Real-Time Predictive Control of a Brachiation Robot

Vinicius Menezes de Oliveira
Center for Computational Sciences
Federal University of Rio Grande (FURG)
Rio Grande, RS – Brazil
vinicius@ieee.org

Walter Fetter Lages
Department of Electrical Engineering
Federal University of Rio Grande do Sul (UFRGS)
Porto Alegre, RS – Brazil
w.fetter@ieee.org

Abstract

The present work addresses the problem of real time predictive control of a brachiation robot, considering that the robot is constrained and a multivariable system, which implies a very difficult problem due to the large amount of on-line computation that is required. We show that it is not possible to consider the nonlinear model-based MPC under real-time constraints. Furthermore, to overcome this problem we present a linearized model-based MPC, which is able to be handled under real-time constraints.

1 INTRODUCTION

In the beginning of the past decade a new class of mobile robot was proposed by Prof. Toshio Fukuda in [4], that imitates the movement of an ape swinging from branch to branch. For a better understanding about brachiation we indicate the work by Bertram [1].

The pioneer work introducing the brachiation control problem to the robotics literature was [10], presenting a simple two-link brachiation robot and they proposed a heuristic learning method for generating feasible trajectory for the robot. Fukuda, Hasegawa, Shimojima and Saito developed a self-scaling reinforcement learning algorithm to generate a feasible trajectory with robust property against some disturbances, while Saito added a feedback controller to improve its robustness [5]. The main drawback of this methodology is that it requires a long training period (about 200 experiments with the physical robot) to generate a successful maneuver.

Nakanishi, Fukuda and Koditschek took another approach, using target dynamics for control of underactuated systems[9]. In [6] is presented the control of the multilocomotion robot to execute brachiation, based on the previous local behavior control developed by the authors. An energy-based control to the swing phase of brachiation is presented in [7] The objective is to inject the minimum amount of energy into the robot during swing and locomotion phases. In [11] an energy-based control combined with Lyapunov stability theory is proposed.

In [8] a new control strategy for brachiation motion considering irregular ladder is proposed, based on Passive Dynamic Autonomous Control (PDAC). The idea is to develop an energy-efficient method that allows to gain the sufficient amplitude during the locomotion based on analysis with PDAC on the dynamics of the robot.

The present work addresses the problem of real time control of a brachiation robot using predictive control strategy. For complex, constrained, multivariable control problems, MPC has become an accepted control scheme, but for systems with fast and/or nonlinear dynamics, its implementation has remained fundamentally limited due to the large amount of on-line computation that is required. The robot considered in this paper is depicted in figure 1, which is slightly different from the Brachiator II robot, having two links acting like two arms and a third link acting like the body.

In our previous work [2] we have considered the application of the MPC strategy using the nonlinear model for the dynamics of the robot, which required a large amount of computational load to solve the non-convex optimization problem. We show in this work that the real-time predictive control using the nonlinear model is not possible to be carried out.

Thus, to overcome such limitation, we present in this paper the comparison between the nonlinear model-based and the linearized model-based real-time predictive control. This linearization is obtained computing an error model between the trajectory executed by the robot and the reference trajectory, previously designed.

In Section 2 we develop the nonlinear model for the
dynamics of the robot and in section 3.1 we formulate the nonlinear model-based predictive control strategy and in the next section 3.2 we present the linearized model-based predictive control scheme. The results are presented in section 4. Some discussions and considerations are carried out in section 5.

2 DYNAMICS MODELING

Considering the brachiation robot (seen in Fig. 1) as a serial open-chain robotic manipulator, its dynamics can be generally given by:

\[ M(\theta)\ddot{\theta}(t) + V(\theta, \dot{\theta}) + G(\theta) = F_v(\dot{\theta}) \quad (1) \]

where \( \theta = [\theta_1 \ \theta_2 \ \theta_3]^T \) is the vector of the joints coordinates, \( M(\theta) \in \mathbb{R}^{3 \times 3} \) is a symmetric matrix representing the inertia terms, given by\(^1\):

\[
M = \begin{bmatrix}
\alpha_1 + \alpha_2 + 2\alpha_3 \cos c_2 & \alpha_2 + \alpha_5 \cos c_2 & \alpha_5 + \alpha_3 \\
\alpha_2 + \alpha_5 \cos c_2 & \alpha_5 & 0 \\
\alpha_5 & 0 & \alpha_4 \\
\end{bmatrix}
\]

\( V(\theta, \dot{\theta}) \in \mathbb{R}^{3 \times 3} \) is a matrix with the Coriolis and centrifugal terms:

\[
V = \begin{bmatrix}
-\alpha_3 \dot{\theta}_2 \dot{s}_2 - \alpha_3 \dot{\theta}_3 s_3 & -\alpha_5 \dot{\theta}_2 \dot{s}_2 - \alpha_5 \dot{\theta}_3 s_3 & -\alpha_5 \dot{\theta}_3 s_3 - \alpha_3 \dot{\theta}_1 s_3 \\
\alpha_3 \dot{\theta}_2 s_2 & 0 & 0 \\
\alpha_5 \dot{\theta}_3 s_3 & 0 & 0 \\
\end{bmatrix}
\]

and \( G(\theta) \in \mathbb{R}^{3 \times 3} \) is a vector with the gravity terms given by\(^2\):

\[
G = \begin{bmatrix}
\alpha_6 c_1 + \alpha_7 c_{12} + \alpha_8 c_{13} \\
\alpha_7 c_{12} \\
\alpha_8 c_{13} \\
\end{bmatrix}
\]

and \( F_v \in \mathbb{R}^{3 \times 1} \) represents the viscous friction terms.

The constants are defined as:

\[
\alpha_1 = m_1 l_1^2 + l_1 + m_2 l_1^2 + m_3 l_1^2 \\
\alpha_2 = m_2 l_2^2 + l_2 \\
\alpha_3 = m_3 l_3^2 + l_3 \\
\alpha_4 = m_3 l_3^2 \\
\alpha_5 = m_3 l_3^2 \\
\alpha_6 = m_1 l_1^2 + m_2 l_1 + m_3 l_1 \\
\alpha_7 = m_2 l_2^2 \\
\alpha_8 = m_3 l_3^2
\]

The robot used in this work is designed as an underactuated system, i.e., a system in which the number of control inputs \( m \) is smaller than the number of degrees of freedom \( n \) \((m < n)\). In other words, the dimension of the input vector \( \tau \in \mathbb{R}^{m \times 1} \) is smaller than the dimension of the state vector \( \theta \in \mathbb{R}^{n \times 1} \). The input selection matrix \( B \) is a real matrix of dimension \((n \times m)\), defined as:

\[
B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (5)
\]

and the input vector is given by:

\[
\tau = \begin{bmatrix} \tau_2 \\ \tau_3 \end{bmatrix} \quad (6)
\]

The equations of the dynamics of the robot given by (1) can be rewritten in a more suitable state space form:

\[
s(t) = h(x, \tau) = f(x) + g(x)u \quad (7)
\]

where \( x = [\theta \ \dot{\theta}]^T \) is a \( 6 \times 1 \) vector of generalized coordinates and

\[
f(x) = \begin{bmatrix} -M^{-1}(x) \dot{V}(x) \dot{\theta} + G(x) + F_v(x) \end{bmatrix}
\]

and

\[
g(x) = \begin{bmatrix} 0_{3 \times 2} \\ M^{-1}(x)B \end{bmatrix}
\]

3 MPC SCHEME

The basic elements of the model-based predictive controller are: nonlinear (linear) prediction model, nonlinear (linear) optimization procedure and objective function. The model is used to predict the future plant outputs, based on past and current values and on the proposed optimal future control actions. The optimization procedure is responsible of calculating the optimal future control, taking into account the objective function, as well as the input and/or state constraints. The objective function defines the criteria to be optimized (usually involving the future errors) in order to force the generation of a control sequence that drives the system as desired.
3.1 Nonlinear MPC Approach

In this control scheme the model considered to predict the real behavior of the robot is given by equation (7). The MPC approach for this case is depicted in figure 2.

Thus, the prediction is obtained according to 3:

\[ x(k + j + 1) = x(k + j) \]

\[ + T[f(x(k + j) + g(x(k + j)) \tau(k + j)] \]

with \( j \in [0, N-1] \), where \( N \) is the prediction horizon.

Formally, the nonlinear optimization problem can be defined as:

\[ x^*, \tau^* = \arg \min_{x, \tau} \Phi(k) \]  \hspace{1cm} (9)

subjected to:

\[ x(k)[k] = x_0 \]  \hspace{1cm} (10)
\[ x(k + j + 1) = x(k + j) + \]
\[ + T[f(x(k + j)] + g(x(k + j)) \tau(k + j) \]

\[ Cx(k + j) \leq c \]  \hspace{1cm} (11)
\[ Du(k + j) \leq d \]  \hspace{1cm} (12)

where \( x_0 \) is the measured state in the actual instant and \( \Phi \) is the objective function generically defined as:

\[ \Phi(k) = \sum_{j=1}^{N} x(k + j)kQx(k + j)k + \tau^T(k + j - 1)[kR \tau(k + j - 1)] \]

where \( N \) is the prediction horizon and \( Q \geq 0, R > 0 \) are weighting matrices. A more detailed discussion about the objective function will be made later.

Equation (12) denotes the existence of a limit in the state magnitude, which in our particular work, denotes a limit in angular displacement and angular velocity of each joint, that is:

\[ x_{min} \leq x(k + j) \leq x_{max} \]  \hspace{1cm} (15)

where \( x_{min} \) and \( x_{max} \) are the minima and maxima values, respectively, for the state. Generally it is possible to describe such constraints as:

\[ Cx(k + j)k \leq c \]  \hspace{1cm} (16)

\[ \text{with} \quad C = \begin{bmatrix} I \end{bmatrix}, \quad c = \begin{bmatrix} x_{max} \\ -x_{min} \end{bmatrix} \]  \hspace{1cm} (17)

Analogously, the input constraints can be described as:

\[ D \tau(k + j) \leq d \]  \hspace{1cm} (18)

\[ \text{with} \quad D = \begin{bmatrix} I \end{bmatrix}, \quad d = \begin{bmatrix} t_{max} \\ -t_{min} \end{bmatrix} \]  \hspace{1cm} (19)

Therefore, the optimization problem (9) must be solved during each sample time \( k \), finding an optimal state sequence \( \{x^*(k + 1) \ldots x^*(k + N)\} \) and an optimal control sequence \( \{\tau^*(k) \ldots \tau^*(k + N - 1)\} \), besides the optimal cost \( \Phi^*(k) \).

The control input that will be applied to the system is the first element of the control sequence \( \tau^* \) obtained from the optimization step, defined as \( \tau^*(k) = \tau^*(k+1) \).

3.2 Linearized MPC Approach

The nonlinear system (1) that describes the dynamics of the brachiation robot (see figure 1) can be linearized by computing an error model with respect to a reference robot, which performed the reference trajectory, described by the following dynamic system:

\[ \dot{x}_r = f(x_r, \tau_r) \]

where \( x_r \) is the generalized reference coordinate vector and \( \tau_r \) is the torque reference vector.

By expanding the right side of (1) in Taylor series around the point \( (x_r, \tau_r) \) and ignoring the higher order terms we have:

\[ \dot{x} = f(x_r, \tau) + \frac{\partial f(x, \tau)}{\partial x} \bigg|_{x=x_r, \tau=\tau_r} (x - x_r) + \frac{\partial f(x, \tau)}{\partial \tau} \bigg|_{x=x_r, \tau=\tau_r} (\tau - \tau_r) \]

or

\[ \dot{x} = f(x_r, \tau_r) + F_{x \tau} (x - x_r) + F_{\tau \tau} (\tau - \tau_r) \]

where \( F_{x \tau} \) and \( F_{\tau \tau} \) are the Jacobians of \( f \) with respect to \( x \) and \( \tau \), respectively, evaluated around the reference point \( (x_r, \tau_r) \).

Now, subtracting (20) from (22) we have:

\[ \dot{\tilde{x}} = F_{x \tau} \tilde{x} + F_{\tau \tau} \tilde{\tau} \]

where \( \tilde{x} = x - x_r \) is reference tracking error and \( \tilde{\tau} = \tau - \tau_r \) is the error associated with the control input.

The discretization of (23) can be carried out using forward differences, resulting in:

\[ \tilde{x}(k + 1) = A(k) \tilde{x}(k) + B(k) \tilde{\tau}(k) \]  \hspace{1cm} (24)

\[ \text{with} \quad A(k) = I_{n \times n} + T \ast f_{x \tau}(k) \]
\[ B(k) = T \ast f_{\tau \tau}(k) \]

Considering the linear error system (24), it is possible to consider the optimization problem as a linear optimization problem and, therefore, solvable using quadratic programming form. Thus, we define the following vectors:

\[ T(k + 1) = \begin{bmatrix} \tilde{x}(k + 1) \\ \tilde{x}(k + 2) \\ \vdots \\ \tilde{x}(k + N) \end{bmatrix}, \quad T(k) = \begin{bmatrix} \tilde{\tau}(k) \\ \tilde{\tau}(k + 1) \\ \vdots \\ \tilde{\tau}(k + N - 1) \end{bmatrix} \]
The objective function can be stated as:

$$
\Phi(k) = x^T(k+1)Q x(k+1) + \tau^T(k)R \tau(k)
$$

with

$$
Q = \begin{bmatrix}
Q & 0 & \cdots & 0 \\
0 & Q & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & Q
\end{bmatrix}, \quad R = \begin{bmatrix}
R & 0 & \cdots & 0 \\
0 & R & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & R
\end{bmatrix}
$$

It is possible to rewrite (24) as:

$$
\tau(k+1) = \mathbf{X}(k)\vec{x}(k) + \mathbf{B}(k)\tau(k)
$$

where $\mathbf{X}$ and $\mathbf{B}$ are defined as:

$$
\mathbf{X}(k) = \begin{bmatrix}
A(k|k) & & \\
& \ddots & \\
& & A(k+1|k)A(k|k)
\end{bmatrix}, \quad \mathbf{B}(k) = \begin{bmatrix}
B(k|k) & \cdots & 0 \\
& \ddots & \vdots \\
& & B(k|k)
\end{bmatrix}
$$

with

$$
\alpha(k,j,l) = \frac{1}{j} \sum_{i=N-j}^{j} A(k+1|k)
$$

Considering (27) and (28) and after some manipulation, it is possible to rewrite (27) according to:

$$
\Phi(k) = \frac{1}{2} \tau^T H(k)^{\frac{1}{2}} \tau(k) + \tau^T(k) \tau(k) + g(k)
$$

with

$$
H(k) \triangleq 2(B^T(k)Q B(k) + R)
$$

$$
f(k) \triangleq 2(B^T(k)Q \mathbf{X}(k)\vec{x}(k))
$$

$$
g(k) \triangleq \vec{x}(k)\mathbf{X}(k)\vec{x}(k)
$$

The quadratic term is described by the positive definite Hessian matrix $H(k)$, the linear part is described by the vector $f(k)$ and $g(k)$ is independent of $T$ and can be ignored to the optimization problem. Moreover, we can redefine the cost function as:

$$
\Phi(k) = \frac{1}{2} \tau^T H(k) + \tau^T(k) \tau(k)
$$

in the standard form used in QP problems. Thus, the optimization problem to be solved at each sampling time is stated as follows:

$$
\hat{\tau}^* = \arg \min_{\tau} \{ \Phi(k) \}
$$

subjected to:

$$
D \tau(k+j|k) \leq d, \quad j \in [0, N-1]
$$

The amplitude constraints in the control variables (35) can be rewritten as:

$$
\tau_{\min} - \tau_r(k+j) \leq \hat{\tau}(k+j) \leq \tau_{\max} - \tau_r(k+j)
$$

Figure 3 depicts the control scheme using the linearized model.

### 3.3 Brachiation Robot Control

The NMPC scheme proposed in the preceding sections requires the description of the plant dynamics in discrete time. Thus, the system given by (7) can be discretized using Euler method and one can rewrite:

$$
x(k+1) = x(k) + T(f(k) + g(k)\tau(k))
$$

where $T$ is the sampling interval.

The objective function takes into account the Cartesian position and velocity of the end-effector of the robot, instead of considering directly the joint coordinates, because the robot must reach the supporting line ($y = 0$) with null velocity, independently of the joint configuration. It is important to highlight the fact that the robot is not fully actuated. In [3] a modified objective function is proposed, in order to increase the state penalty over the horizon. Also, a terminal state cost has been added to the cost function to be minimized. The modified objective function is given by:

$$
\Phi(k) = \sum_{j=1}^{N} X^T(k+j|k)Q(j)X(k+j|k) + \\
\sum_{j=0}^{N-1} \tau^T_r(k+j-1|k)R \tau_r(k+j-1|k) + \\
\Omega(X_r(k+N|k))
$$

with

$$
Q(j) = 2^{j-1}Q
$$

$$
\Omega(X_r(k+N|k)) = X^T_r(k+N|k)P X_r(k+N|k), \quad P \geq 0
$$

where

$$
X_r = \begin{bmatrix}
(x - x_{ref}) \\
(y - y_{ref})
\end{bmatrix}
$$

is the vector of cartesian coordinates error. The matrices $Q, R$ are real and have dimension $2 \times 2$ and $P$ is a real scalar.

### 4 RESULTS

In this section we present the physical parameters of the robot (see table 1), which are based on a simplified model for the multi-locomotion robot proposed by Fukuda Lab. In this simulation we have considered for the objective function the error in the cartesian coordinates of the end-effector and the input control as decision variables. So, the parameters for
The nonlinear version of the MPC control used in the simulation are defined as:

\[ Q = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}, \quad R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}, \quad P = 250 \]

and the constraints values are:

- maximum joint displacement \( q_{\text{max}} = \pi \) (rad);
- maximum joint velocity \( \dot{q}_{\text{max}} = 20 \) (rad/s);
- maximum torque \( t_{\text{max}} = 15 \) (Nm).

The parameters used for the linearized version of the MPC are defined as:

\[ Q = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}, \quad R = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \]

In order to validate our proposal, we carried out some real-time simulations and the results are given in the sequel. The developed software were implemented using the C++ language and to guarantee the real time execution of such programs we have decided to use the RTAI hard real time extension (version 3.6) (http://www.rtai.org) to the Linux kernel (version 2.6.22.15) (http://www.kernel.org). The optimization problem for the nonlinear model-based MPC was solved by using a modified version of the *donlp2* library and for the linearized model-based MPC it was solved using the available *OOQP* library.

We assume, for the simulations presented here, the robot is initially posed as follows: \( x_0 = [-1.4, -1.2, -1, -0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8] \).

---

**Table 1. Physical parameter considered for simulation.**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Joint 1</th>
<th>Joint 2</th>
<th>Joint 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass</td>
<td>Kg</td>
<td>3.5</td>
<td>3.5</td>
<td>5.0</td>
</tr>
<tr>
<td>Length</td>
<td>m</td>
<td>0.64</td>
<td>0.64</td>
<td>0.85</td>
</tr>
<tr>
<td>Viscous Friction</td>
<td>Nm/s</td>
<td>0.02</td>
<td>0.02</td>
<td>0.14</td>
</tr>
<tr>
<td>Coulomb Friction</td>
<td>Nm/s</td>
<td>0.02</td>
<td>0.02</td>
<td>0.45</td>
</tr>
</tbody>
</table>

---

**Figure 4. Cartesian trajectories of the robot \((N = 5)\).**

For the nonlinear model-based MPC we have considered two different prediction horizon \((N = 5\) and \(N = 3\)) in order to reduce the computational load to be handled within the sampling time (10ms). The complete motion of the robot for both prediction horizon can be observed in figures 4 and 5.

**Figure 5. Cartesian trajectories of the robot \((N = 3)\).**

The motion of the robot considering the linearized version of the MPC (considering a prediction horizon \(N = 5\)) can be viewed in figure 6.

**Figure 6. Cartesian trajectory of the robot (linearized model).**

The total time required to compute the control input (to solve the optimization problem) considering the nonlinear model-based MPC can be analyzed in figure 7. One can directly observe that the time required to calculate the control signal for each step is greater than the sampling time considered in the control loop. Now, considering the alternative approach to decrease the amount of time required to compute the control action proposed in this work we can observe in figure 8 that the time measured is smaller than the sampling period, which turns feasible the real-time implementation of the linearized model-based MPC for the control of a brachiation robot.

**5 CONCLUSIONS**

In this work we have addressed the control of an underactuated brachiation robot using the nonlinear and linearized model-based predictive approaches. Once the system is underactuated, we have to consider its constrained part to calculate the necessary control action to move the robot, which is one of the reasons why we have adopted the MPC strategy of control. We observed by numerical simulation that the controller was able to control the brachiation motion of the robot along the horizontal...
line, dealing with the nonholonomic constraint and respecting the limits imposed (torque, angular displacement, angular velocity).

However, we have verified that the real-time implementation considering the nonlinear approach was not feasible. Thus, to avoid such limitation, we have used the linearized approach to be able to execute this control scheme under real-time requirements.

Further analytical work will be carried out to completely understand the properties of the behavior of the robot regarding to the stability of the overall system. It is not simple once the motion of the robot does not converge to an equilibrium point of the system.

6 ACKNOWLEDGMENTS

The first author thanks to Prof. Fukuda the work on his laboratory at the Nagoya University and the very useful discussions and suggestions given by Prof. Fukuda and his team.

The authors gratefully acknowledge the financial support from CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior).

References