Available online at www.IJournalSE.org

Emerging Science Journal

Vol. 3, No. 3, June, 2019

Position Control of a Flexible Joint via Explicit Model Predictive Control: An Experimental Implementation

Massoud Hemmasian Ettefagh [a], Mahyar Naraghi [a*], Farzad Towhidkhah [b]

[a] *Department of Mechanical Engineering, Amirkabir University of Technology, Tehran, Iran*

[b] *Department of Biomedical Engineering, Amirkabir University of Technology, Tehran, Iran*

Abstract

This paper experimentally controls a flexible joint via explicit model predictive control (Explicit MPC) method. The scheme divides the state space into different partitions, then solves the associated multi parametric optimization in off-line computations. The result stores in a look-up table to be used in on-line algorithm. First, the state space equations of the flexible joint are derived and linearized around the working point. Then, in order to meet the plant's specifications, desired performance and the limitation of processor/memory, the constraints, weights, sampling time and prediction horizon are determined for the system. Finally, the algorithm is applied on the experimental plant. Numerous simulations, the result of the experiment and comparison with other methods confirmed that the method was able to control the vibrations of the constrained flexible joint.

Keywords:

Flexible Joint;
Explicit Model Predictive Control;
Multi-Parametric Optimization;
Discrete Linear Time Invariant System.

Article History:

Received:	10	October	2018
Accepted:	18	May	2019

1- Introduction

One of the important issues of industrial arms is backlash phenomenon in the joints and gearboxes and the associated nonlinear effects. One method of dealing with this problem is to use flexible joints. On the other hand, the flexibility produces some demerits, including the accuracy reduction in joint positioning and unwanted vibrations. To solve this problem and accurately position the robot arm, many control algorithms are proposed [1, 2].

Model predictive control uses a model of the plant to predict the future behavior of it through the prediction horizon time-span. The method employs the model to provide a control law that stabilize the system while the constraints are satisfied. At the beginning of the scheme, it was used exclusively to control stable and slow processes with large sampling times [3, 4]. Thanks to the state space approach and fast constrained optimization algorithms the method extended to different applications. Since then, many researches have done to further decrease the computations. Fast algorithms [5], tailored for the system with respect to its characteristics, parameterization of the input vector to decrease the number of decision variables [6, 7] are two successful of these attempts to decrease the computation times. Explicit model predictive control solves a multi parametric optimization problem and divide the state space into different partitions with respect to the active constraints. In addition to suits the benefits of model predictive control, advantages such as no need for an on-line optimization, easily implementation on industrial controllers with limited memory and processor and high reliability have made Explicit MPC a well-known method in industrial communities as well as in research groups.

This paper used Explicit MPC to experimentally control a flexible joint system. Weights, constraints, prediction horizon and sampling time selection as well as their effect on the system's performance were studied here exhaustively. Step-by-step design procedure and the explanation of the implementation on the hardware device were other objects of

[*] **CONTACT**: Maraghi@aut.ac.ir

DOI: http://dx.doi.org/10.28991/esj-2019-01177

this paper. The remainder of the paper is organized as follows. Section 2 is devoted to the derivation of the governing equations for the model and extracting constraints on the input and its incremental value and the state. In section 3, the theory of Explicit MPC is explained and the controller is designed for the system. Implementation of the designed controller and the evaluation of the performance in simulations and experiments are expressed in section 4. Finally, conclusion and final remarks are expressed in section 5.

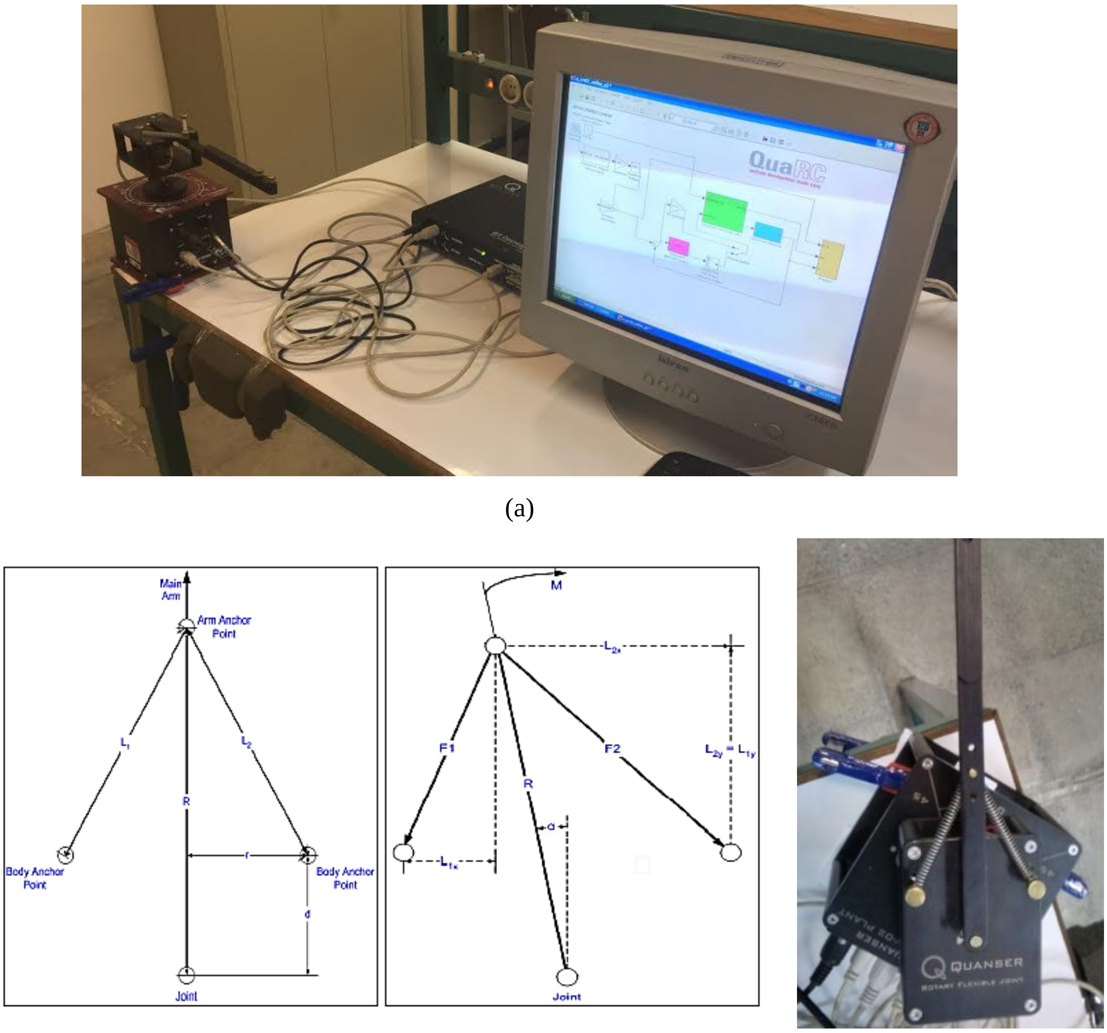
2- Modeling of the Flexible Joint

Because MPC method uses a model to predict the future of the system, one needs to know the dynamics of the system to design the controller, in the first part of this section we drive the system's equations of motion in state space. In the second part, input, rate of input and state constraints are determined for the system. To design a proper stabilizing controller, we have to have a relative knowledge of the values of the constraints because they represent practical restrictions of the system and violating them may lead to instability.

2-1- Dynamical Modeling of the System

Figure 1 depicts the flexible joint system used in this paper. In this section we employ Euler-Lagrange equations to derive equations of motions of the system in state space.

According to Figure 1 the connection between the springs and the base is in such a way that we can substitute that with an equivalent torque. Then, this torque is used to determine the torsional stiffness of the joint. In other words, the applied torque M is $M = K_s(\alpha)\alpha$ where α is the detection angle and $K_s(\alpha)$ denotes the torsional stiffness which is also a function of the deflection angle.



(a)

(c)

(b)

Figure 1. (a) Overview of the experimental setup, (b) the flexible joint and, (c) its schematic.

By assuming linear behavior of the springs, the rotational stiffness coefficient $K_s(\alpha)$ is related to linear stiffness of the springs as is described in Figure 1 (c). The change in springs' length for the deflection angle α are:

$$\begin{aligned} &L_{1x} = r - Rsin(\alpha), \qquad L_{1\,y} = R\cos(\alpha) - d\,, \qquad L_1 = \sqrt{L_{1x}^2 + L_{1y}^2} \\ &L_{2x} = r + Rsin(\alpha), \qquad L_{2\,y} = R\cos(\alpha) - d, \qquad L_2 = \sqrt{L_{2x}^2 + L_{2y}^2} \end{aligned} \tag{1}$$

Which produce the following forces;

$$\begin{aligned} &F_1 = F_{1x}\hat{\imath} + F_{1y}\hat{\jmath} = K_L(L_1 - L) + F_r\,,\ F_{1x} = \frac{F_1 L_{1x}}{L_1}\,, \qquad F_{1y} = \frac{F_1 L_{1y}}{L_1} \\ &F_2 = F_{2x}\hat{\imath} + F_{2y}\hat{\jmath} = K_L(L_2 - L) + F_r\,,\ F_{2x} = \frac{F_2 L_{2x}}{L_2}\,, \qquad F_{2y} = \frac{F_2 L_{2y}}{L_2}. \end{aligned} \tag{2}$$

Where F_r is tension force in springs when $\alpha = 0$, and K_L is the stiffness coefficient of the linear springs. The resultant torque at the joint is;

$$M = R \times \left(F_x\hat{\imath} + F_y\hat{\jmath}\right) = R\cos\alpha\,(F_{2x} - F_{1x}) - R\sin\alpha\,\left(F_{2y} + F_{1y}\right). \tag{3}$$

The left hand side of (3) is equal to $M = K_s(\alpha)\alpha$, so we can readily find the torsional stiffness of the joint. Since (3) is a nonlinear equation, the torsional stiffness is a nonlinear function of α. For small values of α linearization around $\alpha = 0$ gives us

$$K_s = \left.\frac{\partial M}{\partial \alpha}\right|_{\alpha=0} = \frac{2R\left(\left(Rr^2L - dDL + dD^{3/2}\right)k + (Dd - Rr^2)F_r\right)}{D^{3/2}}, D = r^2 - (R - r)^2. \tag{4}$$

The kinetic and potential energy and the equations of motion of the system are given as;

$$\begin{aligned} &U = \frac{1}{2}K_s\alpha^2 \\ &T = \frac{1}{2}J_{hub}\dot{\theta}^2 + \frac{1}{2}J_{arm}\left(\dot{\theta} + \dot{\alpha}\right)^2 \\ &(J_{hub} + J_{arm})\ddot{\theta} + J_{arm}\ddot{\alpha} = \tau - B_{eq}\dot{\theta} \\ &J_{arm}\ddot{\theta} + J_{arm}\ddot{\alpha} + K_s\alpha = 0 \end{aligned} \tag{5}$$

Where θ, α, J_{hub} and J_{arm} denote rotational angle of the hub, rotational angle of the arm, rotational inertia of the hub at the joint and rotational inertia of the arm around the joint respectively. B_{eq} is the viscus damping coefficient between the hub and the arm and τ is the applied torque from the actuator to the system. The relationship between τ and the controller output u is given as [11]:

$$\tau = \frac{\eta_m\eta_g K_t K_g}{R_m}u - \frac{\eta_m\eta_g K_t K_m K_g^2}{R_m}\dot{\theta}. \tag{6}$$

Numerical values of parameters and constants in (5) and (6) are $J_{hub} = 0.0021, J_{arm} = 0.0019,\ K_s = 2.24, B_{eq} = 0.004, K_t = K_m = 0.0077, K_g = 70, R_m = 2.6, \eta_g = 0.9, \eta_m = 0.69$ [11]. The state space vector is defined as $x = \left[x_1 = \theta, x_2 = \alpha, x_3 = \dot{\theta}, x_4 = \dot{\alpha}\right]^T$; hence the linearized equations of motions in state space form are given as.

$$\begin{aligned} &\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\end{bmatrix} = \underbrace{\begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & 982.8149 & -1.7517 & 0\\ 0 & -2163.85 & 1.7518 & 0\end{bmatrix}}_{A}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix} + \underbrace{\begin{bmatrix}0\\ 0\\ 437.9316\\ -437.9316\end{bmatrix}}_{B}u\,, \\ &y = \begin{bmatrix}\theta\\ \alpha\end{bmatrix} = \underbrace{\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\end{bmatrix}}_{C}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix},\ D = 0 \end{aligned} \tag{7}$$

2.2. Modeling of the Constraints

One of the main benefits of MPC is its ability to provide a control law that satisfies constraints. State and output constraints are usually defined for the system's performance, so they are usually considered as soft constraints. Constraints on the input and rate of the input are related to actuators/systems characteristics and violation of them may leads to instability. Therefore, these constraints must be considered as hard constraints. In this section we employ some experiments to extract constraints on input, input rate and states of the system. In order to determine the constraint on the input and rate of the input, we used pole placement method to design a controller. Then, we set the poles of the closed-loop system to the far left of the $j\omega$ axis till the closed-loop response of the system became unstable due to the uncertainties such as friction in gears, and un-modelled dynamics. This type of instability produced chattering behavior of the system under which the control signal was become saturated with the highest possible frequency. The saturation level was around 1.65 $[v]$. Since the controller was working in the highest possible frequency, the constraint on rate could be obtained from $\Delta u = u_k - u_{k-1}$. Figure 2 shows the valuse of constraints on the input and on the rate of the input. The mean value of the maximums of Δu, according to Figure 2 (b), is 0.9 $[v]$.

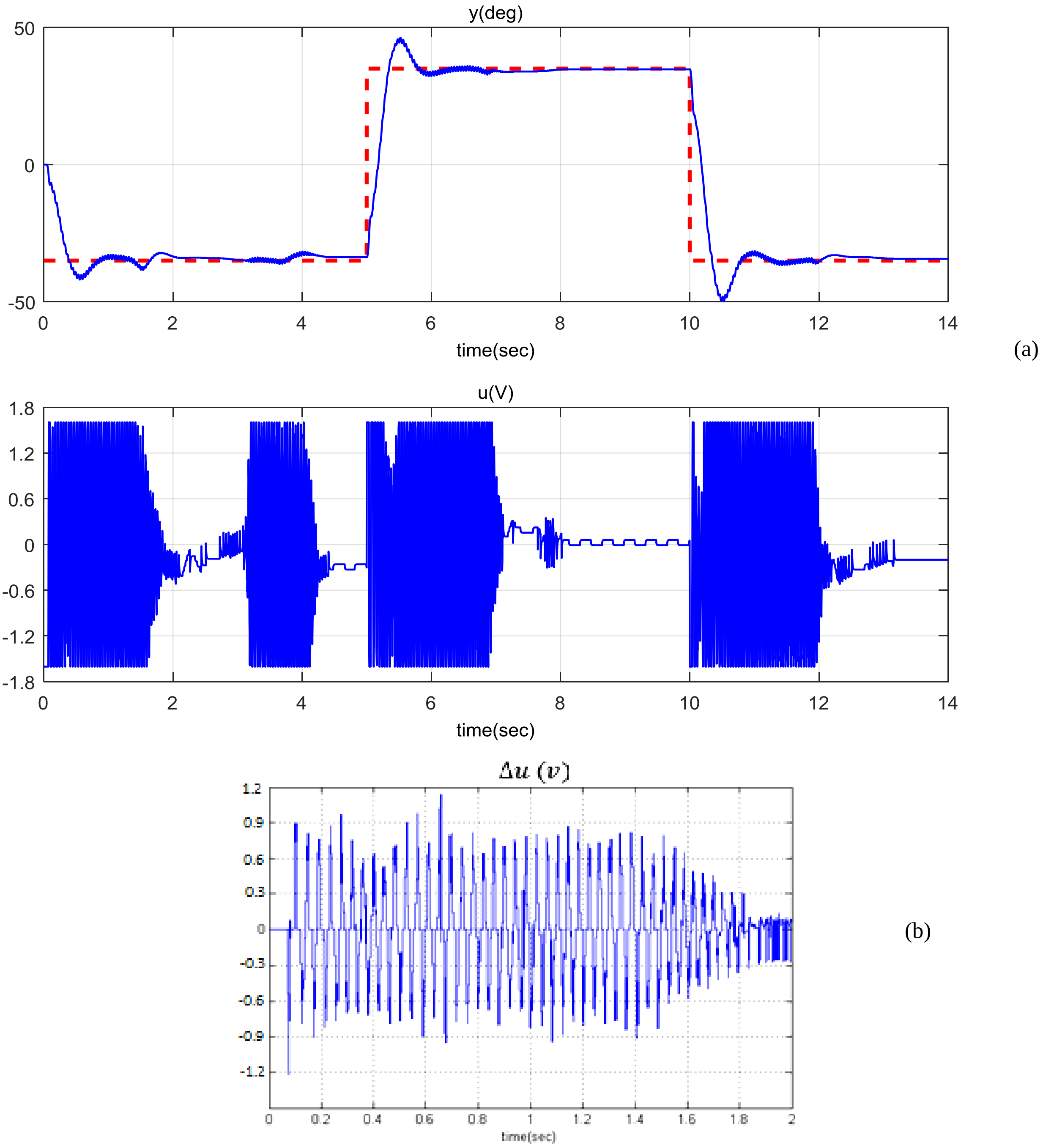


Figure 2. (a) input signal at its peak value, and (b) its rate Δu.

It is worthwhile to note that the required performance and physical restrictions of the system determine the state and output constraints. The required performance includes maximum velocity and error and the allowable value of the overshoot. These values are not very strict and can be modelled as soft constraints. However, there are another group of state/output constraints that deals with physical limitations of the structure, such as restriction of rotation of α in Figure 1, and must be considered as hard constraints. According to our direct measurement and Figure 1, the constraints of α and θ are determined as $\pm 35^0, \pm 50^0$ respectively. Therefore, system's constraints are summarized as;

$$|u| \leq 1.65\ [v], \quad |\Delta u| \leq 1.2\ [v]\ , \quad |\theta| \leq 0.87\ [rad]\ , \quad |\alpha| \leq 0.61\ [rad] \tag{8}$$

3. Designing of the Controller

In this section, we first briefly introduce the theory of Explicit MPC and then we employ the theory to design a controller for the system.

3.1. Explicit Model Predictive Controller

Consider the discrete linear time invariant system

$$\begin{aligned} & x(i+1) = Ax(i) + Bu(i), \qquad y(i) = Cx(i), \\ & y_{\min} \leq y(i) \leq y_{\max}\ , \quad u_{\min} \leq u(i) \leq u_{\max}\ , \end{aligned} \tag{9}$$

For this system, model predictive control solves the following optimization problem with respect to the initial condition $x(0) = x_0$.

$$\begin{aligned} & J^*(x_0) = \min_{u_0, u_1, \cdots, u_{N-1}} x_N^T P x_N + \sum_{i=0}^{N-1} x_i^T Q x_i + u_i^T R u_i \\ & Subject\ to\ \ x_{i+1} = Ax_i + Bu_i, \qquad i = 0,1,\cdots,N \\ & x_0 = x(0), \\ & y_{\min} \leq y(i) \leq y_{\max}\ , \ \ i = 0,1,\cdots,N \\ & u_{\min} \leq u(i) \leq u_{\max}\ , \ \ i = 0,1,\cdots,N-1 \end{aligned} \tag{10}$$

In which N denotes prediction horizon and $\mathrm{Q, P, R}$ are positive-definite matrices denoting weights on states, terminal states and the input. With some calculations, Equation 10 reduces to

$$\begin{aligned} & J^*(x_0) = \min_U U^T H U + x_0^T F U \\ & Subject\ to\ \ GU \leq W + Sx \end{aligned} \tag{11}$$

Where $\mathrm{U} = \left[\mathrm{u}_0^{\mathrm{T}}, \cdots, \mathrm{u}_{\mathrm{N-1}}^{\mathrm{T}}\right]^{\mathrm{T}}$ and matrices $\mathrm{F, H, G, W, S}$ are calculated from Equation 10- see [12] for the details.

The quadratic programming Equation 11 meets KKT conditions [13], so we can solve it as a multi-parametric optimization and reduce the solution to [8];

$$\begin{aligned} & U^*(x_0) = C_{1,j} x_0 + C_{2,j} = -H^{-1}(F x_0 + G_{act}^T \lambda_{act}), \\ & \lambda_{act} = -(G_{act} H^{-1} G_{act}^T)^{-1} (G_{act} H^{-1} S_{act} x_0 + W_{act}) \end{aligned} \tag{12}$$

Where $W_{act}, G_{act}, S_{act}$ are rows of matrices W, G, S in [11] that determine the active constraints with respect to the given initial condition. It is shown in [8] that Equation 12 is a bounded piece-wise continuous function. Therefore, it divides the state space to limited number of partitions where each partition has its own exclusive control law. Next, the partitions and control laws are stored in a look-up table to be used on-line to control the plant [10].

3.2. Controller Design

Here we explain the process of determining the controller's parameters and coefficients. First, we discretize Equation 7 with an appropriate sampling time Δt. Then, we design a series of DLQR controllers with different weights to determine the appropriate matrix weights Q, P, R for the required performance. After selecting the appropriate weights, we compare the results of unconstraints MPC with different N with the result of DLQR to determine the appropriate horizon N. Finally, we impose the constraints on the MPC solution and tune once more the parameters, and then design the Explicit MPC algorithm.

Based on the experimental setup's characteristics we set the sampling time to $\Delta t = 0.002 = 2\,ms$ where is the smallest possible value of the data acquisition board. We start the simulations with $Q = C^T C, R = 1$ and then change Q as $Q = C^T Q^* C$ to reach to the desired performance. Figure 3 shows the simulation result for $Q = C^T C, R = 1$ that has %10 overshoot and 1 sec setteling time. Flexibility of the joint is the main reason of overshoot.

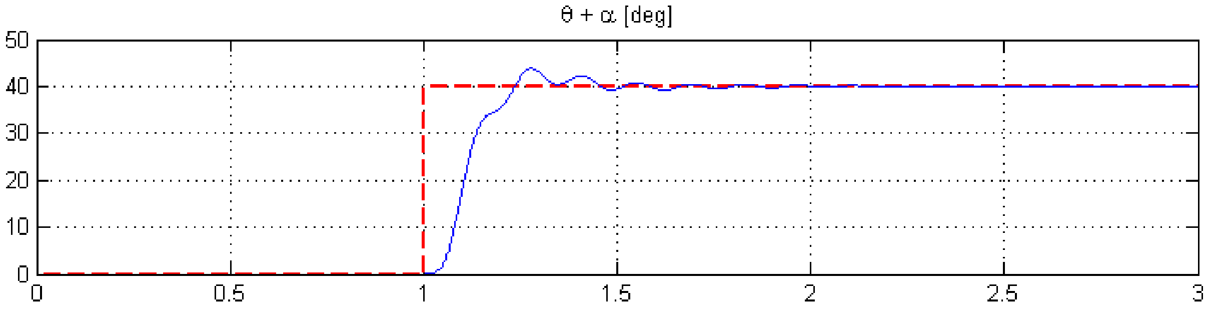


Figure 3. Time response of the closed-loop system with DLQR controller. $R = 1, Q = diag([1, 1, 0, 0])$.

As it is displayed in Figure 4, we can adjust the overshoot by changing the relative value of the elements of Q with eachother. Any increase in the first element, which is respect to θ, decreases the raise time and increases the overshoot, control effort and transient error value of α. Figures 3 and 4 suggest us to choose Q and R as

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad R = 1 \tag{13}$$

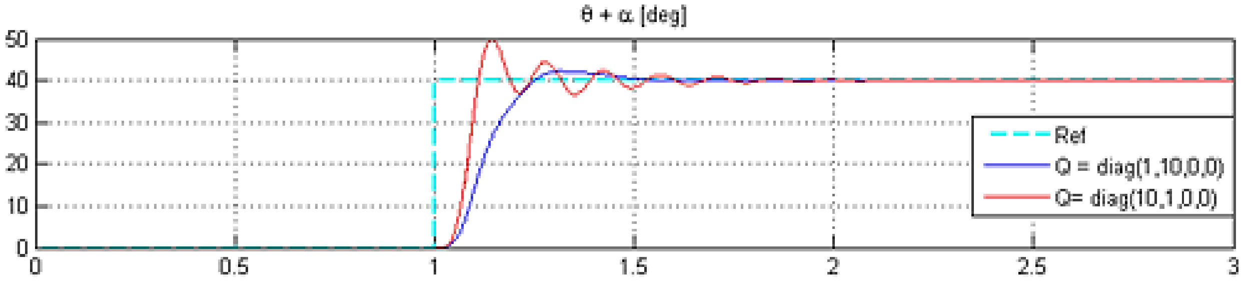


Figure 4. Time response of the closed-loop system with DLQR controller and different weights.

The terminal weight P of Equation 10 is selected as the solution of Algebraic Ricaati Equation to imitate the infinite horizon behavior [14]. The value of P is

$$P = A^T P A + Q - A^T P B (B^T P B + R)^{-1} B^T P A. \tag{14}$$

In order to determine the appropriate N and $\Delta\, t$, we have to keep in mind that as sampling time decreases, the time for on-line calculations also decreases. To prepare the control effort in-time, the on-line calculations must be done during the time span of the sampling time, otherwise the system's stability could not be assured. Therefore, the sampling time should be large enough for on-line calculations to be done. On the other hand, a large sampling time produces delay for the system that deteriorates the stability significantly. Another important parameter is the horizon. A large horizon is desirable in the sense that it increases stability; however, as the horizon increases the on-line computation load increases which means the sampling time has to increase too. To keep a balance between computation load, sampling time and horizon, we have to find the smallest horizon in which closed-loop stability is guaranteed in the simulations.

Figure 5 shows the sensitivity of the closed-loop solution to the prediction horizon. It shows that the quality of response is poor for $N = 1{,}2$ and it improves significantly as N increases (Note the difference between $N = 2$ and $N = 3$). For $N > 10$ the response does not experience a noticeable change. In addition, the horizon does not have any influence on rise time for $N \geq 3$. The horizon in Explicit MPC must be kept as small as possible to prevent time-taking computation loads and large look-up tables [9], so a small N is desirable here.

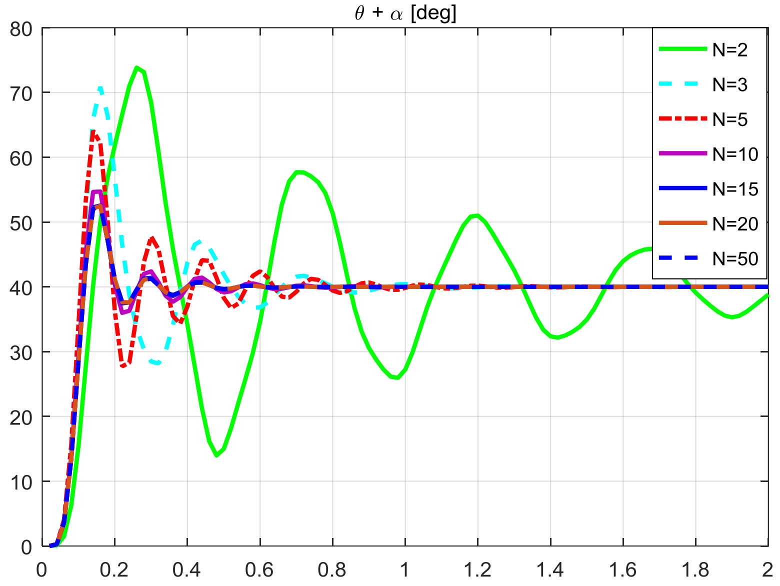


Figure 5. Time response of the unconstrained system with different horizons and $\Delta t = 20\ ms$.

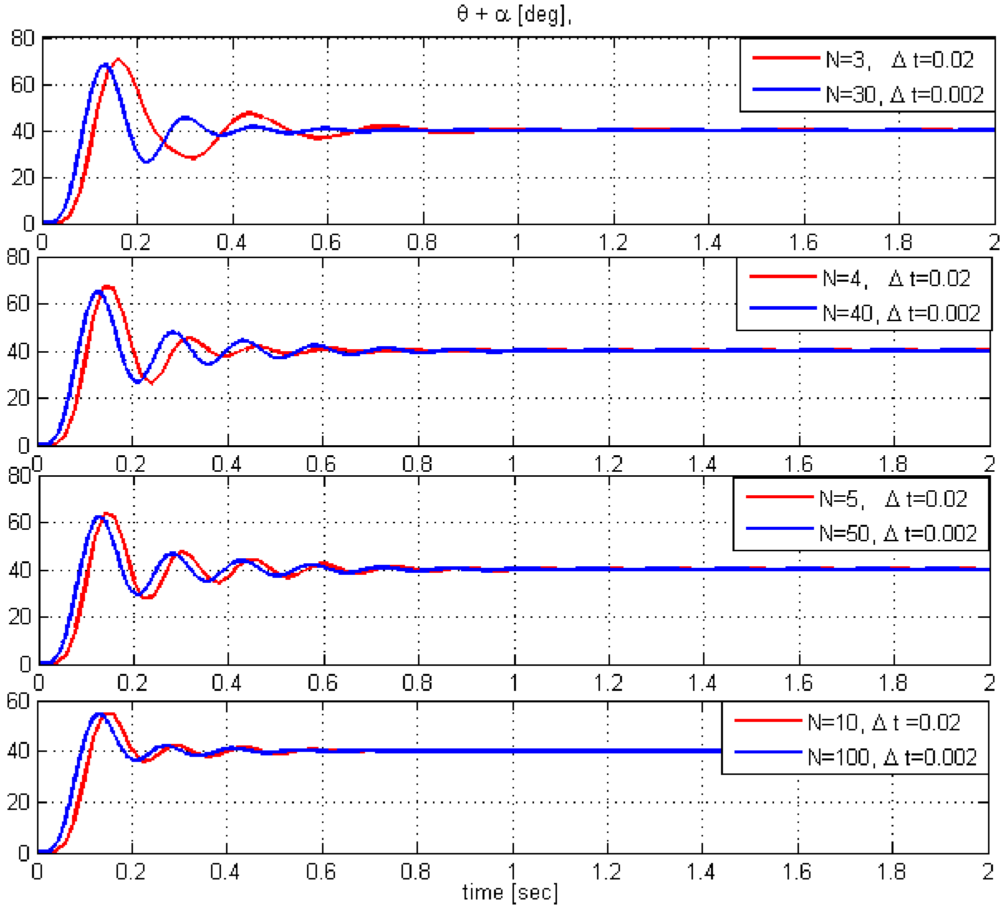


Figure 6. effect of sampling time in system's response.

Figure 6 shows the system's response for two different sampling times. In order to have the same overall prediction horizon in time domain, the prediction horizon in each case is specified in such a way that $N\Delta t$ does not change in each case. It can be seen from the figure that the delay decreases as the sampling time decreases. The next noticeable item in the figure is the change in qualitative behavior of the response for $N = 3, \Delta t = 0.02$ with other cases. The first graph shows that the phase of the red curve is different from the phase of the other curves. This suggests that the value of sampling times is important for small prediction horizons. From figures 4 and 5 we chose $N = 4, 5\ , \Delta t = 20\ ms$ for the Explicit MPC.

The effects of applying constraints on the controller are shown in Figure 7. It depicts that the rise time in MPC is larger than the rise time in DLQR, but the settling time is the same for two controllers. In MPC solution, the selling time is equal to the rise time, while these values are considerably different in DLQR. Moreover, while the solution of DLQR produces overshoot, the MPC removes any overshoot from the closed-loop solution which is significantly important in industrial situations where the overshoot could be damaging.

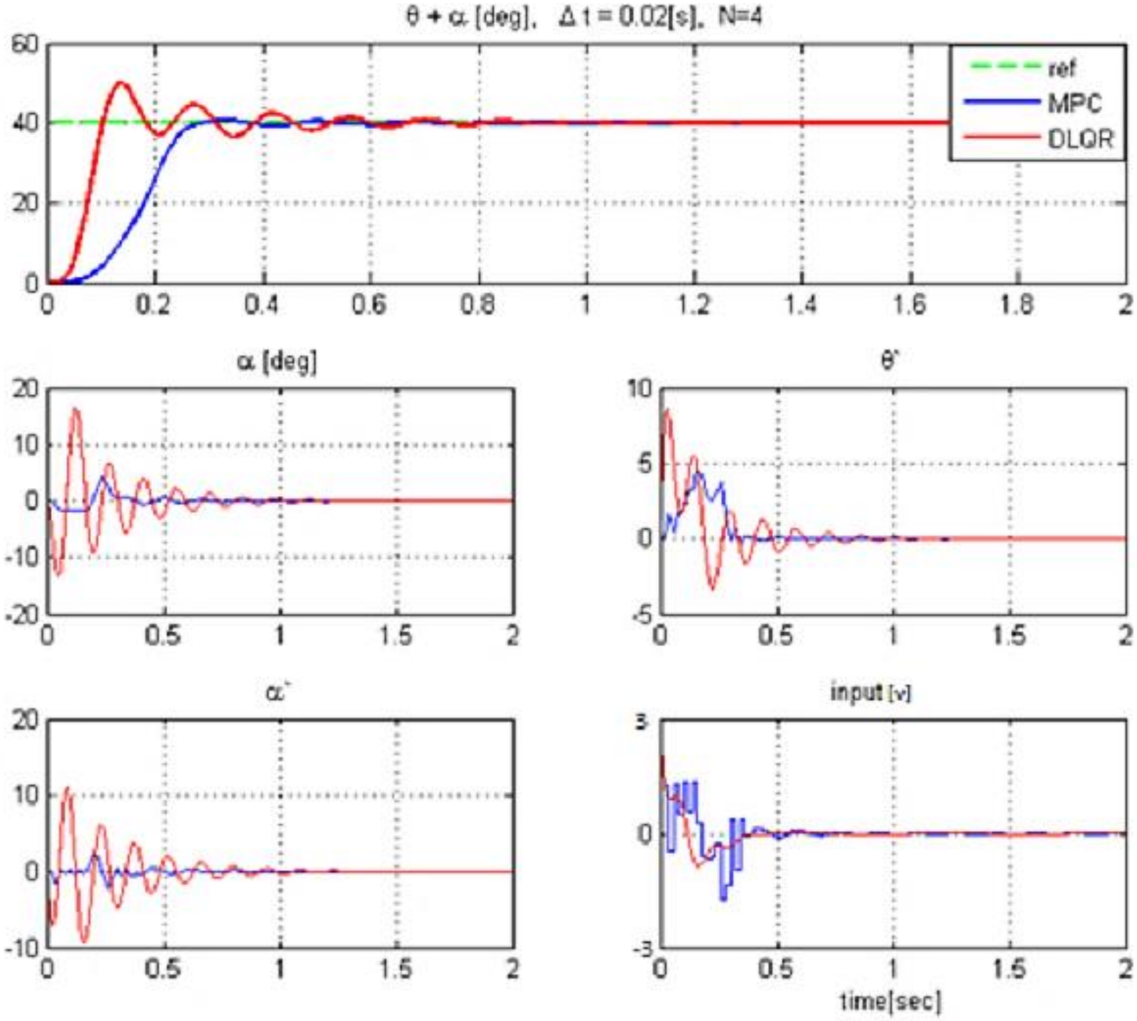


Figure 7. **Closed-loop performance after imposing constraints on input, rate of input and states.**

4. Implementation of the Controller

In designing Explicit MPC, we should keep in mind that the main concern is to keep the on-line computations as small as possible. The on-line algorithm duty is to find the partition in which the measured initial condition is belong to, and then to apply the corresponding control law. Numerous research has shown that the most time-consuming part of the on-line algorithm is the determination of the partition [15, 16, 17]. Therefore, in order to keep the computation time minimum, we must decrease the number of partitions.

As it was mentioned before, the element is to keep the number of partitions as small as possible. We employed Yalmip [18] and mpt-toolbox [19] to extract Explicit MPC algorithm. In the first try, we designed the controller for $N = 5$ that results to 4766 partitions. During the implementation, it was observed that the on-line search algorithm was not able to determine the partition of the given initial condition within the sampling time of 20 $[ms]$. Therefore, we used $N = 4$ for the prediction horizon which results in 1283 partitions. Then, we used the look-up table in Simulink to control the setup in Real Time Workshop. Figure 8 depicts an overview of the Simulink model where the explicit MPC is integrated through an embedded function. The joint rotations are measured with 1024-bit resolution encoders and $G(s) = 275\frac{s}{s+275}$ is employed as the velocity observer.

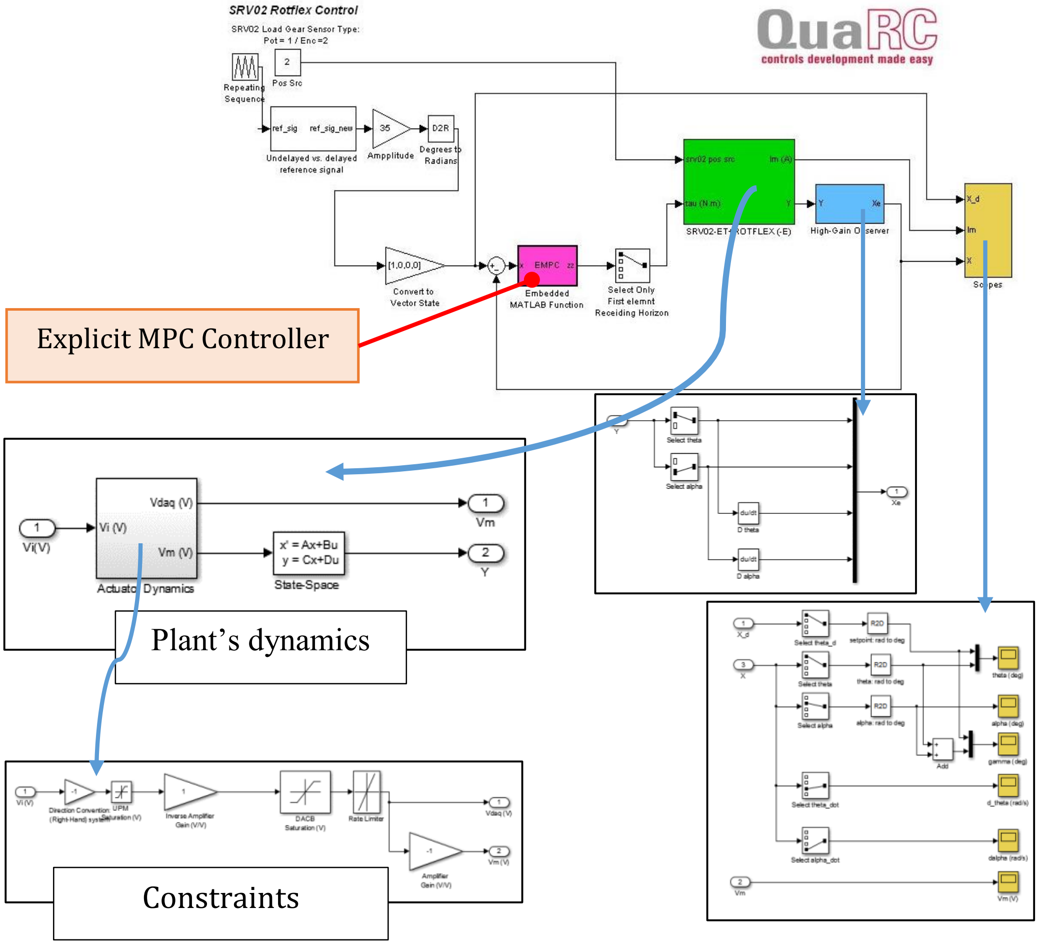


Figure 8. Simulation file for simulations and experiments.

The response of the closed-loop system with explicit MPC controller for step, ramp and sine inputs are shown in Figures 9 to 11. In Figure 9, the constraints are set in such a way that the closed-loop solution is free from any overshoot. The result for step input is compared with the closed-loop result of DLQR controller. Both controller have the same settling time, but the rise time in DLQR is much less than the rise time in explicit MPC. Unlike DLQR, the closed-loop solution of explicit MPC shows no overshoot in the implementation. The discrepancy between the simulation and the implementation is a result of uncertainties, modeling inaccuracies and un-modelled dynamics. Figure 10 and 11 show the closed-loop response of simulation and implementation for ramp and sine inputs. Again, we can see some discrepancy between the simulations and the experiments, but the overall performance of the controller is satisfactory.

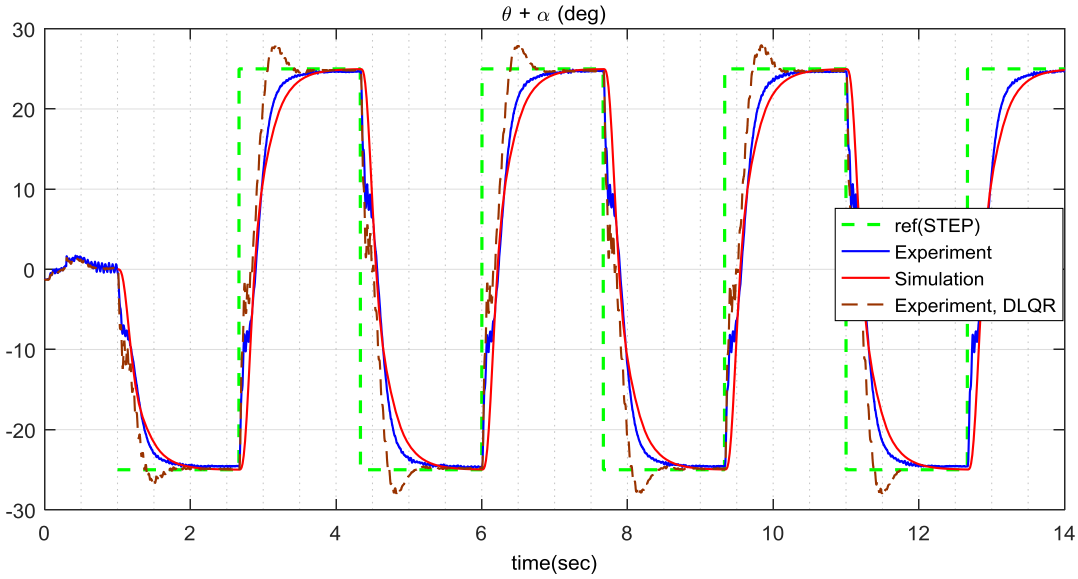


Figure 9. Controller response for step.

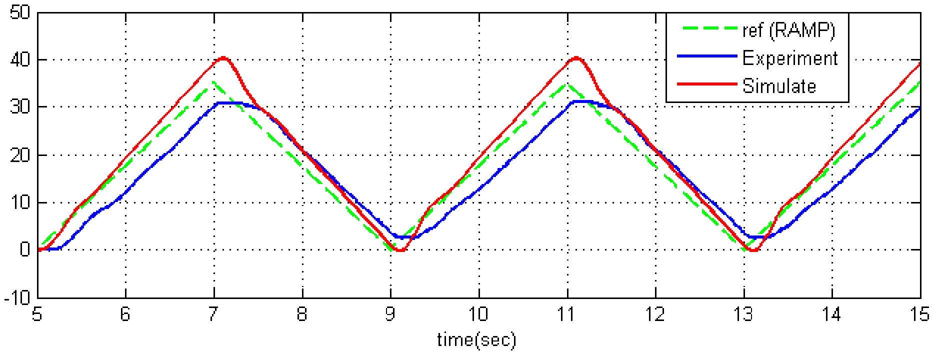


Figure 10. **Controller response for ramp.**

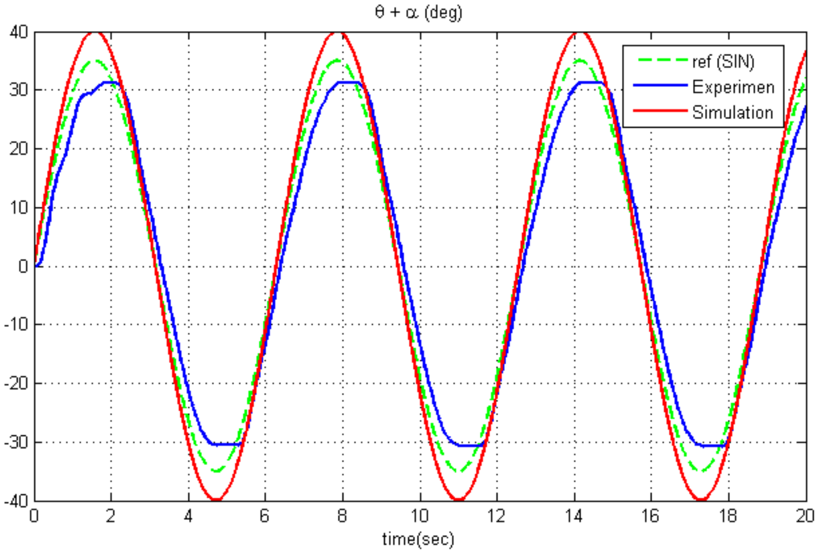


Figure 11. **Controller response for sine.**

5- Conclusion

In this article, the implementation of explicit model predictive control on a flexible joint was studied. The designing process such as weight selection, horizon selection and constraints values was explained step-by-step via a practical approach. The experiments showed the unwanted vibrations of the flexible joint can be reduced significantly with the explicit MPC approach. The ability to accommodate constraints is the main advantage of explicit MPC over DLQR. It was shown that the number of partitions is a major concern in this scheme and the designed should keep the it down as much as it can be.

6- Conflict of Interest

The authors declare no conflict of interest.

7- References

[1] Dwivedy, Santosha Kumar, and Peter Eberhard. “Dynamic Analysis of Flexible Manipulators, a Literature Review.” Mechanism and Machine Theory 41, no. 7 (July 2006): 749–777. doi:10.1016/j.mechmachtheory.2006.01.014.

[2] Shigang, Yue. “Redundant Robot Manipulators with Joint and Link flexibility—I. Dynamic Motion Planning for Minimum End Effector Deformation.” Mechanism and Machine Theory 33, no. 1–2 (January 1998): 103–113. doi:10.1016/s0094-114x(97)00028-1.

[3] Morari, Manfred, and Jay H. Lee. “Model Predictive Control: Past, Present and Future.” Computers & Chemical Engineering 23, no. 4–5 (May 1999): 667–682. doi:10.1016/s0098-1354(98)00301-9.

[4] Qin, S.Joe, and Thomas A. Badgwell. “A Survey of Industrial Model Predictive Control Technology.” Control Engineering Practice 11, no. 7 (July 2003): 733–764. doi:10.1016/s0967-0661(02)00186-7.

[5] Wang, Yang, and Stephen Boyd. “Fast Model Predictive Control Using Online Optimization.” IEEE Transactions on Control Systems Technology 18, no. 2 (March 2010): 267–278. doi:10.1109/tcst.2009.2017934.

[6] Wang, Liuping. “Discrete Model Predictive Controller Design Using Laguerre Functions.” Journal of Process Control 14, no. 2 (March 2004): 131–142. doi:10.1016/s0959-1524(03)00028-3.

[7] Ettefagh, Massoud Hemmasian, Mahyar Naraghi, Farzad Towhidkhah, and Jose De Dona. “Model Predictive Control of Linear Time Varying Systems Using Laguerre Functions.” 2016 Australian Control Conference (AuCC) (November 2016). doi:10.1109/aucc.2016.7868014.

[8] Bemporad, Alberto, Manfred Morari, Vivek Dua, and Efstratios N. Pistikopoulos. “The Explicit Linear Quadratic Regulator for Constrained Systems.” Automatica 38, no. 1 (January 2002): 3–20. doi:10.1016/s0005-1098(01)00174-1.

[9] Hrovat, D., S. Di Cairano, H.E. Tseng, and I.V. Kolmanovsky. “The Development of Model Predictive Control in Automotive Industry: A Survey.” 2012 IEEE International Conference on Control Applications (October 2012). doi:10.1109/cca.2012.6402735.

[10] Alessio, Alessandro, and Alberto Bemporad. “A Survey on Explicit Model Predictive Control.” Lecture Notes in Control and Information Sciences (2009): 345–369. doi:10.1007/978-3-642-01094-1_29..

[11] Quanser Inc. (2011). SRV02 rotary servo base unit. User Manual.

[12] Ettefagh, Massoud Hemmasian, José De Doná, Mahyar Naraghi, and Farzad Towhidkhah. “Control of Constrained Linear-Time Varying Systems via Kautz Parametrization of Model Predictive Control Scheme.” Emerging Science Journal 1, no. 2 (September 19, 2017): 65. doi:10.28991/esj-2017-01117.

[13] Bazaraa, M. S., Sherali, H. D., & Shetty, C. M., Nonlinear programming: theory and algorithms. (2013), John Wiley & Sons.

[14] Mayne, D.Q., J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. “Constrained Model Predictive Control: Stability and Optimality.” Automatica 36, no. 6 (June 2000): 789–814. doi:10.1016/s0005-1098(99)00214-9.

[15] Baotić, Mato, Francesco Borrelli, Alberto Bemporad, and Manfred Morari. “Efficient On-Line Computation of Constrained Optimal Control.” SIAM Journal on Control and Optimization 47, no. 5 (January 2008): 2470–2489. doi:10.1137/060659314.

[16] Geyer, Tobias, Fabio D. Torrisi, and Manfred Morari. “Optimal Complexity Reduction of Polyhedral Piecewise Affine Systems.” Automatica 44, no. 7 (July 2008): 1728–1740. doi:10.1016/j.automatica.2007.11.027.

[17] Bemporad, A., and C. Filippi. “Suboptimal Explicit Receding Horizon Control via Approximate Multiparametric Quadratic Programming.” Journal of Optimization Theory and Applications 117, no. 1 (April 2003): 9–38. doi:10.1023/a:1023696221899.

[18] Lofberg, J. “YALMIP : a Toolbox for Modeling and Optimization in MATLAB.” 2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508) (2004). doi:10.1109/cacsd.2004.1393890.

[19] Kvasnica, Michal, Pascal Grieder, Mato Baotić, and Manfred Morari. “Multi-Parametric Toolbox (MPT).” Lecture Notes in Computer Science (2004): 448–462. doi:10.1007/978-3-540-24743-2_30.