Innovative Control Techniques for Mechatronic Systems

P. Boscariol[1], A. Gasparetto, M. Giovagnoni, A. Lanzutti, R. Vidoni, V. Zanotto
Dipartimento di Ingegneria Elettrica, Gestionale e Meccanica, Università degli Studi di Udine, Italy
E-mail: [1]paolo.boscariol@uniud.it

Keywords: Hardware-In-the-Loop, Model Predictive Control, Vibration.

An increasing number of robot-based application require the use of manipulators with small weight and limited inertia. Unfortunately these devices are also prone to undesirable vibration effects, which are due to the flexibility of their links. In order to reduce the unwanted vibrations in mechanisms retaining their high-speed motion particular control techniques must be employed [1, 2]. For this reason in this paper an innovative controller for flexible-links mechanisms based on MPC (Model Predictive Control) with constraints is proposed [3, 4]. So far this kind of controller has been employed almost exclusively for controlling slow processes, like chemical plants, but the authors' aim is to show that this approach can be successfully adapted to plants whose dynamical behavior is both nonlinear and fast changing [5, 6]. The effectiveness of this control system will be compared to the performance obtained with a standard control strategy employed in industrial applications. The reference mechanism chosen to evaluate the performance of this control strategy is a single-link planar mechanism laying on the vertical plane driven by a torque-controlled electric actuator.

The control strategies for vibration reduction are quite hard to test, since flexible-links mechanisms are quite prone to mechanical failure. In case of a non well-done tuning of the control system the links are exposed to strong strains, especially when dealing with closed-loop structures. During these tests a frequent replacement of the mechanism links is required, and potential safety risks are encountered. A good solution to those problems can be found in the use of Hardware-In-the-Loop-Simulation (HILS) technology [7, 8]. This emerging technology, which is used mainly to design and test control systems, is based upon the interaction between a real hardware and a virtual system (i.e. a simulation based mathematical model) that emulates and physically replace a real system or one of his components. The main advantage of this approach is that it allows the use of a virtual model directly inside of the control loop: in this way a fine and accurate tuning of the control system parameters can be provided without involving the flexible-link mechanism. For this purpose a simulator named FLiMHILS (Flexible Link Mechanisms HIL Simulator) has been developed [9], and will be employed in this paper to test and tune the innovative constrained MPC controller. The results show the optimal performances of control system and the capability of the HIL approach.

1 INTRODUCTION

The need of accurate models for flexible link manipulators and their elastic behavior is a field that have attracted a great interest in researchers. This is due to the fact that the ability to model with accuracy and to control the vibrational phenomena in mechanisms can be directly translated into the development of robots with lighters arms and with an higher ratio between their maximum load and their overall weight. Smaller arms means also a reduction of their inertia value, with a positive influence on the operative speed of manipulators. In the wake of this possibilities, in the past 3 decades a lot of papers and book have been written to propose and investigate both innovative mathematical models and control strategies. The most common approach to flexible links modeling is based on the Finite Element Method (FEM), as can be seen in [10].

In this paper we will use the model introduced by Giovagnoni in [11], whose accuracy has been demonstrated several times. It is based on the equivalent rigid-link system (ERLS) theory and on the assumption that the flexible motion of a body cannot be separated from its rigid motion without degradating the overall accuracy of the model. The control system of choice for this paper is a Model-Based Predicitive Controller (MPC), which will be employed to control the vibration in a single-link flexible mechanism. The design of this control system will be based on the linearization procedure proposed by Gasparetto in [12], from which a state-space model of the mechanism can be obtained.

The experimental tests of control strategies for vibration reduction in flexible link mechanism poses some practical order problems. Flexible link mechanisms are quite prone to mechanical failures, which are encountered when the links are subject to strong strains as consequences of an improper control strategy. This is especially true when dealing with closed-loop mechanisms. This represents also a potential safety risk for the operator. One solution to these problems can be found in Hardware-In-the-Loop (HIL) tests. This technology allows the complete and accurate interaction of a real device with a simulated one. In this case we can develop a software that represents a virtual model of the dynamics of a flexible link mechanism, make it run on a PC-based device, and through an interface board we can establish an interaction with a real control system. With this strategy, we can run all the tests required for the tuning of the control system parameters without involving the fragile mechanism prototype. Other prerogatives of the HIL approach include:

- reproducibility of experiments
- the ability to perform test which would otherwise be impossible, impractical on unsafe
- shorter time required for experimental testing
- testing the effects of components faults
- long-term durability testing

One requirement of the dynamic model employed for HIL is its real-time capability, since we need to make it interact with real-world signals, as the input and outputs of the control system employed in the feedback loop. This is a problem of not easy solution, since the dynamic model used is both non linear and high order, i.e. it involves large and badly conditioned matrices whose computation requires a large amount of resources. In this paper we will first explain the dynamic model of flexible link mechanisms employed, then we will show the strategies used for its real-time implementation on which the simulated model for HIL is based. Then we will concentrate on the implementation of a predictive control strategy (MPC) used together with a state observer. This control strategy will be tested first in a full simulation environment, then a real control system will be tested on both the mechanism simulated with HIL and the real mechanism prototype. In this way we will provide a benchmark of both the accuracy of the HIL system and the effectiveness of the MPC control for active vibration reduction.

2 Dynamic model of planar flexible-links mechanisms

In this section the dynamic model of a flexible-link mechanism proposed by Giovagnoni [11] will be briefly explained. The choice of this formulation among the several proposed in the last 30 years has been motivated mainly by the high grade of accuracy provided by this model, which has been proved several times. Each flexible link belonging to the mechanism is divided into finite elements. Referring to Figure 1 the following vectors, calculated in the global reference frame $\{X, Y, Z\}$, can be defined:

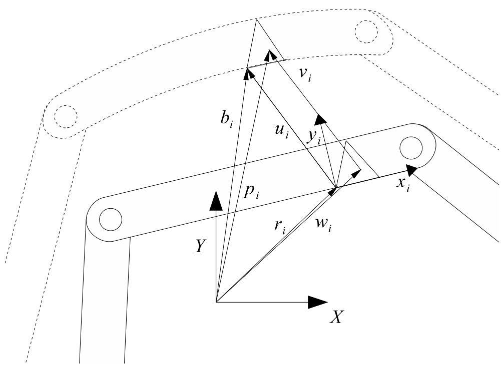


Figure 1: Kinematic definitions

- $\mathbf{r}_i$ and $\mathbf{u}_i$ are the vectors of nodal position and nodal displacement in the ith element of the ERLS and of their elastic displacement
- $\mathbf{p}_i$ is the position of a generic point inside the ith element
- $\mathbf{q}$ is the vector of generalized coordinates of the ERLS

The vectors defined so far are calculated in the global reference frame $\{X, Y, Z\}$. Applying the principle of virtual work, the following relation can be stated:

$$\begin{aligned}&\sum_i \int_{V_i} \delta \mathbf{p}_i{}^T \ddot{\mathbf{p}}_i \rho_i d\nu + \sum_i \int_{V_i} \delta \epsilon_i{}^T \mathbf{D}_i \epsilon_i d\nu \\ &= \sum_i \int_{V_i} \delta \mathbf{p}_i{}^T \mathbf{g} \rho d\nu + (\delta \mathbf{u}^T + \delta \mathbf{r}^T)\mathbf{F}\end{aligned} \tag{1}$$

ϵ_i, $\mathbf{D}_i$, ρ_i and $\delta\epsilon_i$ are, respectively, the strain vector, the stress-strain matrix, the mass density and the virtual strains of the ith link. $\mathbf{F}$ is the vector of the external forces, including the gravity, whose acceleration vector is $\mathbf{g}$. Eq. 1 shows the virtual works of, respectively, inertia, elastic an external forces. From this equation, $\delta\mathbf{p}_i$ and $\ddot{\mathbf{p}}_i$ for a generic point in the ith element are:

$$\begin{aligned}\delta \mathbf{p}_i &= \mathbf{R}_i \mathbf{N}_i \mathbf{T}_i \delta \mathbf{r}_i \\ \ddot{\mathbf{p}}_i &= \mathbf{R}_i \mathbf{N}_i \mathbf{T}_i + 2(\dot{\mathbf{R}}_i \mathbf{N}_i \mathbf{T}_i + \mathbf{R}_i \mathbf{N}_i \dot{\mathbf{T}}_i)\dot{\mathbf{u}}_i\end{aligned} \tag{2}$$

where $\mathbf{T}_i$ is a matrix that describes the transformation from global-to-local reference frame of the ith element, $\mathbf{R}_i$ is the local-to-global rotation matrix and $\mathbf{N}_i$ is the shape function matrix. Taking $\mathbf{B}_i(x_i, y_i, z_i)$ as the strain-displacement matrix, the following relation holds:

$$\delta \epsilon_i = \mathbf{B}_i \delta \mathbf{T}_i \mathbf{u}_i + \mathbf{B}_i \mathbf{T}_i \delta \mathbf{u}_i \tag{3}$$

Since nodal elastic virtual displacements ($\delta\mathbf{u}$) and nodal virtual displacements of the ERLS ($\delta\mathbf{r}$) are independent from each other the resulting equation describing the motion of the system is:

$$\begin{bmatrix} \mathbf{M} & \mathbf{MS} \\ \mathbf{S}^T\mathbf{M} & \mathbf{S}^T\mathbf{MS} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{u}} \\ \ddot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{S}^T\mathbf{f} \end{bmatrix} \tag{4}$$

M is the mass matrix of the whole system and **S** is the sensitivity matrix for all the nodes. Vector $\mathbf{f} = \mathbf{f}(\mathbf{u}, \dot{\mathbf{u}}, \mathbf{q}, \dot{\mathbf{q}})$ takes account of all the forces affecting the system, including the gravity force. Adding a Rayleigh damping, the right-hand side of Eq. 4 becomes:

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{S}^T \end{bmatrix} = \begin{bmatrix} -2\mathbf{M}_G - \alpha\mathbf{M} - \beta\mathbf{K} & -\mathbf{M}\dot{\mathbf{S}} & -\mathbf{K} \\ \mathbf{S}^T(-2\mathbf{M}_G - \alpha\mathbf{M}) & -\mathbf{S}^T\mathbf{M}\dot{\mathbf{S}} & 0 \end{bmatrix} \begin{bmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{q}} \\ \mathbf{u} \end{bmatrix} + \begin{bmatrix} \mathbf{M} & \mathbf{I} \\ \mathbf{S}^T\mathbf{M} & \mathbf{S}^T \end{bmatrix} \begin{bmatrix} \mathbf{g} \\ \mathbf{F} \end{bmatrix} \tag{5}$$

Matrix $\mathbf{M}_G$ accounts for the Coriolis contribution, while **K** is the stiffness matrix of the whole system. α and β are the two Rayleigh damping coefficients. The system in (4) and (5) can be made solvable by forcing to zero as many elastic displacement as the generalized coordinates, in this way ERLS position is defined univocally [11]. Finally, after removing the displacement forced to zero from (4) and (5) one obtains:

$$\begin{bmatrix} \mathbf{M}_{in} & (\mathbf{MS})_{in} \\ (\mathbf{S}^T\mathbf{M})_{in} & \mathbf{S}^T\mathbf{MS} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{u}}_{in} \\ \ddot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{in} \\ \mathbf{S}^T f \end{bmatrix} \tag{6}$$

3 HIL model realization

The purpose of the Hardware-In-the-Loop simulator is to achieve an interaction between a real implementation of a closed-loop control system and a simulated plant. A PXI system has been chosen as the hardware platform used for real-time simulation of the whole system, including sensors and actuators drivers. It integrates a standard PC-based CPU with high performance I/O board, so it's well suited for both control and measurement application. The model used for HIL requires to accomplish 2 targets: (a) *high accuracy* (b) *Real-Time capability*.

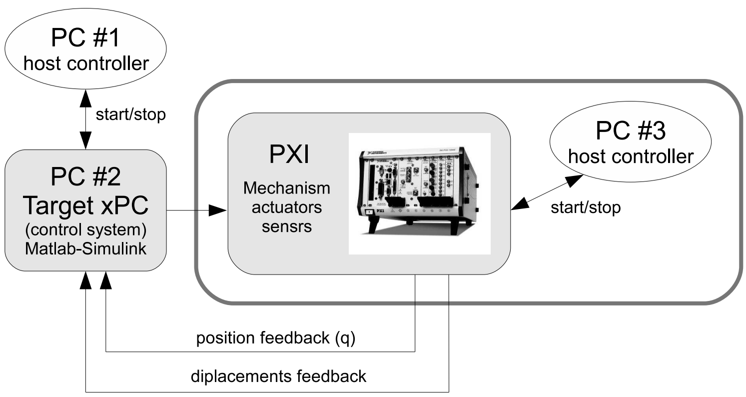


Figure 2: Experimental tests: HIL approach

The need of a deterministic system arises from the use of both real and simulated hardware. So, the model running on the PXI platform has to respect a time constraint condition, or in other words, should have a constant refresh frequency. From eq. (6), which can be rewritten as:

$$\mathbf{M}(\mathbf{x}, t)\dot{\mathbf{x}} = f(\mathbf{x}, t, \mathbf{u}) \tag{7}$$

we can see that it involves a large, non linear and time dependent matrix $\mathbf{M}(\mathbf{x}, t)$. The calculation of the update vector $\dot{\mathbf{x}}$ in this case requires the numerical inversion of such matrix, so the resulting model cannot be run fast enough for Real-Time execution on a standard PC. In order to speed up the time required for the calculus of $\dot{\mathbf{x}}$ at each step we need to make this vector explicit using:

$$d\mathbf{x} = \mathbf{M}^{-1}(\mathbf{x}, t)f(\mathbf{x}, t, \mathbf{u}) \tag{8}$$

An optimized C-code Matlab routine implementation of Eq. (8) has been used for developing real-time capable (or even faster than real-rime) simulations. The speed-up advantage is due to the lack of online power-hungry operations such as matrix inversion, since the calculus of $\mathbf{M}^{-1}(\mathbf{x}, t)$ can be operated off-line. The main drawback of this approach is that a large amount of CPU speed and memory allocation is required for the symbolic computation of the inverse of $\mathbf{M}(\mathbf{x}, t)$ matrix.

The C-code version of Eq. (8) has been used as the basis for an executable .dll file obtained through the use of Visual Studio .NET C compiler. The executable file can be included in a Labview VI that can be deployed on the PXI, where it can run on a real-time OS.

3.1 Reference mechanism

The mechanism chosen as the reference model is a single-link flexible mechanism. It is composed by a square-section metal rod actuated by a brushless motor, so it can swing along the vertical plane. The beam can be modeled as a single dof mechanism, since its position depends only on the angular position q.

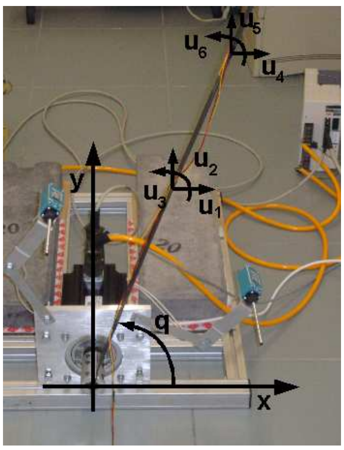


Figure 3: The flexible-link mechanism used

The flexible bar has been modeled using a two finite elements: the total number of elastic degrees of freedom is 9, but 2 of them must be forced to zero in order to take account of the hinge on the

Table 1: Kinematic an dynamic characteristics of the flexible rod

	symbol	value
Joung's modulus	E	$210 \cdot 10^9$ [Pa]
Flexural stiffness	EJ	22.03 [Nm^4]
Beam width	a	$6 \cdot 10^{-3}$ [m]
Beam thickness	b	$6 \cdot 10^{-3}$ [m]
Mass/unit length	m	0.28 [kg/m]
Total length	L	0.7 [m]
Strain sensor position	s	0.35 [m]
First Rayleigh damping constant	α	$8.7 \cdot 10^{-2}$ [s^{-1}]
Second Rayleigh damping constant	β	$2.1 \cdot 10^{-5}$ [s^{-1}]

first node (so both vertical and horizontal displacement must be zero). Then we have to force to zero another one elastic dof to produce a valid ERLS model. We chosed to set to zero the angular displacement u_z of the first node, in this way the resulting model is a double cantilevered beam. The resulting vector of nodal displacement is:

$$\mathbf{u} = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \end{bmatrix}' \tag{9}$$

in this way, the state vector **x** in eq. (5) has 14 components, and the size of the matrix that needs to be inverted in eq. (6) will be 14×14.

4 Model Predictive Control with constraints

In this section the equations leading to the constrained MPC system employed will be briefly analyzed. Basically, MPC control law is calculated as an optimization problem, whose evolution is influenced by both the plant actual input/outputs and its estimated future behavior. In this section a very brief explanation of those concepts is given, more details can be found in [3].

4.1 Model prediction

Given a plant model in state-space form:

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{F}\mathbf{x}(k) + \mathbf{G}\mathbf{u}(k) \\ \mathbf{y}(k) = \mathbf{H}\mathbf{x}(k) \end{cases} \tag{10}$$

where $\mathbf{x}(k)$ is the state vector, $\mathbf{u}(k)$ and $\mathbf{y}(k)$ are the input and output vectors, respectively. Assuming that the whole state $\mathbf{x}(k)$ is measured, the future behavior of the plant at time k over H_p steps, indicated by $[\hat{\mathbf{x}}(k+1|k), \ldots, \hat{\mathbf{x}}(k+H_p|k)]$, can be evaluated as:

$$\begin{array}{l} \hat{\mathbf{x}}(k+1|k) = \mathbf{F}\mathbf{x}(k) + \mathbf{G}\hat{\mathbf{u}}(k|k) \\ \hat{\mathbf{x}}(k+2|k) = \mathbf{F}\hat{\mathbf{x}}(k+1|k) + \mathbf{G}\hat{\mathbf{u}}(k+1|k) \\ \vdots \\ \hat{\mathbf{x}}(k+H_p|k) = \mathbf{F}\hat{\mathbf{x}}(k+H_p-1|k) + \mathbf{G}\hat{\mathbf{u}}(k+H_p-1|k) = \\ = \mathbf{F}^{H_p}\mathbf{x}(k) + \mathbf{F}^{H_p-1}\mathbf{G}\hat{\mathbf{u}}(k|k) + \ldots + \mathbf{G}\hat{\mathbf{u}}(k+H_p-1|k) \end{array} \tag{11}$$

Prediction values of outputs are calculated from predicted states:

$$\hat{\mathbf{y}}(k+n|k) = \mathbf{H}\hat{\mathbf{x}}(k+n|k); \qquad n = 1, 2, \ldots, H_p \tag{12}$$

4.2 Constrained optimization solution

We suppose to have constraints on both control and controlled variables ($u_i(k)$ and z_i(k)) respectively), and on their change rate ($\Delta u_i(k)$), in terms of linear inequalities, such as:

$$u_{imin} \leq u_i(k) \leq u_{imax} \tag{13}$$

$$\Delta u_{imin} \leq \Delta u_i(k) \leq \Delta u_{imax} \tag{14}$$

$$z_{imin} \leq z_i(k) \leq z_{imax} \tag{15}$$

The sequence of predicted output over the prediction horizon, $\mathcal{Z}(k)$ can be calculated in the same form used in Eq.(11) for the state vector:

$$\mathcal{Z}(k) = \Psi\hat{\mathbf{x}}(k|k) + \Upsilon\mathbf{u}(k-1) + \Theta\Delta\mathcal{U}(k) \tag{16}$$

So the minimization problem can be formulated as:

$$\min_{\Delta\mathcal{U}(k)} \Delta\mathcal{U}(k)^T\mathcal{H}\Delta\mathcal{U}(k) - \mathcal{G}^T\Delta\mathcal{U}(k) \tag{17}$$

subject to constraints (13-15). This minimization problem is a standard QP (quadratic programming) problem, since it is in the form: $\min_{\theta} \frac{1}{2}\theta^T\Phi\theta + \phi^T\theta$ with $\Omega\theta \leq \omega$. Moreover, this problem is convex (see [3]), so it can be solved quite easily. Some of the equations shown above contain the state vector **x**, but in practical applications it is impossible to measure all the 6 nodal displacements (and their time derivatives) belonging to the state vector. Hence the need of the state observer to obtain an estimation of the full state vector from a subset of it. Here a standard Kalman asymptotic estimator has been used. Matrix **L** is chosen in order to minimize the mean square error between the estimated and the actual values of the state variable. Being the problem fully observable, **L** is calculated as:

$$\mathbf{L} = \mathbf{P}_k\mathbf{H}^T{\mathbf{U}_k}^{-1} \tag{18}$$

where $\mathbf{P}_k$ is the solution of the Riccati equation: $\mathbf{E}\mathbf{P}_k + \mathbf{P}_k\mathbf{E}^T - \mathbf{P}_k\mathbf{H}^T{\mathbf{U}_k}^{-1}\mathbf{H}\mathbf{P}_k + \mathbf{Q}_k = 0$, where $\mathbf{U}_k$ and $\mathbf{Q}_k$ are the measurement and process noise covariance matrices.

5 Model linearization

In order to compute a linear MPC control, we need to find a linearized version of the dynamic model presented in eq. (6). The whole procedure, together with the proof of its accuracy, has been presented in [12]. Eq. (5) can be written in a more compact form:

$$\hat{\mathbf{A}}\begin{bmatrix} \ddot{\mathbf{u}} \\ \ddot{\mathbf{q}} \end{bmatrix} = \hat{\mathbf{B}}\begin{bmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{q}} \\ \mathbf{u} \end{bmatrix} + \begin{bmatrix} \mathbf{I} \\ \mathbf{S}^T \end{bmatrix}\mathbf{f}_g + \hat{\mathbf{C}}\tau \tag{19}$$

In Eq. (5) the Coriolis contributes $\mathbf{M}_{G1}$ and $\mathbf{M}_{G2}$ have been included in $\mathbf{M}_G$, $\mathbf{f}_g$ represents the the vector of the gravity forces, and τ is the vector of torques provided by the actuators, while $\mathbf{M}_\tau$ is a matrix composed of only zeros and ones representing the relation between the applied torques, the nodal displacements and the free coordinate vector **q**. Looking at Eq. (19), the simplest choice for the state vector of the system is:

$$\mathbf{x} = [\dot{\mathbf{x}}, \dot{\mathbf{q}}, \mathbf{u}, \mathbf{q}]^T \tag{20}$$

so the linearized state-space form of the dynamic model in (6) can be written as:

$$\mathbf{A}_{lin}\dot{\mathbf{x}} = \mathbf{B}_{lin}\ \mathbf{x} + \mathbf{C}_{lin}\tau \tag{21}$$

Following the algebric steps reported in details in [12], the expressions of the linearized version of matrices $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ are:

$$\mathbf{A}_{lin} = \begin{bmatrix} \mathbf{M} & \mathbf{MS} & \mathbf{0} & \mathbf{0} \\ \mathbf{S}^T\mathbf{M} & \mathbf{S}^T\mathbf{MS} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{22}$$

$$\mathbf{B}_{lin} = \begin{bmatrix} -2\mathbf{M}_G - \alpha\mathbf{M} - \beta\mathbf{K} & \mathbf{0} & -\mathbf{K} & -\left.\frac{d\mathbf{K}}{d\mathbf{q}}\right|_{\mathbf{q}=\mathbf{q}_e} \cdot \mathbf{u}_e + \left.\frac{d\mathbf{f}_g}{d\mathbf{q}}\right| \\ \mathbf{S}^T - 2\mathbf{M}_G - \alpha\mathbf{M} - \beta\mathbf{K} & \mathbf{0} & \mathbf{0} & \left.\frac{d(\mathbf{S}^T\mathbf{f}_g)}{d\mathbf{q}}\right|_{\mathbf{q}=\mathbf{q}_e} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{23}$$

$\mathbf{C}_{lin}$ remains unchanged after the linearization process, since it is composed of only zeros and ones. The standard form of the state-space system can be easily found from $\mathbf{A}_{lin}$, $\mathbf{B}_{lin}$ and $\mathbf{C}_{lin}$:

$$\Delta\dot{\mathbf{x}} = \mathbf{F}_{lin}\Delta\mathbf{x} + \mathbf{G}_{lin}\Delta\mathbf{v} \tag{24}$$

where:

$$\mathbf{F}_{lin} = \mathbf{A}_{lin}{}^{-1}\mathbf{B}_{lin} \tag{25}$$

$$\mathbf{G}_{lin} = \mathbf{A}_{lin}{}^{-1}\mathbf{C}_{lin} \tag{26}$$

6 Results of the Model Predictive Controller

Here the results obtained in simulation employing a PID position control and an MPC simultaneous control of both vibration and angular position of the mechanism are presented. This controller acts as an MISO (Multiple-Input, Single-Output) system: it relies on the knowledge of the instantaneous values of displacements u_2 and link angular position q. u_2 and q are the two controlled variables, while the torque applied to the mechanism acts as the control variable. So the tuning of the MPC depends on 5 variables: weight w_2 on u_2 , weight w_q on q , sampling time T_s, prediction horizon H_p and control horizon H_c. Then the constraints on both control and controlled variables should be taken into account. Here the following inequality constraints have been used:

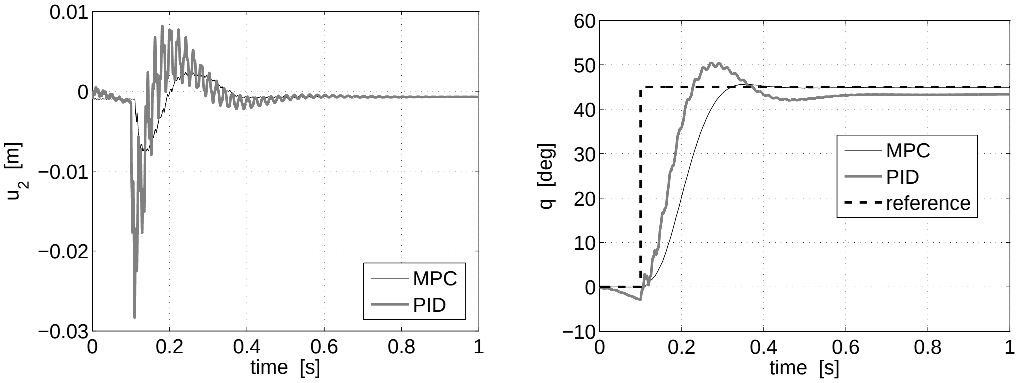
$$u_{2_{min}} \leq u_2 \leq u_{2_{min}} \quad q_{min} \leq q \leq q_{min} \quad \tau_{min} \leq \tau \leq \tau_{min}$$

The overall behavior of the controller depends on a large set of variables. While τ_{min} and τ_{max} depend on actuator peak torque, all the others parameters can be tuned quite freely. As a simple rule of thumb, the inequality constraints should be chosen considering the desired performance of the closed-loop system, but always taking care of not setting them too tight, otherwise the system may behave unexpectedly.

Other parameters whose values have a strong influence on the closed-loop dynamic behavior are the prediction horizon H_p and the control horizon H_c. Values of T_s, H_p and H_c should be chosen, in practical applications, according to the available computational resources. Every choice of T_s requires to solve the optimization problem $1/T_s$ times every second, and the computational cost of every evaluation is directly proportional both to H_p and H_c. Here $T_s = 10$ ms has been chosen as a tradeoff between the performance and the need for computational resources.

6.1 MPC control performances

Here a comparison between the system performance under PID and MPC is set. As it can be seen from figure 6, MPC provides a big step forward in vibration damping. Lateral displacement is effectively damped in a very short time (about 250 ms), and the reference position is being tracked with a remarkably high precision and speed: in roughly 200 ms the mechanism can reach its final position showing a very limited overshoot. This overshoot is also dramatically reduced in comparison to PID control [13]: the ability of MPC to predict the future behavior of the system allows to reduce the spring-back effects of the flexible link that usually arises when a flexible element is subject to high angular accelerations.



(a) (b)

Figure 4: Comparison between PID and MPC control system: (a) transverse vibration u_2 at the mid-point of the follower link, (b) Crank position q

7 CONCLUSIONS

A high-accuracy FEM-based dynamical model of a single-link mechanism with both rigid and flexible elements has been presented in this paper. This model has been implemented in a HIL real-time simulator to investigate the effectiveness of Model-based Predictive Control (MPC) with constraints for vibration damping in flexible mechanisms during high-speed rotations. In order to implement the control system, a linearized model of the dynamic system has been developed. This linearized state-space model is capable of a high precision approximation of mechanism dynamic behavior, on both position and vibration dynamics. The performances this control systems is compared to the ones that can be obtained trough a standard PID control. MPC control proved to be very effective both for reference position tracking and vibration suppression.

References

[1] Gasparetto, A., Zanotto V., “Vibration reduction in a flexible-link mechanism through synthesis of an optimal controller” *Meccanica*, **41**, 6, 611-622 (2006)

[2] Caracciolo, R., Richiedei, D., Trevisani, A., Zanotto, V., “Robust mixed-norm position and vibration control of flexible link mechanisms” *Mechatronics*, **15**, 7, 767-791 (2005)

[3] Maciejowski, J.M., “Predictive Control with Constraints”, Prentice Hall (1990).

[4] Camacho, E.F., Bordons, C., “Model Predictive Control”, Springer-Verlag, London (1999)

[5] Hassan, M., Dubay, R., Li, C., Wang, R., “Active vibration control of a one-link manipulator using a multivariable predictive controller” *Mechatronics* **17**, 311-323, (2007)

[6] Zmeu, K., Shiptico, E., “Predictive controller design with offline model learning for flexible beam control”, in *Proceedings of Physical and Cotrol* (2005).

[7] Röck, S., Pritshow, G., “Real-time capable Finite Element Models with closed-loop control: a method for Hardware-In-the Loop simulation of flexible systems”, *Production Engineering* **1** 37-43 (2007)

[8] Aghilli, F., Piefbœuf J.C., “Hardware-In-the-Loop Simulation of Robots Interacting with Environment via Algebric Differential Equations”, *IEEE/RSJ Internation Conference on Intelligent Robots and Systems* 1590-1596 (2000)

[9] Boscariol, P., Gasparetto, A., Lanzutti, A., Vidoni, R., Zanotto, V., “FLiMHILS: A Hardware-in-the-Loop Simulator of Flexible Links Mechanisms”, *Acta Mechanica Slovaca* **108** 2-A, 81-94 (2008)

[10] S. K. Dwivedy, P. Eberhard, Dynamic analysis of flexible manipulators, a literature review. *Mechanism and Machine Theory* 41, 749-777 (2006)

[11] M. Giovagnoni, A numerical and experimental analysis of a chain of flexible bodies, *Journal of Dynamic Systems, Measurement, and Control* 113, 73-80 (1994)

[12] A. Gasparetto, Accurate modeling of a flexible-link planar mechanism by means of a linearized model in the state-space form for design of a vibration control. *Journal of sound and vibration* vol. 240, no2, pp. 241-262 (2001)

[13] P. Boscariol, A. Gasparetto, A. Lanzutti, R. Vidoni, V. Zanotto, FLIMHILS: A Harware-In-the-Looop Simulator of Flexible Links Mechanisms, *Acta Mechanica Slovaca*, 2-A, 81-94 (2008)