Robust Model Based Predictive Control for Trajectory Tracking of Parallel Robots

A. Zubizarreta[1], I. Cabanes[1], M. Marcos[1], Charles Pinto[2], and J. Corral[2]

[1] Automatics and System Engineering Dpt., University of the Basque Country, Spain
asier.zubizarreta@ehu.es
[2] Dpt. of Mechanical Engineering, University of the Basque Country, Spain

Abstract. Model Based Predictive Control (MPC) is an interesting approach due to its ability to consider the constraints of the controlled system and easily adapt to the future reference changes. In this paper, a novel robust MPC controller is presented, which considers the effect of the Tool Center Point (TCP) estimation errors and the model uncertainties of the mechanical structure. In order to show its effectiveness, its application to the 5R parallel manipulator is detailed. Simulation validation is provided to demonstrate that the proposed approach can exploit all the theoretical capabilities of the mechatronic system.

Keywords: Parallel Robots, Model Based Predictive Control, Robust Control, Five Bar Mechanism.

1 Introduction

Parallel robots [8], have recently drawn attention from both academy and industry due to their performance when handling high-speed, high-precision or heavy load handling tasks. This performance is derived from their parallel structure, composed by multiple kinematic chains, or ”limbs”. However, its complexity presents some drawbacks, such as a reduced workspace, presence of singularities or highly coupled kinematics and dynamics.

In order to reduce the effect of these disadvantages, an optimized mechanical design, adequate actuator selection and proper control law that allows to exploit all the capabilities of the mechatronic system is required. In the literature, many different control approaches have been proposed, such as simple independent joint control approaches based on PID [3, 9] , or more advanced, model based control laws such as the Computed Torque Control (CTC) [11], adaptive control [5] or robust control [7]. However, none of the aforementioned approaches considers the physical limitations of the parallel robot (torques, workspace, speed) in the control law or adapts in a predictive way to the programmed reference trajectory.

V. Petuya et al. (eds.), *New Advances in Mechanisms, Transmissions and Applications*, Mechanisms and Machine Science 17,
DOI: 10.1007/978-94-007-7485-8_42,

Model Based Predictive Control (MPC) [4] groups a set of control strategies that use the dynamic model of a process to predict its behaviour in a finite future time window (the horizon). This way, its future error can be minimized by calculating a proper control action. Moreover, this control action can be calculated considering the physical constraints, which guarantees near optimal performance of the process and its actuators. These features are very interesting in robot control, as the potential of the mechatronic system can be maximized, actuator limits considered and workspace singularities avoided.

However, MPC controllers are usually complex and computationally intensive, and most approaches are based on constant references in the prediction horizon [10, 2]. In this paper, a novel, robust MPC for trajectory tracking (RMPC-T) is presented, based on the one developed in [1]. This approach presents three main features over the ones previously proposed: 1) it considers the physical constraints of the parallel robot, 2) it is robust against model uncertainties, and 3) it allows tracking of changing references, which enhances significatively the tracking capabilities of parallel robots. In order to detail this approach, the rest of the paper is structured in three sections. First, the robust MPC approach for tracking control law formulation is detailed for a generic system. Then, the application to the 5R parallel robot is detailed. Third, the effectiveness of the approach is demonstrated by simulation. Finally, the most important ideas are summarized.

2 Robust MPC for Trajectory Tracking

The novel robust MPC for trajectory tracking (RMPC-T) presented in this section is based on the one developed by Alvarado *et al* in [1]. However, while in the mentioned work constant trajectories are considered in the prediction horizon, in the formulation presented in this paper, changing trajectories are considered, which are more appropriate for parallel robots.

Next, the formulation of the MPC for a generic system is presented. For the sake of simplicity, some definitions are not fully detailed in this section. The reader is referred to [12] for more detail in the formulation and the concepts involved.

Consider the following uncertain system, defined in discrete time using state-space formulation,

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{w}(k) \\ \mathbf{y}(k+1) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \end{aligned} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ represents the state, $\mathbf{u} \in \mathbf{R}^m$ the control input and $\mathbf{y} \in \mathbf{R}^p$ the system output. $\mathbf{w} \in \mathscr{W}$ are the external additive disturbances that model parameter uncertainties and measurement errors. The system of Eq. (1) is considered to be time-invariant and controllable.

All sets are considered bounded, so that a convex polytope can be defined for each set to represent all points included within the bounds,

$$\mathscr{Z} = \left\{ \mathbf{z} = \left[\mathbf{x}^T \ \mathbf{u}^T \right]^T \in \mathbb{R}^{n+m} : \mathbf{A_z}\mathbf{z} \leq \mathbf{b_z} \right\}$$

where $\mathbf{A_z}$ and $\mathbf{b_z}$ are used to define the bounds of the set in the *h-representation* of a convex polytope.

The objective of the RMPC-T controller is, for each time step k, to stabilize the system and steer the state $\mathbf{x}$ to a neighbourhood of a steady state $\mathbf{x_s}$ associated to a setpoint $\mathbf{r}$, guaranteeing all the constraints even in presence of disturbances. For that purpose, a three level control approach is defined: a local robust control to reduce the effect of disturbances, a MPC controller which ensures feasibility with the system constraints and a local tracking controller to ensure convergence to the desired state.

The **local robust controller** is designed to avoid the exponential growth of the prediction error due to unknown $\mathbf{w}$ disturbances. Its goal is to ensure that the *real* state $\mathbf{x}$ lies within a bounded hipertube arround the trajectory of the nominal state $\overline{\mathbf{x}}$ (without disturbances), which is characterized by a *Robust Positively Invariant* (RPI) set [6] Φ_K. Hence, the local control law

$$\mathbf{u}(k+j) = \overline{\mathbf{u}}(k+j) + \mathbf{K}\left(\mathbf{x}(k+j) - \overline{\mathbf{x}}(k+j)\right) \tag{2}$$

where the nominal system $\overline{\mathbf{x}}$, $\overline{\mathbf{u}}$, $\overline{\mathbf{y}}$ is the *ideal* one and presents no disturbances. The gain matrix $\mathbf{K}$ is defined so that the error caused by the disturbances $\mathbf{w} = \mathbf{x} - \overline{\mathbf{x}} \in \Phi_K$ lies always in a bounded and fixed RPI set Φ_K. So, a bounded trajectory tube is generated for $\mathbf{z}$ that considers all possible uncertainties,

$$\overline{\mathscr{X}} = \mathscr{X} \ominus \Phi_K\ ,\ \overline{\mathscr{U}} = \mathscr{U} \ominus \mathbf{K}\,\Phi_K\ ,\ \overline{\mathscr{Z}} = \mathscr{Z} \ominus \left(\Phi_K \times \mathbf{K}\,\Phi_K\right), \overline{\mathbf{z}} = \left[\overline{\mathbf{x}}^T\ \overline{\mathbf{u}}^T\right]^T \in \overline{\mathscr{Z}}$$

where $\ominus$ is the Pontryagin difference.

Using the local robust control law defined in Eq. (2), it is ensured that the trajectory of the real state $\mathbf{x}$ lies within a bounded tube. This allows to consider the nominal trajectory of the state $\overline{\mathbf{x}}$ in the predictions of the MPC and the calculation of a feasible nominal control action $\overline{\mathbf{u}}^*$ that satisfies the contraints. However, in order to ensure convergence to the desired reference, the final predicted state of the MPC must lie within a neighbourhood of the desired steady state. This neighbourhood is defined as an *Invariant Set for Tracking* $\Omega^a_{t,\overline{\mathbf{K}}}$ and ensures that once reached this set both the state and internal reference $\boldsymbol{\theta}$ associated to the changing trajectory setpoint $\mathbf{r}$, evolve within the bounds of this set[1].

Based on the aforementioned sets and local controllers, the **MPC control law** is calculated by solving the minimization problem V_t^*,

$$V_t^* = \min_{\overline{\mathbf{u}},\overline{\mathbf{x}},\overline{\theta}} \mathbf{V}_t\left(\mathbf{x}(k), \boldsymbol{\theta}; \overline{\mathbf{u}}, \overline{\mathbf{x}}, \overline{\theta}\right)\ \text{s.t.} \begin{cases} \overline{\mathbf{x}}(k) \in \mathbf{x}(k) \oplus (-\Phi_K) \\ \left[\overline{\mathbf{x}}(k+j)^T\ \overline{\mathbf{u}}(k+j)^T\right]^T \in \overline{\mathscr{Z}} \\ \left[\overline{\mathbf{x}}_\mathbf{s}(k+j)^T\ \overline{\mathbf{u}}_\mathbf{s}(k+j)^T\right]^T = \mathbf{M}_{\boldsymbol{\theta}}\,\overline{\boldsymbol{\theta}} \\ \left[\overline{\mathbf{x}}(k+h)^T\ \overline{\theta}(k+h)^T\right]^T \in \Omega^a_{t,\overline{\mathbf{K}}} \end{cases} \tag{3}$$

where h is the prediction horizon, i.e, the number of time steps into the future that the controller uses to calculate the optimal control action sequence $\overline{\mathbf{u}}^*(k)$, $\boldsymbol{\theta}$ is the sequence associated to the changing trajectory setpoint $\mathbf{r}$, $\Omega^{a}_{t,\overline{\mathbf{K}}}$ is the Invariant Set for Tracking, Φ_K is the robust positively invariant set and the cost function,

$$\begin{aligned}\mathbf{V}_t\left(\mathbf{x}(k),\boldsymbol{\theta};\overline{\mathbf{u}},\overline{\mathbf{x}},\overline{\theta}\right) = &\sum_{j=0}^{h-1}\left(\overline{\mathbf{x}}(k+j)-\overline{\mathbf{x}}_{\mathbf{s}}(k+j)\right)^T \mathbf{Q}\left(\overline{\mathbf{x}}(k+j)-\overline{\mathbf{x}}_s(k+j)\right)\\ &+\sum_{j=0}^{h-1}\left(\overline{\mathbf{u}}(k+j)-\overline{\mathbf{u}_{\mathbf{s}}}(\mathbf{k}+\mathbf{j})\right)^T \mathbf{R}\left(\overline{\mathbf{u}}(k+j)-\overline{\mathbf{u}}_{\mathbf{s}}(k+j)\right)\\ &+\left(\overline{\mathbf{x}}(k+h)-\overline{\mathbf{x}}_{\mathbf{s}}\right)^T \mathbf{P}\left(\overline{\mathbf{x}}(k+h)-\overline{\mathbf{x}}_{\mathbf{s}}\right)\\ &+\sum_{j=0}^{h}\left(\overline{\theta}(k+j)-\boldsymbol{\theta}(k+j)\right)^T \mathbf{T}\left(\overline{\theta}(k+j)-\boldsymbol{\theta}(k+j)\right)\end{aligned}$$

where $\mathbf{Q}$, $\mathbf{R}$, $\mathbf{T}$ and $\mathbf{P}$ are ponderation matrices, whose tuning is discussed in [12].

Finally, the real one control action by means of the **local tracking controller**, which considers the error generated by disturbances,

$$\mathbf{u}^*(k) = \overline{\mathbf{u}}^*(k) + \mathbf{K}\left(\mathbf{x}(k) - \overline{\mathbf{x}}^*(k)\right) \tag{4}$$

3 Application to the 5R Parallel Robot

The control law detailed in the previous section is defined for discrete, space state time invariant systems, and requires the definition of bounds in order to be implemented. In this section the procedure to implement the RMPC-T to parallel robots is detailed by analyzing a study case based on the 5R parallel robot. However, it should be noted that this procedure can be applied to any parallel robot.

The first requirement to be fulfilled is to linealize the dynamics of the 5R parallel robot (Fig. 1), which can be calculated using the traditional formulation in the *task space* $\boldsymbol{\tau} = \mathbf{D}\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q},\dot{\mathbf{q}})$, where $\mathbf{q} = \left[x\ y\right]^T$ are the Tool Center Point (TCP) cartesian coordinates[13]. Table 1 summarizes the parameters selected for this study case.

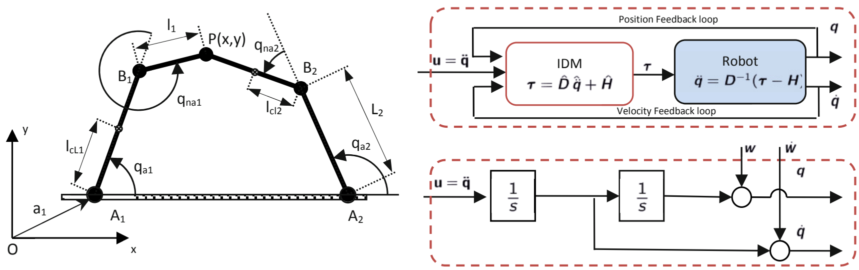


Fig. 1 5R Parallel Robot and nonlinear feedback linearization approach

Table 1 5R parallel robot model parameters (IS units)

Parameter	Real Value	Identified Value (MPC Model)
$\mathbf{a_1}$	$\begin{bmatrix} -0.5 & 0 \end{bmatrix}^T$ (m)	$\begin{bmatrix} -0.4950 & -0.005 \end{bmatrix}^T$ (m)
$\mathbf{a_2}$	$\begin{bmatrix} 0.5 & 0 \end{bmatrix}^T$ (m)	$\begin{bmatrix} 0.4950 & -0.005 \end{bmatrix}^T$ (m)
$L_1 = L_2$	0.5 (m)	0.495 (m)
$l_1 = l_2$	1 (m)	0.995 (m)
m_c	0.5 (kg)	0.6 (kg)
m_{L_1}	0.4239 (kg)	0.3239 (kg)
m_{L_2}	0.4239 (kg)	0.4239 (kg)
$m_{l_1} = m_{l_2}$	0.8477 (kg)	0.7477 (kg)
$I_{L_1} = I_{L_2}$	$8.800 \cdot 10^{-3}$ (kg m^2)	$8.800 \cdot 10^{-3}$ (kg m^2)
$I_{l_1} = I_{l_2}$	$7.070 \cdot 10^{-2}$ (kg m^2)	$7.770 \cdot 10^{-2}$ (kg m^2)
I_c	$8.3333 \cdot 10^{-4}$ (kg m^2)	$8.3333 \cdot 10^{-4}$ (kg m^2)

For that purpose, the linearization by nonlinear feedback technique is used, so that in the ideal case, if no model errors arise, the linearized system can be reduced to a set of two decoupled double integrator systems (Fig. 1) $\mathbf{u} = \ddot{\mathbf{q}}$. Hence, the state vector of the linealized system can be defined as $\mathbf{x} = \begin{bmatrix} \mathbf{q}^T & \dot{\mathbf{q}}^T \end{bmatrix}^T$.

One of the main issues in parallel robotics is the difficulty measuring the real TCP position and speed, which is usually estimated using the actuated joint data and the use of the kinematic relations. In presence of uncertainties, this estimation can present errors, so that the real state $\mathbf{x}$ and the estimated one $\hat{\mathbf{x}}$ diverge. In general, it is possible to bound this error within the operational workspace of the robot, so that the linearized dynamic model can be approximated to,

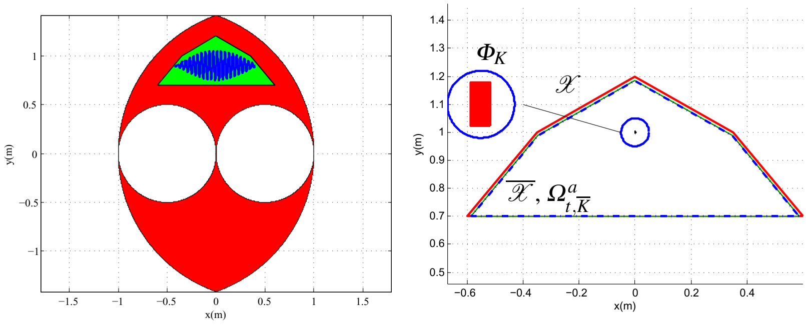
$$\begin{array}{c} \hat{\mathbf{x}}(k+1) = \mathbf{A}\hat{\mathbf{x}}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{w}'(k) \\ \mathbf{x} = \hat{\mathbf{x}} + \mathbf{v} \end{array} \rightarrow \begin{array}{c} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{w}(k) \\ \mathbf{w}(k) = \mathbf{w}'(k) + (\mathbf{I} - \mathbf{A})\mathbf{v}(k) \end{array} \tag{5}$$

where $\mathbf{v}(k) = \mathbf{x} - \hat{\mathbf{x}}$ models estimation errors and $\mathbf{w}'(k)$ models errors due to the uncertainties of the dynamic model and uncompensated dynamics of the robot. Hence, if the real state $\mathbf{x}$ is considered for application of the MPC control law, $\mathbf{w}(k)$ group the disturbances of both types.

If a sample time of $T_s = 10ms$ is selected, the discretized dynamics of the set of double integrator systems in the 5R parallel manipulator is,

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0.01 & 0 \\ 0 & 1 & 0 & 0.01 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 5 \cdot 10^{-5} & 0 \\ 0 & 5 \cdot 10^{-5} \\ 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}$$

In order to implement the proposed MPC, all variables have to be bounded, and the disturbances maximum and minimum values defined. For that purpose, first the operational workspace of the 5R parallel robot will be defined arbitrarily in a



(a) Operational Worskpace (b) Invariant sets

Fig. 2 Validation trajectories, operational workspace and invariant sets

nonsingular region of its workspace associated to a fixed working and assembly mode. This region will be defined by a set $\mathscr{X}$. However, as the real state $\mathbf{x}$ cannot be measured directly, the MPC will be implemented using the estimated one $\hat{\mathbf{x}}$. The RMPC-T controller will ensure that the real state $\mathbf{x}$ will be always within the tube of trajectories defined by the estimated one $\hat{\mathbf{x}}$ and the RPI Φ_K,

$$\hat{\mathbf{x}} \in \hat{\mathscr{X}} = \mathscr{X} \ominus \boldsymbol{\Phi}_K \Rightarrow \hat{\mathbf{x}}' \in \hat{\mathscr{X}} \tag{6}$$

where $\mathscr{X}$ is defined by a polytope defined in XY plane, Fig. 2a, and the maximum linear speed of the TCP, which has been limited to $\pm 3m/s$.

The bounds of the estimation error $\mathbf{v}$ are calculated by discretizing the operational workspace defined for $\mathscr{X}$ and measuring the speed and positioning errors for each set. In order to bound the state disturbance $\mathbf{w}'$, the performance of the ideal double integrator system and the linealized dynamics of the 5R parallel robot considering the parameters of Table 1 are considered. The resulting bounds are,

$$\begin{array}{ll} v_x \in [-0.0014, 0.0016] \text{ (m)} & w_x \in \left[-5.4131 \cdot 10^{-5}, 2.6455\, 10^{-5}\right] \text{ (m)} \\ v_y \in [0.0025, 0.0091] \text{ (m)} & w_y \in \left[-6.1584 \cdot 10^{-5}, 1.1981\, 10^{-4}\right] \text{ (m)} \\ v_{\dot{x}} \in [-0.0324, 0.0324] \text{ (m/s)} & w_{\dot{x}} \in [-0.01131, 0.0039132] \text{ (m/s)} \\ v_{\dot{y}} \in [-0.0425, 0.0425] \text{ (m/s)} & w_{\dot{y}} \in [-0.0093394, 0.022516] \text{ (m/s)} \end{array} \tag{7}$$

Finally, the system input is bounded. For that purpose, the inverse dynamic model is required $\mathbf{u} = \hat{\mathbf{D}}^{-1}\left(\boldsymbol{\tau} - \hat{\mathbf{H}}\right)$, which relates the input of the linealized robot, $\mathbf{u}$ and the torque exerted by the motors $\boldsymbol{\tau}$. Being nonlinear and dependent on the state $\mathbf{x}$, a conservative approach is considered, in which the set of all possible admissible sets for $\mathbf{u}$ are calculated considering all admissible states $\mathbf{x} \in \mathscr{X}$. Then, the intersection of all sets is considered as the admissible set $\mathscr{U}$.

4 Simulation Results

The linealized dynamic model and the bounded sets defined previously have been used to implement the RMPC-T. The prediction horizon has been set in $h = 3$ steps and the ponderation gains and local robust controller gains have been tuned following the procedure detailed in [12]. The resulting invariant sets are defined in Fig. 2b.

In order to demonstrate the effectiveness of the robust MPC for tracking approach, it has been compared with the robust MPC for tracking proposed in [1] (RMPC) and the classical Computed Torque Control (CTC) approach. Both controllers have been tuned to achieve maximum performance within the defined physical constraints. A senoidal trajectory in the XY plane and within the operational workspace has been selected as reference (Fig. 2a).

TCP positioning errors and mean of the squared error (mse) performance indexes are shown in Fig. 3. As it can be seen, the best performance is achieved by the RMPC-T, reducing the tracking error in 47% in comparison with the RMPC and 30% with respect to the CTC. Hence, the proposed RMPC-T is able to adapt to the future changing trajectory before changes actually occur, resulting in very low tracking error. The RMPC [1], however, considers a constant reference in its prediction horizon, which penalizes its performance. Finally, the classical CTC is not able to anticipate to future reference changes, and focuses in compensating the actual error, which leads to larger trajectory tracking errors. Hence, the main advantage of the proposed RMPC-T controller is demonstrated, which can be implemented to reduce significatively the tracking error of parallel robots.

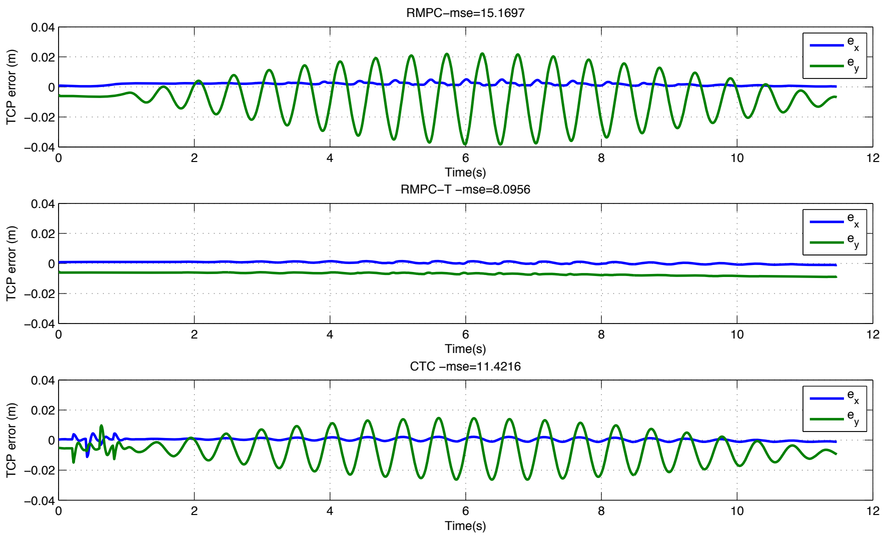


Fig. 3 Performance of Robust MPC, Robust MPC for traj. tracking and Computed Torque Control approaches

5 Conclusions

Control is a key issue in parallel robotics, as proper control approaches are required in order to exploit the theoretical capabilities. In this work a novel Robust Model Predictive Control approach for trajectory tracking (RMPC-T) has been presented, and its application to a 5R parallel robot prototype detailed. Simulation results show that this approach can provide enhanced tracking capabilities to parallel robots in comparison with classical CTC or MPC approaches.

Acknowledgements. This work was supported in part by the Government of Spain under projects DPI2011-22955 and DPI2012-32882, the Government of the Basque Country (Project IT719-13 and IT445-10) and UPV/EHU under grant UFI11/28 and UFI 11/29.

References

1. Alvarado, I.: Model predictive control for tracking constrained linear systems. Ph.D. thesis, Escuela Técnica Superior de Ingenieros Universidad de Sevilla (2007)
2. Belda, K., Böhm, J., Valášek, M.: State-Space Generalized Predictive Control for Redundant Parallel Robots. Mechanics Based Design of Structures and Machines 31(3), 413–432 (2003)
3. Brecher, C., Ostermann, T., Friedrich, D.: Control concept for pkm considering the mechanical coupling between actuator. In: Proceedings of the 5th Chemnitz Parallel Kinematics Seminar, pp. 413–427 (2006)
4. Camacho, E., Bordons, C.: Control predictivo: Pasado, presente y futuro. Revista Iberoamericana de Automtica e Informtica Industrial (RIAI) 1(3) (2004) (in Spanish)
5. Honegger, M., Codourey, A., Burdet, E.: Adaptive control of the hexaglide, a 6 dof parallel manipulator. In: IEEE International Conference in Robotics and Automation (1997)
6. Kolmanovsky, I., Gilbert, E.G.: Theory and conmputation of disturbance invariant sets for discrete-time linear systems. Mathematical Problems in Engineering: Theory, Methods and Applications 4, 317–367 (1998)
7. Lee, S.H., Song, J.B., Choi, W.C., Hong, D.: Position control of a stewart platform using inverse dynamics control with approximate dynamics. Mechatronics 13, 605–619 (2003)
8. Merlet, J.-P.: Parallel Robots, 2nd edn. Kluwer (2006)
9. Stan, S., Manic, M., Maties, M., Balan, R.: Kinematics analysis, design, and control of an isoglide3 parallel robot (ig3pr). In: Proc. 34th Annual Conference of the IEEE Industrial Electronics Society, pp. 1265–1275 (2008)
10. Vivas, A., Poignet, P.: Model based predcitive control of a fully parallel robot. In: Proceedings of the 7th IFAC Symposium on Robot Control (SYROCO 2003), pp. 253–258 (2003)
11. Yen, P., Lai, C.: Dynamic modeling and control of a 3-dof cartesian parallel manipulator. Mechatronics 19(3), 390–398 (2009)
12. Zubizarreta, A.: Estrategias de control avanzado para robots paralelos. Ph.D. thesis, University of the Basque Country, UPV/EHU (2010) (in spanish)
13. Zubizarreta, A., Cabanes, I., Marcos-Muñoz, M., Pinto, C.: Experimental validation of the extended computed torque control approach in the 5r parallel robot prototype. In: Su, C.-Y., Rakheja, S., Liu, H. (eds.) ICIRA 2012, Part II. LNCS (LNAI), vol. 7507, pp. 509–518. Springer, Heidelberg (2012)