_i, \mathbf{x}_f, Q_f, Q, R, dt)$
  $\mathbf{u}_k^{'} \leftarrow \mathbf{u}_k + \alpha\,\delta\mathbf{u}_k$
  $cost_{new} \leftarrow$ `ILQRCost` $\left(\mathbf{u}_k^{'}, \mathbf{x}_i, \mathbf{x}_f, Q_f, Q, R, dt\right)$
  if $cost_{new} - cost_{curr} < cost_{threshold}$ **then**
    `TerminateIteration`
  end if
  $cost_{curr}, \leftarrow cost_{new}$
  $\mathbf{u}_k \leftarrow \mathbf{u}_k$
end for
return $\left(\mathbf{u}_k^{opt}, cost^{opt}\right)$

In Algorithm 1, the `ILQRCost` procedure, computes the cost of the nominal trajectory using (12). The `ILQRIterate` implements the linearization procedure mentioned earlier. Eventually, once the cost converges, the `ILQR` procedure returns the optimal cost-to-go between the states $\mathbf{x}_i$ and $\mathbf{x}_f$.

IV. EXPERIMENTAL RESULTS

The above proposed algorithm is used to control the model identified in section III. The goal is to make the system achieve final position with the minimum amount of energy spend. Here let the final position be $Xf = 2.0$ starting from zero initial position. The figures below show the optimal trajectory give by ILQR in time domain and in the phase space. The system converges within 40 iterations in 200 steps with sampling interval of 0.01s.

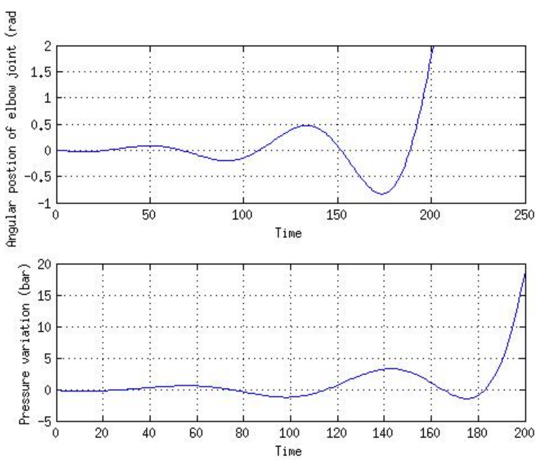

Fig. 2. Optimal trajectory give by ILQR for larger steps

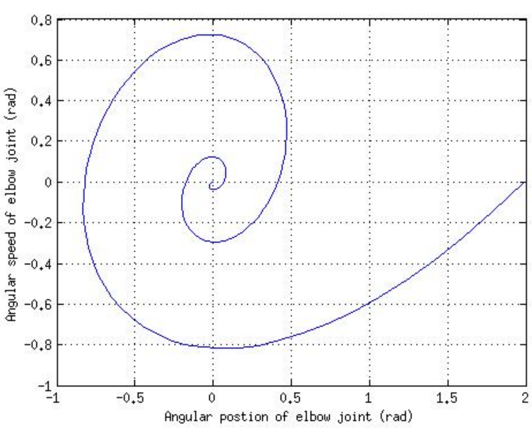

Fig. 3. Optimal trajectory in phase space

The system is able to converge even for 8 steps when sampling interval for ILQR is chosen to be 0.25. The above algorithm is not input constrained by itself so the above choice of sampling interval and number of steps is made to keep the input change in pressure within the limit (2.5 bar).

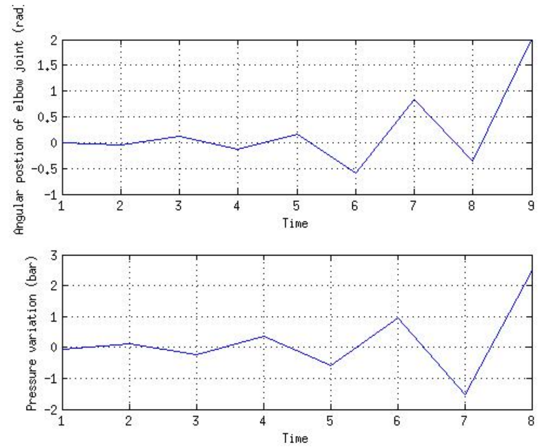

Fig. 4. Optimal trajectory given by ILQR

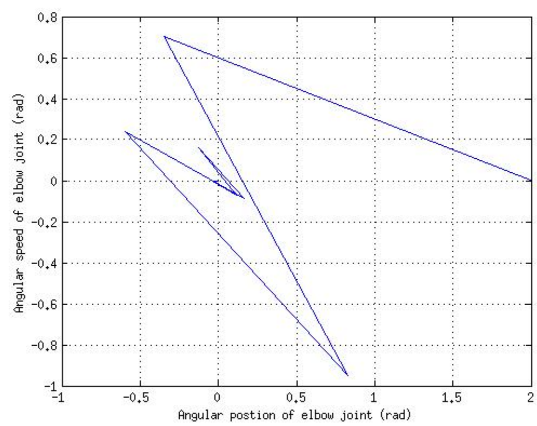

Fig. 5. Optimal trajectory in phase space

V. COMPARISON WITH MODEL BASED CONTROLLER

VI. CONCLUSION AND FUTURE WORK

APPENDIX A

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES

[1] L. WeiWei, E.Todorov, ""

[2] B.Tondu, S. Ippolito, J. Guiochet, "A Seven-degrees-of-freedom Robot-arm Driven by Pneumatic Artificial Muscles for Humanoid Robots," The International Journal of Robotics Research Vol. 24, No. 4, April 2005, pp. 257-274