A new fast nonlinear model predictive control of parallel manipulators: Design and experiments

Ahmed Chemori, Rihab Kouki, Faouzi Bouani

A New Fast Nonlinear Model Predictive Control of Parallel Manipulators: Design and Experiments

Ahmed Chemori[a,*], Rihab Kouki[b], Faouzi Bouani[b]

[a] *LIRMM, University of Montpellier, CNRS, Montpellier, France*
[b] *Université de Tunis El Manar, Ecole Nationale d'Ingénieurs de Tunis, Laboratoire Analyse, Conception et Commande des Systèmes, LR11ES20 Tunis, Tunisia*

Abstract

High-speed parallel manipulators are characterized by fast sampling rates and challenges owing to the presence of constraints, high nonlinearities, uncertainties, and fast dynamics. The development of a reliable nonlinear model predictive control (NMPC) approach for this type of robotic systems is challenging because of the complex dynamics that involves high computational burden. To address this control problem, this paper proposes a new fast NMPC strategy applied to high-speed parallel robots. Based on (i) a parameterization technique, (ii) a fast gradient solver, (iii) a proportional integral derivative (PID) control term, (iv) a nonlinear dynamic model, and (v) an artificial neural network (ANN) model, two fast NMPC frameworks were developed to reduce the computational cost of the classical NMPC and address online implementation. In this study, we focus on improving the performance of the standard predictive control strategy outside of the range of its classical applicability by developing a fast NMPC controller for high-speed parallel kinematic manipulators (PKMs). Numerical simulations and real-time experimental results were presented and discussed to validate the relevance of the proposed controllers. Experiments were conducted on a four-degree-of-freedom (4-DOF) PKM called VELOCE developed in our laboratory. The results show that the proposed solution outperforms the original scheme in terms of real-time tracking performance.

Keywords: `Parallel kinematic manipulators, fast nonlinear model predictive control, real-time experiments`

Abbreviations List

ANN	Artificial neural network
CAD	Computer aided design
GA	Genetic algorithm
LSM	Least square method
MPC	Model predictive control
NMPC	Nonlinear model predictive control
PID	Proportional - integral - derivative
PKM	Parallel kinematic manipulator
PSO	Particle swarm optimization
RMSE	Root meat square error
Ex-Fast NMPC	Extended fast NMPC

1. Introduction

Model predictive control strategy has been considerably developed over the last few years within both the research control community and industry. The history of this control technique dates back to the 1960s, which has led to powerful characterizations, such as the maximum principle of Pontryagin [1] and the dynamic programming method developed by Bellman [2]. Owing to its remarkable success, predictive control strategies have become a popular topic in both academic and industrial studies [3][4]. They have evolved rapidly to include many processes with uncertain parameters and nonlinear systems [5, 6, 7, 8]. The nonlinear model predictive control (NMPC) approach is an advanced control strategy that provides anticipatory actions using an internal nonlinear model to predict the future behavior of the system. However, one of the major challenges in real-time implementation of NMP controller lies in its computational efficiency.

In recent years, many efforts have been made to extend the applications of NMPC strategies in the field of fast dynamical systems (e.g., robots, mechatronic systems, and automotive and drive processes [9, 10, 11, 12]). In the literature, various optimal control techniques have been developed based on the requirements of NMPC to improve its computational efficiency. For instance, in [13], nonlinear optimization of the NMPC based on a neural network was developed for an embedded system. However, the optimization problem was reformulated as a convex quadratic programming problem with unknown nonlinear terms, and it was solved using a simplified dual network. Wu et al. [14] proposed the implementation of nonlin-

*Corresponding author
Email addresses: `ahmed.chemori@lirmm.fr` (Ahmed Chemori), `rihab.kouki@enit.rnu.tn` (Rihab Kouki), `faouzi.bouani@enit.utm.tn` (Faouzi Bouani)

ear model predictive control with an economic objective function to reduce computational effort. The effectiveness of the proposed approach has been demonstrated only through numerical simulations. Inspired by the work of [15], a parameterized NMPC scheme was proposed in [16] to improve the output power control of a 10 kW proton exchange membrane fuel cell system. This approach significantly reduces the optimization problem dimension, compared to a standard NMPC controller. It was validated offline through numerical simulations and proved to be a promising candidate for online implementation in future studies. In [17], a low-dimensional parameterized NMPC strategy was proposed for quadrotors using the fmincon optimization toolbox MATLAB environment to solve nonlinear optimization problems. However, this function considerably increases the computational time load, which is inadvisable for real-time applications. A fast NMPC firmware based on a control-updating period method was developed by [18] to meet real-time control requirements of fast systems. The efficiency of the proposed strategy was proven only through numerical simulations using a simple example of a constrained triple integrator.

Despite the successful implementation of fast NMPC controllers in various industrial applications, they are typically limited to a specific class of simple fast systems [19][20]. This concept has motivated the development of a fast NMPC approach for more complex robotic systems. The proposed approach was directly applied to parallel kinematic manipulators (PKMs). These robots are of growing interest in various fields, such as 3D printing [21], pick-and-place [22], laser cutting [23], and even surgical interventions [24]. PKMs are typically characterized by their high precision, high nonlinear dynamics, obtained high-speed singularities, time-varying parameters, and uncertainties [25] [26]. However, they are characterized by certain limitations and challenges, such as the limited size of the workspace, rotational motion, complexity of forward kinematic solutions, real-time constraints, and the issue of actuation redundancy in some cases. Furthermore, their closed-kinematic-chain structure gives rise to highly nonlinear dynamics that must be carefully considered in control design [27]. To the best of authors' knowledge, the NMPC controller has never been tested experimentally on high-speed parallel manipulators and has become an active research area. Then, we extend state-of-the-art fast NMPC controllers by proposing a solution that allows the explicit handling of nonlinear high-speed robots.

To address this issue, two extended fast NMPC control schemes were introduced in this study to improve the computational performance of the standard NMPC controller. This work is substantial research that aims to introduce extensions and address new challenges beyond the results presented in the last decade [16], [28], [29], [14], [13] and [30]. Regarding the first contribution, an extended fast NMPC controller based on a dynamic model, parametrization approach, and PID control feedback term is proposed. Compared with previous studies, in this study, a theoretical analysis and an experimental tuning process related to the proposed extended fast nonlinear MPC was conducted to reduce the optimization problem dimension. This study has particularly improved the computation time of the standard NMPC and fulfill real-time control requirements. For the second contribution, an artificial neural network (ANN) model was employed as an internal model to predict the system future behavior to improve the overall closed-loop performance of the proposed NMPC scheme for PKMs. The introduction of a specific parameterized approach and an internal ANN model in the control scheme of the second contribution has significantly improved the computational efficiency (allowing an increase in the values of NMPC parameters (prediction horizon N_p, control horizon N_c) to improve the controller performance.

In the literature, several recent studies proposed the combination of ANNs with MPC. Overall, the main existing studies can be classified into the following two classes. The first class concerns *ANN-based learned models for MPC*, in which the main concept is to create an ANN to model the dynamics of the system to be controlled. System data (history of states and controls) were used to construct a neural network model that was integrated as the system model in the MPC scheme to predict its future behavior. Four examples of recent studies within this class include [31] [32] [33] and [34]. The second class concerns *ANN-based learned MPC controllers* wherein the concept involves creating and training an ANN based on data collection from an MPC algorithm to emulate this last one. Accordingly, this can effectively reduce the computational burden of online control. Four examples of recent works within this class are as follows: [35] [36] [37] and [38].

Particularly, in this paper, we discuss in more details the improvement of the computational burden of the standard NMPC controller to achieve the real-time control for PKMs. Compared with [16], [28], [14], [13] and [30], this study considers the nonlinear predictive control of high-speed PKMs. The design of a suitable predictive control scheme for PKMs is a challenging task owing to their complex dynamics. This type of robot requires a high computational performance and nonlinearities that may increase when operating at high speeds and accelerations. Furthermore, we intend to emphasize the difficulty of working with high-speed systems (e.g., VELOCE PKM whose sampling time is equal to 0.1 ms), where the NMPC controller requires an online solution of a receding-horizon optimization problem at each sampling instant. To overcome the computational burden of standard NMPC controllers, our proposed control solutions were developed without using any linearization techniques based on a fast gradient solver and without using an optimization toolbox, such as those used in [16] and [28]. After reducing the complexity of the optimization problem, controlling a high-speed dynamic system becomes a concrete real-time application possibility. This study mainly aims to surpass theoretical developments to validate the proposed control strate-

gies through real-time experiments on a high-speed parallel robot, namely, VELOCE PKM. Besides, it is worth to emphasize that the present work brings new contributions and extensions beyond the results of [29], including (i) a more deeply literature review, (ii) the second contribution is completely new, (iii) new simulation and experimental results, (iv) algorithms 2, 3, 4 and 5 are completely new. The remainder of this paper is organized as follows. A dynamic modeling of VELOCE PKM is presented in Section 2. Section 3 introduces the transformation of the standard NMPC approach into a fast NMPC strategy for a fast dynamic system. Section 4 describes the development of the proposed fast NMPC extension method. In this context, we explain how nonlinear model predictive control schemes and fast solving methods can be proposed to improve performance of PKMs. Section 5 presents the numerical simulation and real-time experimental results. Finally, general conclusions and future work are presented in Section 6.

2. Veloce Prototype: Description and Modeling

VELOCE robot is a 4-DOF parallel mechanism designed and fabricated at LIRMM Laboratory (France). It is a fully parallel manipulator; that is, it has the same number of kinematic chains as DOFs [39]. The CAD view of VELOCE is shown in Figure 1. The VELOCE PKM

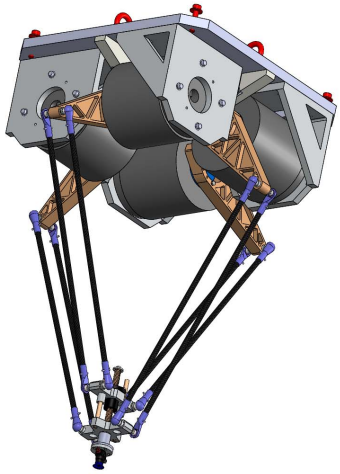


Figure 1: CAD view of VELOCE parallel kinematic manipulator.

consists of four linked actuated kinematic chains. for a typical moving platform. Each kinematic chain is a serial arrangement of an actuator, rear arm, and forearm (Figure 2). The moving platform of VELOCE comprises two essential parts: an upper and lower part. Both parts were mounted on a single screw, and the relative motion between the lower and upper parts generated the rotational DOF of the platform. The four actuators responsible for the movement of the mechanical structure were all located on the same plane.

2.1. Robot Kinematics

The pose of the moving platform can be described by a four-dimensional coordinate vector $X = [x, y, z, \alpha]^T$ such that x, y, z are the translational coordinates and α is the rotational angle around the z-axis. Note VELOCE is a fully actuated mechanism, and the actuated joint coordinates are defined by vector $q = [q_1, q_2, q_3, q_4]^T$. Joint position q can completely define the configuration of the entire mechanism. The relationship between this configuration vector and the platform pose was obtained through a kinematic study of the constraints of closing the loop formed by the four kinematic chains. Note that the mapping from Cartesian to joint velocities is achieved using the inverse Jacobian matrix, which is expressed as follows:

$$\dot{q} = J_m \dot{X} \tag{1}$$

where $J_m \in \mathbb{R}^{4\times4}$ denotes the inverse Jacobian matrix. This matrix is square and invertible for full PKMs (VELOCE). The robot configuration was assumed to be away from singularities. By differentiating equation (1) with respect to time, we obtain the relationship between Cartesian and joint accelerations as follows:

$$\ddot{q} = J_m \ddot{X} + \dot{J}_m \dot{X} \tag{2}$$

2.2. Nonlinear Dynamic Model of VELOCE PKM

To compute the dynamic model of VELOCE robot, the following assumptions were made [40]:

- The inertia of the forearms is neglected and their mass is split up into two equivalent parts and considered in both the dynamics of the actuators and the moving platform.
- Both dry and viscous frictions of all joints are not considered.

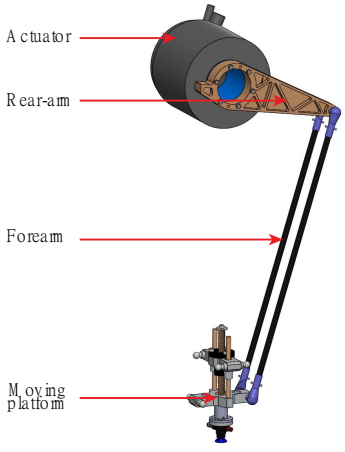


Figure 2: Schematic view for one kinematic chain of VELOCE PKM as well as its main components.

According to [41], a dynamic model can be established by analyzing the equilibrium of the arms and moving platform. These two quantities were summed.

The relationship between the torque vector Γ and joint acceleration vector $\ddot{q}$ is expressed as follows:

$$\Gamma - J_q^T F = (I_{act} + I_{arm} + \frac{l^2 M_{forearm}}{2})\ddot{q} \quad (3)$$

where $\Gamma \in \mathbb{R}^4$ denotes the control input vector, $J_q \in \mathbb{R}^{4\times4}$ denotes the joint Jacobian matrix, and $F \in \mathbb{R}^4$ represents the force vector associated with the acceleration of the traveling plate. I_{act}, $I_{arm} \in \mathbb{R}^{4\times4}$ are the inertia matrices of the motor drives and the arms, respectively. $l = diag\{l_1, \ldots, l_4\}$ with l_i the length of the i^{th} arm. $M_{forearm} \in \mathbb{R}^{4\times4}$ is the mass matrix of the forearms. The dynamic parameters of the robot are reported in Table 1

Table 1: Main dynamic parameters of VELOCE parallel robot.

Parameter	Value	Description
M_{tp}	0.257	Mass of the traveling plate (kg)
$M_{forearm}$	0.080	Mass of the forearm (kg)
I_{act}	0.041	Inertia of the actuators $(kg \cdot m^2)$
I_{arm}	0.0053	Inertia of the arms $(kg \cdot m^2)$

The equation of motion of the moving platform is expressed as

$$J_x^T F = (M_{tp} + 4\frac{M_{forearm}}{2})\ddot{X} \quad (4)$$

where $J_x \in \mathbb{R}^{4\times4}$ denotes the Cartesian Jacobian matrix, $M_{tp} \in \mathbb{R}^{4\times4}$ denotes the mass matrix of the moving platform, and $\ddot{X} \in \mathbb{R}^4$ denotes the acceleration vector. Note that the complete Jacobian matrix of the manipulator is expressed as a function of the manipulator's pose and is given by $J_m = J_q^{-1} J_x$.

By substituting (3) into (4), the entire equilibrium of the manipulator can be formulated as

$$\ddot{X} = (M_{tp} + 4\frac{M_{forearm}}{2})^{-1} J_m^T[\ \Gamma - (I_{act} + I_{arm} + \frac{l^2 M_{forearm}}{2})\ddot{q}\ ] \quad (5)$$

To simplify, the dynamic model (5) can be expressed in Cartesian space as follows:

$$\ddot{X} = (M_{tot} + J_m^T I_{tot} J_m)^{-1} J_m^T(\Gamma - I_{tot}\dot{J}_m\dot{X}) \quad (6)$$

where $M_{tot} = M_{tp} + 4\frac{M_{forearm}}{2}$ is the total mass matrix of the manipulator and $I_{tot} = I_{act} + I_{arm} + \frac{l^2 M_{forearm}}{2}$ is the total inertia matrix.

Based on the appropriate kinematic relationships, the dynamic model of VELOCE can be rewritten in a standard joint space form as follows:

$$M(q)\ddot{q} + N(q,\dot{q})\dot{q} + G(q) = \Gamma \quad (7)$$

where $M(q) = I_{tot} + (J_m^T)^{-1} M_{tot} J_m^{-1} \in \mathbb{R}^{4\times4}$ is the total inertia matrix, $N(q,\dot{q}) = -(J_m^T)^{-1} M_{tot} J_m^{-1} \dot{J}_m J_m^{-1} \in \mathbb{R}^{4\times4}$ is the Coriolis and centrifugal forces vector, and $G(q) = -(J_m^T)^{-1} G_{tp} - \Gamma_{G_a}$ is the gravity force vector.

The dynamic model in Equation (7) is suitable for joint-space control design and is used in the proposed controller development. It is worth emphasizing that the proposed control schemes are designed for a 4-DOF manipulator, remain valid, and can be easily generalized to n-DOF manipulators, given the properties of the dynamic model of such mechanical/robotic systems. The dynamic model (7) is suitable for the joint-space control design. Regardless of whether the number of joints of the robotic manipulator is planar, this equation and the terms that are constitutive are quite interesting for the structural analysis, study of control designs, and stability analysis via the Lyapunovs direct method. Thus, we can implement and experimentally validate our proposed controllers with any other system by using the identified dynamic model.

3. From Classical to Fast NMPC Control Scheme

As previously described, VELOCE is a non-redundantly actuated parallel manipulator proposed to validate the developed fast NMPC controller. The following sections proposes first a brief introduction to the standard NMPC controller and its evolution over the years to meet the needs of fast system control. An overview of existing fast solver solutions in the literature regarding predictive control strategies is provided before finally discussing the main contributions of this study.

3.1. Problem Formulation of Classical NMPC

The main concept behind predictive control in our case is to predict the future behavior of 4-DOF parallel manipulators based on the dynamic model. It then computes an optimal control sequence by solving online, an open-loop nonlinear optimization problem. This optimization problem aims to determine the optimal solution by minimizing a nonlinear cost function J expressed by the mean squared difference between the predicted outputs and the reference trajectory under a set of constraints [5]. Before proceeding to the development of a fast NMPC control law, let us briefly recall the block diagram of a standard NMPC controller based on a nonlinear solver and discretization method (Figure 3). The nonlinear programming (NLP) problem is solved by computing the minimum of a constrained nonlinear multivariable function using numerical methods, such as Runge-Kutta of order 4 and the ODE

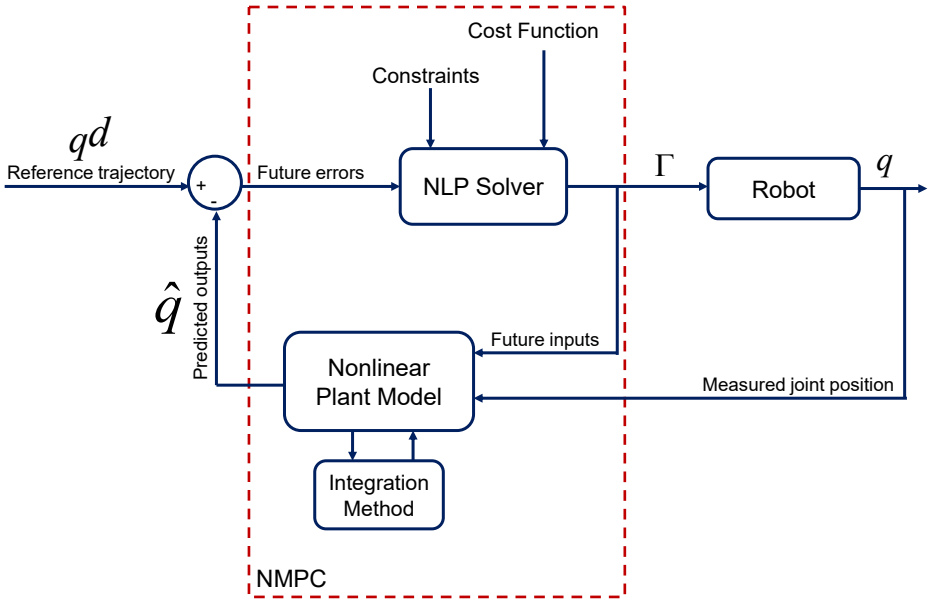


Figure 3: Block diagram of the computed torque control based on the standard NMPC scheme.

45 numerical integrator of MATLAB software. Predictive control strategies involve receding horizon optimization, which may be quite time-consuming, especially for nonlinear optimization problems with constraints. Consequently, such strategies are typically used for slow systems, such as chemical plants and oil refineries. Thus, we must reduce the computation time per control iteration of the following optimization problem if we intend to control high-speed parallel manipulators using a control scheme.

$$\hat{\Gamma}(q) := Arg\min_{\Gamma}[J(\Gamma, q)] \tag{8}$$

with cost function J, which can be defined as follows:

$$J(\Gamma, q) = \sum_{i=0}^{N_p-1} (\hat{q}(i, q, \Gamma) - q^d(k, i))^2 + \lambda_1 \sum_{i=0}^{N_c-1} (\Delta\Gamma(k, i))^2 \tag{9}$$

where $q^d(k, i)$ denotes the desired trajectory and λ_1 is a weighting coefficient. Further, $\Delta\Gamma(k, i))$ represents the control input increment at instant k, which is defined as $\Delta\Gamma(k, i) = \Gamma(k, i) - \Gamma(k-1, i)$; $q \in \mathbb{R}^n$ is the joint position vector for n actuators; and $\hat{q}(i, q, \Gamma)$ defines the future output predictor vector defined by the optimal control Γ over $[k, k + N_c - 1]$ and started from the one step ahead joint position q. The general control sequence $\Gamma \in \Re^n$ of the optimal solution of the problem (8) comprises N_c input values and can be defined as follows:

$$\Gamma = [\Gamma(k), \Gamma(k+1), \cdots, \Gamma(k+N_c-1)] \tag{10}$$

According to the basic principle of NMPC, the first sample of the computed optimal control sequence $\hat{\Gamma}$ at time k, namely, $\hat{\Gamma}(k)$, is applied to the system until the next sampling instant. Then, the prediction horizon is initially moved forward, and the optimization problem is resolved based on new system measurements.

To address nonlinear optimization problems, the *fmincon* MATLAB routine is a possible solution for the development of numerical algorithms. This function solves the optimization problem of the NMPC strategy using the interior-point method, sequential quadratic programming (SQR), active-set algorithms, etc. However, we note that the *fmincon* function is appropriate for a PC-based environment, but not suitable for the embedded implementation. Indeed, the *fmincon* function requires high computation time, which is not suitable for real-time applications. The standard NMPC strategy then becomes a challenging task for highly nonlinear systems that require an online solution to a receding-horizon optimization problem at each sampling instant. The high online computational burden has limited the application of NMPC in many fields, such as fast dynamic systems.

3.2. Fast Solver Methods

To handle fast systems, efficient methodologies for fast NMPC have been proposed in the literature, such as the continuation/generalized minimum residual (C/GMRES) method [42], real-time iteration scheme multiple shooting [43], path-following method [44], and C/C++ code generation tools [45]. These techniques have an interesting feature of being effective solutions for addressing the real-time issue of fast dynamic systems. The C/GMRES method is a numerical method proposed for nonlinear receding-horizon control to reduce its computational complexity [42]. In [43], a real-time iteration scheme with multiple shots was

introduced to handle the nonlinear optimization of NMPC. This method is suitable for controlling small problem sizes in nonlinear dynamic systems. The path-following method is one of the most efficient approaches among the available algorithms. This controller was applied within the advanced-step NMPC framework to obtain fast and accurate approximate solutions to the NMPC problem [44]. In [18], a fast NMPC approach based on a control-updating period method was designed to decrease the optimization problem dimension for the tracking control requirements of fast systems. However, the effectiveness of this approach has been demonstrated only through numerical simulations. Some of optimization algorithms, such as particle swarm optimization (PSO) [46] and genetic algorithms (GA) [47], have been implemented in NMPC formulations to expedite the solving of optimization problems. These methods are coded in programming language codes to replace the *fmincon* routine in MATLAB. In addition, these algorithms were proposed to reduce the execution time of the optimization problem. Other applicable solutions to the NMPC scheme have emerged, such as The ACADO toolkit [48], Nlopt library [49], and qpOASES solver [50], which include a variety of algorithms for dynamic optimization. These libraries contain several tools to design and evaluate the NMPC controller, export low-level C/C++ code-based numerical methods, and solve optimal control problems at each real-time iteration. These methods have been designed to reduce the computation time and meet real-time control requirements of fast systems. However, despite its successful application, it is limited to a typical class of fast systems. Subsequently, the standard NMPC strategy is not yet suitable and can be widely used for real-time online optimization with a low sampling time. Fast NMPC schemes to control more complex systems, such as PKMs, were developed in this study.

3.3. Challenges and Limitations Regarding Fast Systems

This study investigates the development of a fast NMPC for PKMs. The main associated challenges of this study are outlined below:

- How we can reduce the computation time of a standard NMPC controller without using any linearization methods is the first challenge. Therefore, the applicability and relevance of the NMPC strategy in the field of robotics [51] and [52] can be confirmed, such as parallel manipulators.

- As PKMs belong to nonlinear systems, their control is often described in the literature as a nontrivial task. Then, another challenge is how to find a compromise between model complexity, control design, and real-time constraints for fast parallel robots based on NMPC schemes.

Owing to these issues, we propose the development of an extended fast NMPC controller for PKM robots. This includes design, implementation, and experimental validation. A major outcome of the proposed strategy is to achieve real-time control requirements and enhance global tracking performance.

4. Proposed Control Solutions for VELOCE PKM

The motivation behind the development of the extended fast NMPC is to improve the computation time of standard NMPCs while satisfying the control objectives related to PKMs. To reduce the computational time and enable online implementation of the NMPC strategy for PKM robots, several optimization techniques have been tested, such as the Nlopt library [49] and the control-updating period method [18]. However, these solutions cannot solve real-time control problems of VELOCE parallel robot because of its highly nonlinear and complex dynamics. In the following sections, two contributions of nonlinear model predictive control firmware are proposed for the real-time control of PKMs.

4.1. Contribution 1: Extended Fast NMPC Controller

The proposed strategy is based on an exponential parametric approach and a proportional integral derivative (PID) control term. To improve the overall performance of the system, a control law was formed using two terms. The first is computed by a simple PID feedback, and the second is given by the fast NMPC controller based on a PKM dynamic model.

In the literature, MPC has mainly been combined with PID controller into two ways. The control architecture in the first one is based on a cascaded structure [53] [54] [55] [56] [57], including two control loops (inner and outer), one for the MPC and one for PID.

In the second way the MPC and PID controllers are combined in a parallel structure as in the works of [58] [59] [60] [57], where two parallel control loops are considered. This combination has proved its efficiency in terms of closed-loop performance with respect to the standard MPC scheme alone, especially in the presence of unmodelled dynamics, parametric and modelling uncertainties as well as external disturbances. In our case, taking account the targeted application of high-speed PKMs as well as the associated real-time constraints, implementation issues, and challenges, including (i) the highly nonlinear dynamics of the robot, (ii) the uncertain nature of the system as well as its environnement, and (iii) the unexpected internal and external disturbances, we opted for the parallel control structure. The basic idea of the proposed parallel structure is as follows:

At each sample time the two control terms (from Fast NMPC and from PID) are computed. In the objective of tracking of the reference trajectories, the fast NMPC, based on the Nonlinear plant model, predicts the future behavior of the robot over a finit time horizon and compute an optimal control sequence by resolving an open-loop optimization problem while ensuring satisfaction of

the given system constraints. The PID control term is computed based on the tracking error between the reference and the measurements, as well as its time derivative and its integral. Once both control terms are computed, they are summed and sent to the actuators of the robot. After one sample time the measurements are fed back to both controllers (Fast NMPC and PID) and the horizon is shifted and the whole procedure is repeated. It is worth to note that this feedback of the measured outputs resulting from the application of a control input combining Fast NMPC and PID represents a closed loop where the fast NMPC will take into account indirectly the participation of the PID controller and vice versa through the measured outputs. This feature can be explained by the robustness of the NMPC controller, that compensates for the uncertainties or unmodelled dynamics (both not present in the prediction model), through the closed-loop resulting from the feedback of the measured outputs which differs from the predicted ones. While time going ahead the system outputs will converge to their desired trajectories.
Besides, the proposed parallel structure has also another advantage as a fault-tolerant control. For instance in the case of a remote control, the loss of the data of one controller does not necessarily lead to the loss of system's control since the other controller, thanks to the closed-loop and the feedback of the measured outputs, can compensate for the lost controller and ensure the continuity of the control.
Finally, in terms of design, the PID controller parameters (K_p, K_I, K_D) and the fast NMPC parameters (N_p, N_c, λ) are tuned sequentially. The control loop of the PID is firstly designed in order to obtain a stable closed-loop system and its design parameters are tuned to get the best closed-loop performance. Then the fast NMPC is secondly designed using the nonlinear plant model.

To provide an overview of the proposed solution, a block diagram summarizing the first contribution as an extended fast NMPC-PID control solution (Ex-Fast NMPC-PID) is depicted in Figure 4, where both control actions are applied to the real process and the resulting output is simultaneously fed back to both NMPC and PID controllers. As only the joint positions q are measured, the desired trajectory X_d in Cartesian space is converted into its equivalent in joint space through the inverse kinematics of the robot. Subsequently, the corresponding tracking error in joint space can be obtained. Note that the Runge-Kutta of the 4th-order method is used to solve the differential equation of the robot dynamics. This method provides a good compromise between performance and precision.

The proposed extension of NMPC should preserve all its advantages while improving the overall closed-loop behavior and considering the high-speed features of the manipulator, with a focus on tracking performance. The extended fast NMPC control law with feedback gains is expressed as follows:

$$\Gamma = Sat_{\Gamma_{min}}^{\Gamma_{max}}\Big(\Gamma_p+\alpha_1(p)e^{-\lambda T_e}+\alpha_2(p)e^{-\gamma\lambda i T_e}+K_P(.)e(k) + K_I(.)(\frac{1}{1-z^{-1}})e(k)+K_D(.)(1-z^{-1})e(k)\Big) \quad (11)$$

where $i \in \{0, \cdots, N_p-1\}$. α_1 and $\alpha_2 \in \mathbb{R}^n$ are coefficients of the exponential parameterization technique. The stationary control is denoted by Γ_p, and the constants $\lambda > 0$ and $\gamma \in \mathbb{N}$ are tuning parameters. $Sat : \mathbb{R}^n \mapsto \mathbb{R}^n$ is the saturation function defined as follows for $i \in \{1, \cdots, n\}$:

$$Sat_{\Gamma_{min}}^{\Gamma_{max}}(\Gamma_i) = \begin{cases} \Gamma_{min} & \text{if } \Gamma_i \leq \Gamma_{min} \\ \Gamma_{max} & \text{if } \Gamma_i \geq \Gamma_{max} \\ \Gamma_i & \text{otherwise} \end{cases} \quad (12)$$

where $e(k) = q_d(k) - q(k)$ defines the joint tracking error between the reference trajectory $q^d(k)$ and the measured trajectory $q(k)$. We define $K_P(.)$, $K_I(.)$ and $K_D(.)$ as feedback gains of the PID controller. The feedback PID control parameters were obtained using the trial-and-error tuning method. The adaptation gain was relatively easy to tune. Mainly, we proceed by increasing the gain from zero until we reach the parameter value, resulting in a good transient behavior of the estimated parameter while preserving the stability of the closed-loop system. Owing to its adjustable gains in the error state, the PID controller is more robust against model parameter variations and perturbation rejection. For PKMs, these gains are are typically chosen as diagonal matrices, indicating that no coupling between the joints is considered. Hence, they can be expressed as follows:

$$\begin{aligned} K_P(.) &= diag\{k_{p_1}(.); k_{p_2}(.); \cdots; k_{p_n}(.)\}, \\ K_I(.) &= diag\{k_{i_1}(.); k_{i_2}(.); \cdots; k_{i_n}(.)\}, \\ K_D(.) &= diag\{k_{d_1}(.); k_{d_2}(.); \cdots; k_{d_n}(.)\} \end{aligned}$$

The n-degree-of-freedom parameter setting p is used to improve the transient behavior of the tracking error, which is denoted by exponential terms. The structure of the future control sequence based on this technique over the sampling period $[k, k+1]$ can be defined by

$$\Gamma_f(k+i) = Sat_{\Gamma_{max}}^{\Gamma_{min}}(\Gamma_p + \alpha_1 e^{-\lambda i T_e} + \alpha_2 e^{-\gamma\lambda i T_e}) \quad (13)$$

According to Equation (13), the decision control sequence Γ_f for the PKM is defined based on the control value Γ_p, which is part of the low-dimensional vector p, rather than a future control sequence, as defined in equation (19).

Based on the new low-dimensional vector $p \in \mathbb{R}^{2\times n}$, the control sequence Γ_p in Equation (11) can be defined by the latest value of the optimal control sequence, p where $\Gamma_p = [p_{n+1}, \cdots, p_{2\times n}]^T$. The parameter vector p is denoted

by $[p_1, \ldots, p_n]^T \in [-1, +1]^n$ and $[p_{n+1}, \ldots, p_{2\times n}]^T \in \mathbb{R}^n$. This constraint was selected to compute the parameters of the control law. The coefficients of the exponential parameterization technique α_1 and $\alpha_2 \in \mathbb{R}^n$ are obtained by solving a simple linear equation system resulting from replacing $i = 0$ in (13) over the sampling period $[k-1, k]$, which leads to the following constraints:

$$\Gamma_p + \alpha_1 + \alpha_2 = \Gamma_f(k-1) \tag{14}$$

Then, by replacing $i = 1$ in (13) over the sampling period $[k-1, k]$ leads to the following constraints:

$$\Gamma_p + \alpha_1 e^{-\lambda T_e} + \alpha_2 e^{-\gamma\lambda T_e} = \Gamma_f(k) \tag{15}$$

Now by considering the difference between equations (14) and (15) leads to:

$$\alpha_1(e^{-\lambda T_e} - 1) + \alpha_2(e^{-\gamma\lambda T_e} - 1) = p\delta_{max} \tag{16}$$

Equation (14) represents the first condition to guarantee the continuity of the control sequence, whereas Equation (16) defines the constraints on the variation rates of the control input $(\Gamma_f(k) - \Gamma_f(k-1)) = \delta_{max})$, which should not exceed a fraction $p \in [-1, +1]^n$ for some maximal admissible values $\delta_{max} \in \mathbb{R}^n$. The coefficients $\alpha_1 = \left[\ \alpha_1^{\Gamma_1};\ \cdots;\ \alpha_1^{\Gamma_n}\ \right]$ and $\alpha_2 = \left[\ \alpha_2^{\Gamma_1};\ \cdots;\ \alpha_2^{\Gamma_n}\ \right]$ are determined by solving the following equations:

$$\Psi(p) = M^{-1}U(p) \tag{17}$$

where

$$\Psi(p) = \begin{bmatrix} \alpha_1^{\Gamma_1}(p) \\ \alpha_2^{\Gamma_2}(p) \\ \vdots \\ \vdots \\ \alpha_1^{\Gamma_n}(p) \\ \alpha_2^{\Gamma_n}(p) \end{bmatrix},\ U(p) = \begin{bmatrix} \Gamma_1(k-1) - \Gamma_{p_1} \\ p_1\delta_{max}^1 \\ \vdots \\ \vdots \\ \Gamma_n(k-1) - \Gamma_{p_n} \\ p_n\delta_{max}^n \end{bmatrix} \tag{18}$$

$$M = \begin{bmatrix} \frac{1}{e^{-\lambda T_e} - 1} & \frac{1}{e^{-\gamma\lambda T_e} - 1} & 0 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots & \vdots \\ 0 & \vdots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \cdots & \cdots & \frac{1}{e^{-\lambda T_e} - 1} & \frac{1}{e^{-\gamma\lambda T_e} - 1} \\ 0 & 0 & \cdots & \cdots & e^{-\lambda T_e} - 1 & e^{-\gamma\lambda T_e} - 1 \end{bmatrix}$$

The optimal set value from Equation (8) is redefined by the new variable $\hat{p}$, which is obtained by minimizing the cost function J as follows:

$$\hat{p} = arg\min_p [J(p, q)] \tag{19}$$

$$J(p,q) = \sum_{i=1}^{N_p} (\hat{q}(i,q,p) - q_f(q^d, i))^2 + \lambda(\hat{q}(N_p, q, p) - \hat{q}_f(\hat{q}(N_p, q(k), p), \Gamma_f^{(N_p)}(p)))^2 \tag{20}$$

where $\hat{q}(i, q, p)$ denotes the future-predicted output vector parameterized by p over $[k, k+N_p-1]$ and starts from the joint position q. Further, $q_f(q^d, i)$ represents the filtered version of the desired trajectory enabling to decouple the selling time from the overshoots. It is defined by

$$q_f(q^d, i) = q^d + e^{-3T_e i/t_r}(q - q^d) \tag{21}$$

where T_e and t_r denote the sampling and desired setting times of the closed-loop system, respectively. Further, $\hat{q}(N_p, q(k), p)$ is the joint position value at the end of the prediction horizon, and the vector $\hat{q}_f(\hat{q}(N_p, q, p), \Gamma_f^{(N_p)}(p))$ is computed according to the joint prediction $\hat{q}(N_p, q, p)$ and the control signal Γ_f obtained from Equation (13).

To solve the real-time problem of high-speed PKMs, it is worth understanding that simply changing the overall shape of the NMPC approach based on the exponential parametrization technique is not sufficient. Consequently, the fast gradient method introduced in [61] is used to compute the parameterized control signal p with reduced computation time when compared to that of conventional solvers. The proposed optimization problem solver was executed in the millisecond range, as described in Algorithm 1. The following four-step procedure in the algorithm is used to solve Equation (19) to obtain the best optimized control parameters for the nonlinear dynamic model. It is worth to note that in this algorithm, the parameter β is initialized by the designer and then tuned according to Algorithm 3. Besides, for the calculation of $\nabla J(p(k), q(k))$, depending on the implemented control scheme, $J(p(k), q(k))$ is computed using formula (19) or (24).

Algorithm 1 Fast gradient solver

Require: $\beta \in [0, 1[, L \geq 0$

Ensure: $p(k) \in \mathbb{R}^n$

1: Initialize $\tilde{p}(k) \leftarrow p(k)$

2: $\tilde{p}(k+1) = p(k) - \frac{1}{L}\nabla J(p(k), q(k))$

3: $p(k+1) = \tilde{p}(k+1) + \beta(\tilde{p}(k+1) - \tilde{p}(k))$

4: Send the obtained control variable p to the parameterized NMPC block at each instant k.

Motivated by the results obtained in [15, 62, 63, 17], where the computation time of the conventional NMPC controller was enhanced using a parameterization technique and a standard optimization toolbox, the proposed controller in Equation (11) is expected to provide better tracking performances for PKMs. By using the exponential parameterization technique and the fast gradient method, as well as the using of the PID feedback control

term, the compensation of the model nonlinearities and real-time control of high-speed PKMs can be achieved.

4.2. Contribution 2: ANN-Based Fast NMPC Controller

To improve the overall closed-loop performance of the NMPC scheme for parallel manipulators, a combination of parameterized techniques based on an ANN approximated model may yield better tracking performance and reduced control effort. The proposed a predictive law with a PID control term that was sufficiently simplified to be implemented in PKMs. A block diagram of the extended fast NMPC based on PID control scheme (Ex-Fast NMPC-PID) is depicted in Figure 5. To improve tracking performance of the 4-DOF VELOCE PKM, the enhanced fast NMPC control law with feedback gains can be expressed as follows

$$\Gamma = \Gamma_{f_2} + K_P(.)e(k) + K_I(.)(\frac{1}{1-z^{-1}})e(k) + K_D(.)(1-z^{-1})e(k) \quad (22)$$

where Γ_{f_2} denotes the fast NMPC control term given by Equation (25). Further, K_P, K_I, $K_D \in \mathbb{R}^{n\times n}$ are positive definite matrices that represent the feedback gains of the PID control term. The PID feedback gains were evaluated using a trial-and-error tuning method. Owing to its adjustable gains with the error state, this controller is more robust against model parameter variations and perturbation rejection.

Note that no universal parameterization can be applied to any system. However, the parameterized technique used in the design of this control strategy is different from the above exponential method and that introduced in [28]. The low-dimensional optimization problem using this particular parameterization of the control sequence Γ_{f_2} is defined as follows:

$$\Gamma_{f_2} = [\Gamma_{f_2}(k,p), \Gamma_{f_2}(k+1,p), \cdots, \Gamma_{f_2}(k+N_c-1,p)] \quad (23)$$

where $\Gamma_{f_2}(k,p)$ is the control vector at time instant k and p is the low-dimensional parameter vector.

Hence, the optimal parameter set $\hat{p}$ from Equation (19) is solved using the fast gradient method of the parallel manipulator from Algorithm 1. In this case, the optimization problem is obtained by minimizing the cost function $J(.)$ chosen as follows :

$$J(k,\Gamma_{f_2}) = \sum_{i=1}^{N_p}(q_f(q^d,i,k) - \hat{q}(i,q,k))^2 + \lambda\sum_{j=1}^{N_c}(\Gamma_{f_2}(j,k,p) - \Gamma_{f_2}(j,k-1,p))^2 \quad (24)$$

Note that the control signal is represented by $\Gamma_{f_2}(j,k,p) \in [\Gamma_{min}, \Gamma_{max}]$, where Γ_{min} and $\Gamma_{max} \in \mathbb{R}^4$, for $j = 1, 2, \cdots, N_c$. q_f defines the filtered version of the desired trajectory as denoted in equation (21), and $\hat{q}(i,q,p)$ represents the future output predictor vector. In this context, to obtain a low-dimensional optimization problem decoupled from the choice of the predictive horizon, the following parameterization is proposed:

$$\Gamma_{f_2}(j,k,p) = \Gamma^*(j,k) + pE^*(j,k), \text{ for } j \in 1, 2, \cdots, N_c \quad (25)$$

where Γ^* is the steady-state control action obtained in the stationary regime at time instant k and $E^*(k)$ is the trajectory tracking error obtained using $\Gamma^*(k)$. In this case, the parameterized control action is $p \in [-1, 1]$. We note that, because the system is open-loop stable, steady control Γ^* is used in the parameterization of the control profile to reduce the dimension (number of degrees of freedom) of the control signal.

The PKM prototype used for the real-time experimental validation of the nonlinear MPC is characterized by a sampling time of 0.1 ms. The computation time of the NMPC control scheme must be executed in the millisecond range on this type of high-speed robots. Consequently, even when a coding NMPC controller in the low-level C/C++ programming language to reduce the computational time, it does not solve the real-time control problem of VELOCE parallel robot because of its highly nonlinear and complex dynamics. In practice, where a robot should be in an industrial environment, executing various tasks and matrices $M(.)$ and $N(.)$ are likely to vary depending on the task being executed. Furthermore, the robot manipulator may be subject to various external disturbances, uncertainties, and unmodeled dynamics (e.g., friction and interaction with the environment). Hence, it is important to design a control law that considers all of these variations and uncertainties. This concept motivates the development of a second proposed solution to control more complex systems, such as PKMs based on an ANN internal model. An ANN is used to train the dynamic model of the PKM robots to reduce the computational time and enable online implementation of the NMPC strategy. Thus, for the second proposed solution, we developed an extended fast NMPC controller, including a control-based parameterized term, coupled with an ANN internal model. Based on the prediction-optimization principle, the proposed strategy uses the ANN-based approximation model of the system to predict future outputs and synthesize the control law. The prediction sequence generated by the ANN model was exploited directly by the NMPC algorithm, and no other model was used for the nonlinear MPC. The proposed approach involves computing an optimal control sequence by solving online a nonlinear optimization problem based on a fast gradient solver. This strategy is referred to as receding horizon control because the control problem is solved over a future horizon, which progressively shifts into the future as time evolves. Based

on the experimental results, the proposed controller can perform better than simply computing the control input at each time sample. Therefore, an optimal control sequence that can track the reference trajectories and enhance the closed-loop performance, despite the eventual errors in the identification of the ANN approximating the model, can be generated. Accordingly, the motivation behind this experimental result is to demonstrate the robustness of the proposed controller and to assess the tracking capabilities with an efficient computational time on VELOCE parallel robot while considering real-time constraints. In particular, the addition of more knowledge of dynamics can improve tracking performance and help avoid high-frequency control efforts. ANNs have the ability of modeling complex nonlinear systems while considering all possible interactions between input variables. An ANN is characterized mainly by its powerful universal approximation features [64], which is why an ANN was proposed in this study to approximate the dynamics of a parallel manipulator. The approximated model is expected to predict the future process behavior over a prediction horizon with reduced computational burden, compared to the original analytical model given by Equation (7). Because there is no systematic method for determining the optimal ANN topology, the number of layers and neurons is often determined using a trial-and-error method. They were selected by conducting a grid search where each model performance was analyzed and compared. Within a set of models that perform well, we finally chose the best model meeting the computational requirements of real-time implementation on the real robot. In this study, a feedforward neural network was selected after comparing different architectures. A fully connected three-layer feedforward network was considered (Figure 6). The input layer comprises four inputs

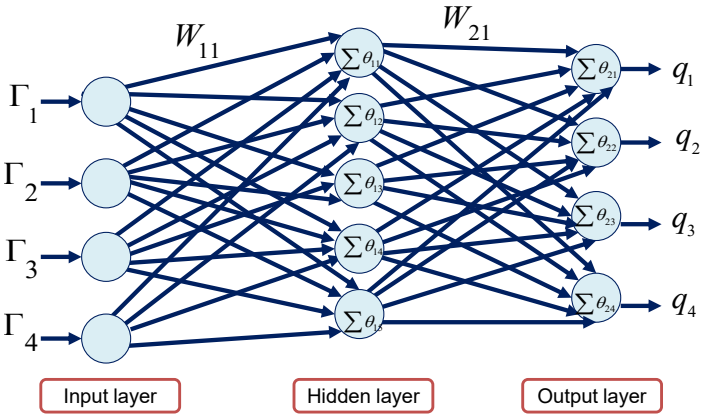


Figure 6: View of the topology of the proposed artificial neural network (ANN) architecture for the 4-DOF parallel robot.

Γ=($\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4$), representing the four actuators of VELOCE parallel robot. The output layer of the ANN provides the articular positions of the robot $q = (q_1, q_2, q_3, q_4)$ using four neurons. The hidden layer of the proposed ANN included five neurons with tangential sigmoid activation functions. The linear activation function is used as the output layer. Three formulations were considered to describe the network output signals.

- ANN outputs are expressed by

$$\hat{q}_j(k) = \sum_{i=1}^{n} W2_{i_j} \Theta_{1_i}(k) \;\; j = 1, \ldots, 4 \tag{26}$$

- Hidden layer outputs

$$\Theta_{1_j}(k) = f(\sum_{i=1}^{N_1} W1_{i_j} \Gamma_i(k)) \;\; j = 1, \ldots, n \tag{27}$$

are N_1 and n is the number of neurons that comprise the input and hidden layers, respectively. Further, $W_{(.)}$ is the weight vector. The weights were updated as follows:

$$W_{ij}(k+1) = W_{ij}(k) - \mu \frac{dJ}{dW_{ij}(k)} \tag{28}$$

The performance criterion J is given by

$$J = \frac{1}{2}(q - \hat{q})^T (q - \hat{q}) \tag{29}$$

There are several suitable activation functions; however, the most typical sigmoid function was used in this study:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{30}$$

This unipolar activation function produces an output between zero and one. Once the parameters of the ANN model are computed using the LSM algorithm, the next step is to use the developed approximation model in the development of the enhanced NMPC strategy. The input-output data used in the off-line training process of the network have been generated using the nonlinear plant model.

Remark: The proposed extended fast NMPC controller has been developed and implemented in discrete time. Accordingly, the proposed ANN is fully compatible and can be naturally integrated into the proposed NMPC controller of Contribution 2 (ANN-based fast NMPC controller). The optimization problem of the latter is formulated in discrete time, and the proposed ANN model is recursively used to compute the optimal future control sequence.

The proposed control law in Equation (22) comprises two distinct control elements. The first element is the model-based predictive term, which is designed based on a fast gradient solver, particular parameterization technique, and neural network based on an internal model. The second one lies in the feedback gains defined in the PID control term. The motivation behind the proposed strategy is to achieve a good tracking performance. The main reason for the proposed solution was to improve the performance of the standard NMPC strategy for PKMs. The proposed enhancement to the NMPC scheme should preserve all its ad-

vantages while improving the overall closed-loop behavior using the ANN model, with a greater focus on tracking performance. Another outcome that can be achieved using the proposed scheme is diminution computation time, allowing real-time application of the proposed control solution. Finally, compared with [16], [28], [29], [14], [13] and [30], the proposed enhanced fast NMPC controller may improve the robustness of the resulting closed-loop system and is suitable for real-time implementations of robotic systems. The main contribution of the present work, compared to existing schemes, lies in the use of various forms of parameterization techniques and a fast solver method in the development of a fast NMPC extension theory, particularly for complex high-speed parallel robots. To demonstrate the relevance of the proposed control strategy, we numerically simulated and experimentally validated the proposed extended fast NMPC on VELOCE PKM.

5. Simulation and Experimental Results

This section mainly aims to validate the proposed enhanced nonlinear model predictive control online on a high-speed parallel robot. The desired trajectory to be tracked by VELOCE robot is a sequence of point-to-point motions with a duration of T= 0.5 s for each portion of the trajectory. The corresponding Cartesian reference trajectory is illustrated in a 3D view in Figure 7.

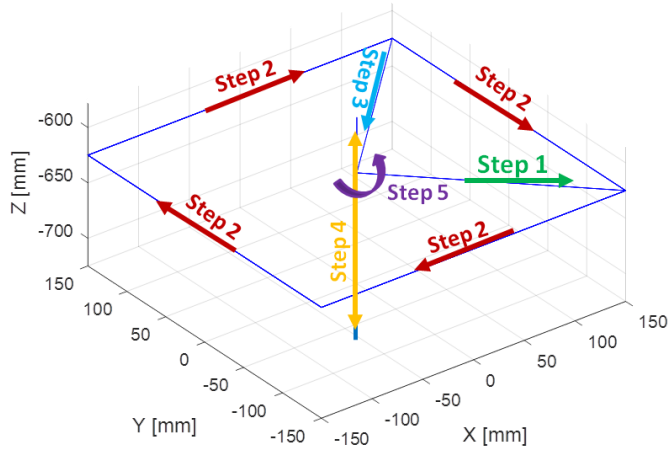


Figure 7: 3D view of the point-to-point desired Cartesian trajectory for VELOCE parallel robot.

To quantify the control performance of the proposed extension of the fast NMPC control scheme, we introduce the following root mean square tracking error (RMSE)-based criterion:

$$RMSE_J = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{4} e_{q_j}^2(i)} \tag{31}$$

$$RMSE_T = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(e_x^2(i) + e_y^2(i) + e_z^2(i))} \tag{32}$$

$$RMSE_R = \sqrt{\frac{1}{N}\sum_{i=1}^{N} e_\alpha^2(i)} \tag{33}$$

where $RMSE_J$ represents the RMSE criterion based on the joint tracking errors and e_{q_j} $j = 1, ..., 4$ denotes the tracking error of the j^{th} joint. Further, $RMSE_T$, $RMSE_R$ is the RMSE criterion based on the computed Cartesian tracking errors. Her, the translational movements are separated from the rotational ones; N denotes the number of recorded samples; e_x, e_y and e_z define the tracking error along x, y, and z axes, respectively; and e_α is the tracking error.

5.1. Tuning of the Control Gains

A well-known strategy for tuning the control gains in experiments used for complex robotic systems is the trial-and-error method. It is characterized by manually and continuously attempting different sets of control gains in a real-time framework until the desired control performance is achieved. It is generally used when the formulated dynamic model does not exactly match the physical system. Thus, automatic numerical closed-loop tuning methods may provide unsuitable control gains for real-time experiments.

The tuning process of the proposed controllers is characterized by reduced computation time, as presented in the following subsection (algorithms 2 and 3). It is worth to note that depending on the target application and the needed performance, it is the designer who should decide about the quantitative values for the guidance of the tuning process while taking into account the specific real-time constraints of the actual system.

5.1.1. Tuning of the parameterized NMPC control gains

The process of tuning the standard parameterized NMPC control gains is achieved using Algorithm 2.

5.1.2. Tuning of the proposed extended fast NMPC-PID control gains

The main tuning procedure used for the proposed extended fast NMPC-PID control gains is summarized in the following steps in Algorithm 3.

5.2. Numerical Simulation Results

To demonstrate the relevance and effectiveness of the proposed control solutions, we begin with a numerical simulation and comparative analysis with the parametrized

Algorithm 2 *Tuning algorithm of the parameterized NMPC control gains*
Involved calculations:(11),(13),(17)–(20),(30)–(32)

1: *Set* $N_p = 1$, $N_c = 1$, $\lambda = 0$ *and* $\gamma = 0$,
2: *Increase the value of* λ, *until a satisfied tracking is reached,*
3: *Tune* N_p *and* N_c *by increasing till we reach as best performance index as possible.*
4: *Start increasing the value of* γ, *until obtaining less chattering input signal and better performance index and modifying again* λ *either increasing or decreasing to ensure a smooth control signal.*

Algorithm 3 *Tuning algorithm of the proposed extended fast NMPC-PID control gains*
Involved calculations:(19),(22)–(25),(30)–(32)

1: *Initialization:* $\beta = 0$, $L = 0$, $\lambda = 0$, *and* $\gamma = 0$, $N_p = 1$, $N_c = 1$, $K_P = 0$, $K_I = 0$ *and* $K_D = 0$,
2: *Start increasing* λ, N_p *and* γ *until we obtain acceptable tracking performance,*
3: *Increase the value of* N_c *and returning again the values of* λ, N_p *and* γ *either increasing or decreasing until a smooth control signal is obtained and a trade-off between computational time and sufficient insight into the system behavior is satisfied,*
4: *Tune* β *within the interval* $[0, 1[$ *and increase* L *until guaranteed* $J(p^{i+1}) < J(p^i)$. *The weighting factor* λ *is decreased until the convergence of the system response is faster*
5: *Tune the values of* K_P *and* K_D *gains that are started from zero until acceptable tracking performance is achieved. Then, increase the value of* K_I *until the static error increases.*
6: *Repeat steps (4) and (5) until obtaining the best possible closed-loop performance.*
7: *Decrease* L *until we get a better performance index.*

NMPC controller introduced in [15]. Hence, an extended fast NMPC control scheme is designed based on a fast gradient solver (see Algorithm 1) and an exponential parameterized NMPC approach, which is introduced in Equation (13). The proposed controller was developed with a reduced computation time (up to 0.0781 s) compared to the standard NMPC which is greater than 117.2430 s, and the parametrized NMPC is up to 104.4537s. The control architecture was implemented using the Matlab/Simulink tool running on Intel Core i7 processor. The obtained results are depicted in Figure 8, where the robot's platform starts from an arbitrary position to join the desired initial position, located at (0,0,-625,0) mm, then moves from this initial position to the final Cartesian position located at (0,0,-625,300) mm, and finally returns to the initial position (Figure 7). This simulation scenario aims to assess the tracking capabilities in terms of Cartesian and articular tracking errors and to show that the developed controller can ensure a satisfactory computation performance, as required by VELOCE PKM, without any linearization techniques. This numerical simulation provided a good overview of the real-time feasibility of the proposed controller. For comparison, a parameterized NMPC technique was introduced in [15] based on the fmincon MATLAB function, and the proposed extended fast NMPC (Ex-Fast NMPC) control schemes were implemented in a discrete-time scheme similar to real-time robot control. The simulation results obtained for the controllers are shown in Figure 9, while the control parameters are reported in Table 2. These control parameters were tuned such that each controller could provide the best performance. From Figure 9, it is clear that the proposed extended fast NMPC controllers provide significantly better tracking performance compared to the standard parameterized NMPC control strategy. To quantify the enhancement caused by the proposed extended fast NMPC, the RMSE-based criteria introduced above were evaluated for each controller and are summarized in Table 3. Clearly, the tracking improvement of the proposed controller is significant (up to 70.91% for translation and up to 76.91% for rotation) compared with the standard one. The evolution of The control input torques generated by both the controllers are shown in Figure 10. Clearly, the generated control signals for both controllers are in admissible ranges and do not exhibit any high-frequency components. Note that the computation time (Figure 11) obtained by the proposed Ex-Fast NMPC (around 0.0631 with a maximum of 0.0781 s) is faster than the parameterized NMPC controller (around 50 s with a maximum of 104.453 s). Indeed, the computational burden is substantially reduced owing to the optimization problem of low dimensions obtained by using the parameterized approach and the fast gradient solver. Consequently, our results can be easily applied to other types of fast mechatronic/robotic systems after modeling the of their nonlinear dynamics. However, this reduced computation time is not sufficient for the real-time application of the proposed controller in the case of thw VELOCE high-speed PKM. Therefore, two solutions are proposed in this section to demonstrate the improvement steps of the proposed NMPC control strategy for handling high-speed parallel robots. The proposed controller is enhanced by adding feedback gains that are adequate for the experimental implementation on VELOCE parallel robot and compared it with a PID regulator. The experimental results are presented and discussed in the following subsec-

tions.

Algorithm 4 Proposed extended Fast NMPC-PID control firmware (Contribution 1)

1: Initialize the maximum number of iterations value, the parameters of the fast gradient solver, the prediction horizon N_p, control horizon N_c, feedback gains (K_P, K_I and K_D), and weighting factor λ.
2: Define the matrix M, set the reference trajectory (q^d), and fix the control/output initial values. We define the parameterized vector and iteration counter ($k = 0$).
3: Acquisition of the joint position vector q at iteration k.
4: Compute the low-dimensional vector p using Algorithm 1.
5: Fined the coefficients of the parameterized technique α_1 and α_2.
6: Compute the future control sequence Γ_f of the extended fast NMPC from equation (13) and the first element of control.
7: Compute the control law Γ of the proposed extension of the fast NMPC controller with feedback gains by using equation (11).
8: Send the control actions to the VELOCE actuators.
9: Waiting the end of the sampling period.
10: Increment the iteration counter k.
11: If the iteration count k has exceeded the limit, stop. Otherwise, set $k = k+1$, return to step 3 and continue the iteration process.

5.3. Experimental Results on VELOCE parallel Robot

The VELOCE PKM prototype used for the real-time experiments in the present study is depicted in Figure 12, which is an experimental testbed for the validation of the proposed fast NMPC controller (Contribution 1) and its augmented version with the ANN model (Contribution 2). The global design of the parallel manipulator is capable of reaching 10 m/s of maximum velocity, 20 G of maximum acceleration and is able to carry a maximum payload of 10 Kg. A fixed-step solver was chosen with a sample time of 0.1 ms. The control schemes were implemented using Simulink from Mathworks and compiled using the XPC/Target and real-time toolbox. The generated code was uploaded and executed on a target PC, an industrial computer running at a frequency of 10 KHz (i.e., a sampling time of 0.1 ms).

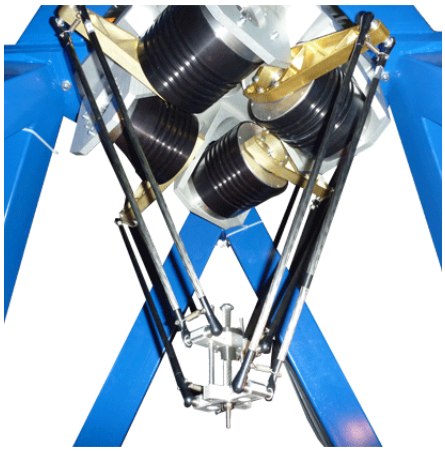


Figure 12: View of VELOCE parallel robot used for real-time experiments.

Algorithm 5 Proposed enhanced Fast NMPC-PID control firmware (Contribution 2)

1: Initialize the maximum number of iterations, the parameters of the fast gradient solver, the prediction horizon N_p, control horizon N_c, feedback gains (K_P, K_I and K_D), and weighting factor λ.
2: Define the ANN internal model, set the reference trajectory (q^d), fix the initial values of the control output vectors, parameterized vector, and iteration counter ($k = 0$).
3: Acquisition of the joint position q at iteration k.
4: Compute the low-dimensional vector p by minimizing the cost function (24) based on the fast gradient Algorithm 1.
5: Find the coefficients of the specific parameterized technique, and compute the future control sequence Γ_{f_2} of the extended fast NMPC from equation (25) and the first element of control.
6: Compute the control law of the enhanced fast NMPC controller with feedback gains by using equation (22).
7: Apply the control to the VELOCE robot.
8: Waiting the end of the sampling period.
9: Increment the iteration count k.
10: If the iteration count has exceeded the limit, stop. Otherwise, set $k = k + 1$ and proceed to Step 3.

5.3.1. Experimental results of contribution 1 on VELOCE PKM

In the following, we provide details and discuss the experimental results obtained from the proposed fast NMPC

controller for VELOCE robot. The control firmware described in Algorithm 4, which allows the determination of the control law, was used. To highlight the outcomes of the proposed controller, its performance was compared to that of a PID controller. The control parameters of proposed controller are summarized in Table 7. Following the reference trajectory illustrated in Figure 7, the Cartesian and joint tracking errors of both controllers are shown in Figures 13 and 14. The obtained results based on the proposed performance evaluation criteria are presented in Table 4. We can notice that the proposed extended fast NMPC control strategy with PID feedback gains reduces the tracking errors in all axes and providesa better results. We can then observe that the general tracking errors of the proposed controller are better than those of the standard PID controller. For instance, there was an improvement of 14.95% in terms of the tracking precision and 12.39% in terms of the rotation coordinate of the traveling plate. The control input torques of the four direct-drive motors for both controllers are shown in Fig. 15. Clearly, both control algorithms generated input signals within the admissible limits of each actuator (i.e., 127 N.m). According to the obtained results, we can conclude that the extended fast NMPC-PID controller outperforms the standard PID in terms of precision and performance owing to the exponential parameterized control technique, fast gradient solver, feedback gains, and their specific behavior. The proposed controller can ensure the stability and improve the tracking capability of high-speed manipulators.

5.3.2. Experimental results of contribution 2 on VELOCE PKM

To demonstrate the relevance of the second contribution, a real-time VELOCE PKM testbed was required. In this experimental case study, the same desired trajectory (illustrated in Figure 7) is considered. The second proposed extended fast NMPC-PID controller is based on a specific parameterization technique, fast gradient algorithm, and internal ANN model. The last model was used to predict the future behavior of the parallel robot. It has reduced computation time with a greater prediction horizon ($N_p = 3$) than that used in the first control scheme ($N_p = 1$). The proposed strategy is a potential candidate for the real-time control of parallel robots. The motivation behind this scenario was to test the robustness of the proposed controller and assessed its tracking capabilities with an efficient computational time for VELOCE robot. To use the proposed controller software, some steps were considered, as illustrated in Algorithm 5. To demonstrate the relevance of the proposed contribution and highlight its benefits, it was implemented and compared with a standard PID controller in real time on VELOCE PKM. It is important to emphasize that, as this nonlinear control solution takes Considering the high-speed dynamics of VELOCE robot, its performance is expected to have reduced computation time. the control parameters of the proposed controller are summarized in Table 5. The Cartesian tracking errors of both the controllers are shown in Figure 16. It can be clearly observed that the proposed Ex-Fast NMPC-PID controller provides better results than the PID controller. The RMS-based criteria introduced above were used to evaluate the tracking performance of both controllers. The obtained results are summarized in Table 6, from which it can be seen that the proposed controller improves the tracking capabilities of the robot manipulator significantly with respect to the PID controller (more than 43 % for the rotation DOF). As expected, a significant improvement of 46.48 % in terms of Cartesian space accuracy was observed when evaluating the RMSE performance index, and 46% for the joint space accuracy. This result highlights the benefits of using the dynamics of the manipulator in control design in terms of tracking performance. The torques generated by both controllers are depicted in Figure 18. Evidently, these torques are far from the admissible limits of the actuators, that is, 127 Nm. Moreover, they were smooth and did not exhibit any discontinuities. The movements of the robot while tracking the reference trajectories have a computation time in the interval [0.000056, 0.000077]s, which is significantly reduced compared to the numerical simulation case (up to 0.0313 s), showing the real-time applicability of the Ex-Fast NMPC-PID control scheme. The experimental results show that the extended fast NMPC controller can ensure precise tracking of the reference trajectory with a small computation time compared to the classical NMPC, which is not suitable for the control of fast systems.

6. Conclusion and Future Works

This study addressed the problem of controlling high-speed PKMs manipulators. One of the main objectives of this study is to achieve real-time control of PKMs without any linearization techniques based on the NMPC control strategy. The proposed solution is a new extension of the fast NMPC controller that includes two variants. The motivation behind this proposition is to improve the tracking capability of parallel manipulators. Real-time experiments on the 4-DOF non-redundant VELOCE robot were conducted to validate the proposed control schemes. The experimental results show that the proposed extension significantly enhances the closed-loop performance. Compared with the standard NMPC, the proposed controller reduces the computation time and allows real-time applicability to fast dynamical systems. In addition, the extended fast NMPC controller exhibits considerably better performance than the standard PID controller.

Despite all the advantages and the reached breakthrough of the proposed contributions, they have few limitations that may be considered in future directions, such as robustness towards noise and the off-line training of the ANN. Furthermore, this work can be extended by studying other control approaches for the proposed NMPC technique and their optimization. Other fast-solving methods can improve the performance of such control laws.

More extensive results for the proposed nonlinear control schemes for other industrial applications will be addressed in the future.

References

[1] O. L. Mangasarian, Nonlinear programming, SIAM, 1994.

[2] J. Rust, Numerical dynamic programming in economics, Handbook of Computational Economics 1 (1996) 619–729.

[3] I. Jammeli, A. Chemori, S. Elloumi, S. Mohammed, An assistive explicit model predictive control framework for a knee rehabilitation exoskeleton, IEEE/ASME transactions on Mechatronics-DOI: 10.1109/TMECH.2021.3126674.

[4] G. Poulin, A. Chemori, N. Marchand, Minimum energy oriented global stabilizing control of the pvtol aircraft, International journal of Control 80 (3) (2007) 430–442, doi: 10.1080/00207170601069505.

[5] E. F. Camacho, C. Bordons, Nonlinear model predictive control: An introductory review, in: Springer Assessment and future directions of nonlinear model predictive control, Berlin, Heidelberg, 2007, pp. 1–16.

[6] L. Grüne, J. Pannek, Nonlinear model predictive control, in: Nonlinear model predictive control, Springer, 2017, pp. 45–69.

[7] L. Dutta, D. K. Das, A new adaptive explicit nonlinear model predictive control design for a nonlinear mimo system: An application to twin rotor mimo system, International Journal of Control, Automation and Systems 19 (7) (2021) 2406–2419.

[8] T. Péni, B. Csutak, G. Szederkényi, G. Röst, Nonlinear model predictive control with logic constraints for covid-19 management, Nonlinear Dynamics 102 (4) (2020) 1965–1986.

[9] A. Chemori, M. Alamir, Multi-step limit cycle generation for rabbit's walking based on a nonlinear low dimensional predictive control scheme, Mechatronics 16 (2006) 259–277.

[10] T. Rybus, K. Seweryn, J. Z. Sasiadek, Control system for free-floating space manipulator based on nonlinear model predictive control (nmpc), Journal of Intelligent & Robotic Systems 85 (3-4) (2017) 491–509.

[11] A. J. Prado, M. Torres-Torriti, F. A. Cheein, Distributed tube-based nonlinear mpc for motion control of skid-steer robots with terra-mechanical constraints, IEEE Robotics and Automation Letters 6 (4) (2021) 8045–8052.

[12] R. M. Saback, A. G. S. Conceicao, T. L. M. Santos, J. Albiez, M. Reis, Nonlinear model predictive control applied to an autonomous underwater vehicle, IEEE Journal of Oceanic Engineering 45 (3) (2020) 799–812.

[13] P. Chaber, Fast nonlinear model predictive control algorithm with neural approximation for embedded systems: Preliminary results, in: Advanced, Contemporary Control, Springer, 2020, pp. 1067–1078.

[14] X. Wu, J. Chen, L. Xie, Fast economic nonlinear model predictive control strategy of organic rankine cycle for waste heat recovery: Simulation-based studies, Energy 180 (2019) 520–534.

[15] M. Alamir, A framework for real-time implementation of low-dimensional parameterized nmpc, Automatica 48 (1) (2012) 198–204.

[16] C. Damour, M. Benne, J.-J. Kadjo, S. Rosini, B. Grondin-Perez, Fast nmpc scheme of a 10 kw commercial pemfc, International Journal of Hydrogen Energy 38 (18) (2013) 7407 – 7413.

[17] A. Murilo, R. V. Lopes, A low dimensional parameterized nmpc scheme for quadrotors, in: 2018 5th International Conference on Control, Decision and Information Technologies (CoDIT), IEEE, 2018, pp. 58–63.

[18] M. Alamir, Monitoring control updating period in fast gradient based nmpc, in: 2013 European Control Conference (ECC), IEEE, 2013, pp. 3621–3626.

[19] A. Chemori, N. Marchand, Global discrete time stabilization of the pvtol aircraft based on a fast predictive controller, in: Proc. 17th IFAC World Congress, Seoul, South Korea, 2008.

[20] M. Taktak, A. Chemori, N. Ghommam, J. Derbel, Track following using nonlinear model predictive control in hard disk drives, in: IEEE International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 2013.

[21] P. Mannam, O. Kroemer, F. Z. Temel, Characterization of compliant parallelogram links for 3d-printed delta manipulators, in: International Symposium on Experimental Robotics, Springer, 2020, pp. 75–84.

[22] A. Martin-Parra, D. Rodriguez-Rosa, S. Juarez-Perez, G. Rubio-Gomez, A. Gonzalez-Rodriguez, F. J. Castillo-Garcia, Gateway configuration for rapid pick-and-place operations of 2-degrees-of-freedom parallel robots, Journal of Mechanisms and Robotics 13 (1) (2021) 014504.

[23] L. Bruzzone, R. Molfino, R. Razzoli, Modelling and design of a parallel robot for lasercutting applications, in: International Conference on Modeling, Identification and Control (IASTED02), 2002, pp. 518–522.

[24] H. Natalius, P. Lambert, M. K. Tiwari, L. da Cruz, C. Bergeles, Design, static and performance analysis of a parallel robot for head stabilisation in vitreoretinal surgery, in: International Workshop on Medical and Service Robots, Springer, 2020, pp. 169–179.

[25] X. Shen, L. Xu, Q. Li, Motion/force constraint indices of redundantly actuated parallel manipulators with over constraints, Mechanism and Machine Theory 165 (2021) 104427.

[26] G. Sartori-Natal, A. Chemori, F. Pierrot, Dual-space adaptive control of a redundantly actuated parallel manipulators for extremely fast operations with load changes, in: International Conference on Robotics and Automation (ICRA), St Paul, MN - USA,, 2012.

[27] J. P. Merlet, Parallel robots, Vol. 128, Springer Science & Business Media, 2006.

[28] A. Murilo, M. Alamir, D. Alberer, A general nmpc framework for a diesel engine air path, International Journal of Control 87 (10) (2014) 2194–2207.

[29] R. Kouki, A. Chemori, F. Bouani, A new fast nmpc scheme for parallel kinematic manipulators: Design and real-time experiments, in: IEEE International Conference on Signal, Control and Communication (SCC), Hammamet, Tunisia, Tunisia, 2019, pp. 69–75.

[30] S. Mathisen, K. Gryte, S. Gros, T. A. Johansen, Precision deep-stall landing of fixed-wing uavs using nonlinear model predictive control, Journal of Intelligent & Robotic Systems 101 (1) (2021) 1–15.

[31] Z. Xiao, M. Hu, D. Fu, C.and Qin, Model predictive trajectory tracking control of unmanned vehicles based on radial basis function neural network optimization, Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering.

[32] N. A. Spielberg, M. Brown, J. C. Gerdes, Neural network model predictive motion control applied to automated driving with unknown friction, IEEE Transactions on Control Systems Technology.

[33] M. T. Gillespie, C. M. Best, E. C. Townsend, D. Wingate, M. D. Killpack, Learning nonlinear dynamic models of soft robots for model predictive control with neural networks, in: Proc. IEEE Int. Conf. Soft Robot., 2018, pp. 39–45.

[34] X. Bao, Z. Sun, N. Sharma, A recurrent neural network based mpc for a hybrid neuroprosthesis system, in: IEEE 56th Annual Conference on Decision and Control (CDC), Melbourne, Australia, 2017, pp. 4715–4750.

[35] R. Winqvist, A. Venkitaraman, B. Wahlberg, Learning models of model predictive controllers using gradient data, IFAC PapersOnLine 54 (7) (2021) 7–12.

[36] J. Nubert, J. Kohler, V. Berenz, F. Allgwer, S. Trimpe, Safe and fast tracking on a robot manipulator: Robust mpc and neural network control, IEEE Robotics and Automation Letters 5 (2) (2020) 3050–3057, doi: 10.1109/LRA.2020.2975727.

[37] L. H. Cseko, M. Kvasnica, B. Lantos, Explicit mpc-based rbf neural network controller design with discrete-time actual kalman filter for semiactive suspension, IEEE Transactions on Control Systems Technology 23 (5) (2015) 1736–1753, doi: 10.1109/TCST.2014.2382571.

[38] S. Wang, T. Dragicevic, Y. Gao, R. Teodorescu, Neural network based model predictive controllers for modular multilevel converters, IEEE Transactions on Energy Conversion 36 (2) (2021) 1562–1571, doi: 10.1109/TEC.2020.3021022.

[39] H. Saied, A. Chemori, M. Bouri, M. El Rafei, C. Francis, F. Pierrot, A new time-varying feedback RISE control for 2nd-order nonlinear MIMO systems: Theory and experiments, International Journal of Control 94 (8) (2021) 2304–2317, doi: 10.1080/00207179.2019.1704063.

[40] D. Corbel, M. Gouttefarde, O. Company, F. Pierrot, Towards 100G with PKM. Is Actuation Redundancy a Good Solution for Pick-and-Place?, in: International Conference on Robotics and Automation (ICRA), IEEE, Anchorage, Alaska, United States, 2010, pp. 4675–4682.

[41] M. Bennehar, G. El-Ghazaly, A. Chemori, F. Pierrot, A novel adaptive terminal sliding mode control for parallel manipulators: Design and real-time experiments, in: International Conference on Robotics and Automation (ICRA), IEEE, Marina Bay Sands, Singapour, 2017, pp. 6086–6092.

[42] N. Guo, X. Zhang, Y. Zou, B. Lenzo, T. Zhang, D. Göhlich, A fast model predictive control allocation of distributed drive electric vehicles for tire slip energy saving with stability constraints, Control Engineering Practice 102 (2020) 104554.

[43] P. Hespanhol, R. Quirynen, A real-time iteration scheme with quasi-newton jacobian updates for nonlinear model predictive control, in: 2018 European Control Conference (ECC), IEEE, 2018, pp. 1517–1522.

[44] L. Yang, M. Yue, T. Ma, Path following predictive control for autonomous vehicles subject to uncertain tire-ground adhesion and varied road curvature, International Journal of Control, Automation and Systems 17 (1) (2019) 193–202.

[45] J. Mattingley, S. Boyd, CVXGEN: A code generator for embedded convex optimization, Optimization and Engineering 12 (1) (2012) 1–27.

[46] L. Chen, S. Du, Y. He, M. Liang, D. Xu, Robust model predictive control for greenhouse temperature based on particle swarm optimization, Information processing in agriculture 5 (3) (2018) 329–338.

[47] W. Chen, X. Li, M. Chen, Suboptimal nonlinear model predictive control based on genetic algorithm, in: 2009 Third international symposium on intelligent information technology application workshops, IEEE, 2009, pp. 119–124.

[48] S. Adhau, S. Patil, D. Ingole, D. Sonawane, Implementation and analysis of nonlinear model predictive controller on embedded systems for real-time applications, in: 2019 18th European Control Conference (ECC), IEEE, 2019, pp. 3359–3364.

[49] S. S. Oyelere, The application of model predictive control (mpc) to fast systems such as autonomous ground vehicles (agv), IOSR J. Comput. Eng.(IOSR-JCE) 16 (3) (2014) 27–37.

[50] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, M. Diehl, qpoases: A parametric active-set algorithm for quadratic programming, Mathematical Programming Computation 6 (4) (2014) 327–363.

[51] L. Rey, R. Clavel, The delta parallel robot, in: Parallel Kinematic Machines, Springer, 1999, pp. 401–417.

[52] S. Staicu, Dynamics of Parallel Robots, Springer, 2019.

[53] R. Singh, A. Sahay, K.-M. Karry, F. Muzzio, M. Ierapetritou, R. Ramachandran, Implementation of an advanced hybrid mpc-pid control system using pat tools into a direct compaction continuous pharmaceutical tablet manufacturing pilot plant, International Journal of Pharmaceutics 473 (1).

[54] M. Giacomelli, M. Faroni, D. Gorni, A. Marini, L. Simoni, A. Visioli, Mpc-pid control of operator-in-the-loop overhead cranes: A practical approach, in: 7th International Conference on Systems and Control (ICSC), 2018, pp. 321–326.

[55] L. Tang, F. Yan, B. Zou, W. K., C. Lv, An improved kinematic model predictive control for high-speed path tracking of autonomous vehicles, IEEE AccessDoi: 10.1109/ACCESS.2020.2980188.

[56] R. Singh, M. Ierapetritou, R. Ramachandran, System-wide hybrid mpcpid control of a continuous pharmaceutical tablet manufacturing process via direct compaction, European Journal of Pharmaceutics and BiopharmaceuticsDoi: 10.1016/j.ejpb.2013.02.019.

[57] L. Balbis, R. Katebi, A. Ordys, Model predictive control design for industrial applications, in: International Control Conference (ICC), 2006.

[58] J. O. Pedro, T. Tshabalala, Hybrid nnmpc/pid control of a two-link flexible manipulator with actuator dynamics, in: 10th Asian Control Conference (ASCC), 2015, pp. 1–6.

[59] L. Bauersfeld, L. Spannagl, G.-J.-J. Ducard, C.-H. Onder, Mpc flight control for a tilt-rotor vtol aircraft, IEEE Transactions on Aerospace and Electronic SystemsDoi: 10.1109/TAES.2021.3061819.

[60] M. Kumavat, S. Thale, Analysis of cstr temperature control with pid, mpc & hybrid mpc-pid controller, in: International Conference on Automation, Computing and Communication (ICACC), 2022.

[61] K. Graichen, B. Käpernick, A real-time gradient method for nonlinear model predictive control, in: Frontiers of model predictive control, IntechOpen, 2012.

[62] M. Alamir, Stabilization of nonlinear systems using receding-horizon control schemes: a parametrized approach for fast systems, Vol. 339, Springer, 2006.

[63] A. Murilo, Contributions on Nonlinear Model Predictive Control for Fast Systems, Ph.D. thesis, Institut National Polytechnique de Grenoble - INPG (2009).

[64] A. Hajian, P. Styles, Artificial neural networks, in: Application of soft computing and intelligent methods in geophysics, Springer, 2018, pp. 3–69.

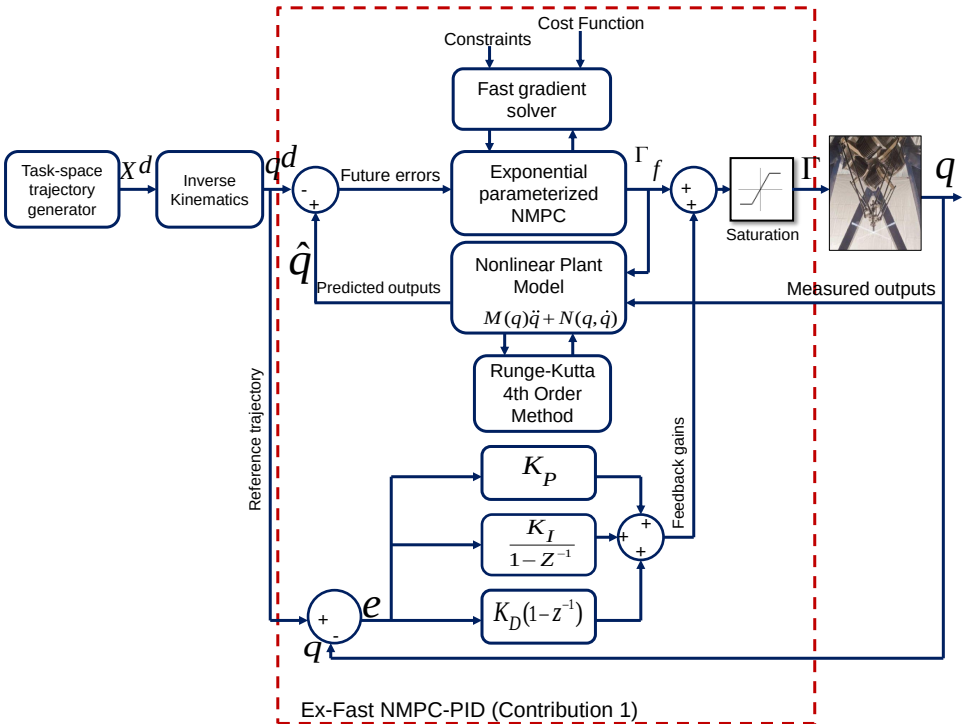


Figure 4: Block diagram of the proposed fast NMPC control scheme based on the exponential parametric approach and feedback gains for parallel robots.

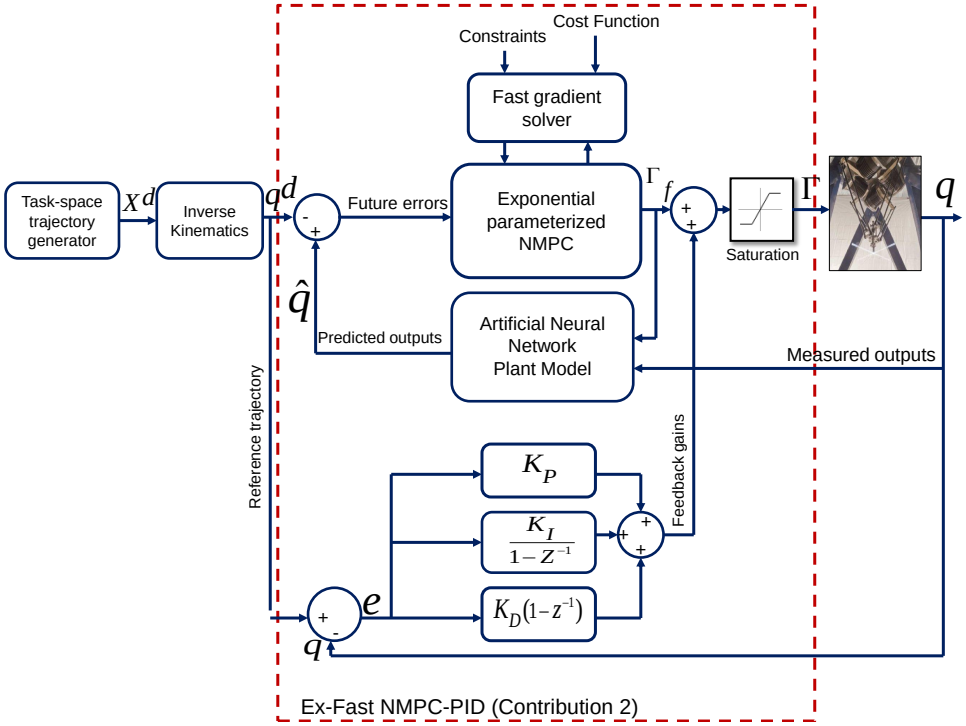


Figure 5: Block diagram of the enhanced fast NMPC control scheme based on an ANN model for parallel robots.

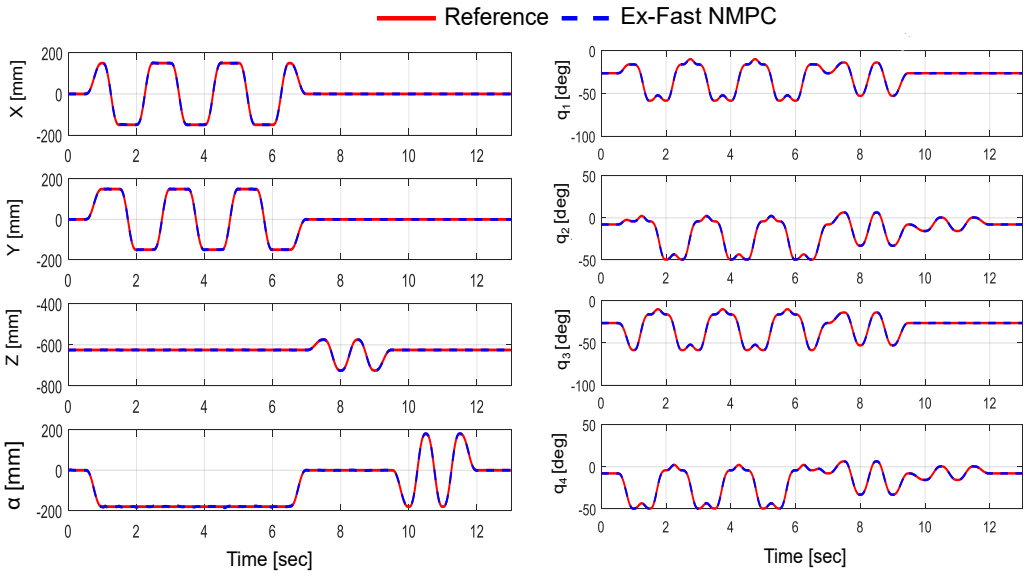


Figure 8: Trajectory tracking versus time, left: Cartesian space coordinates, right: Joint space.

Table 2: Summary of the controller parameters for the numerical simulation.

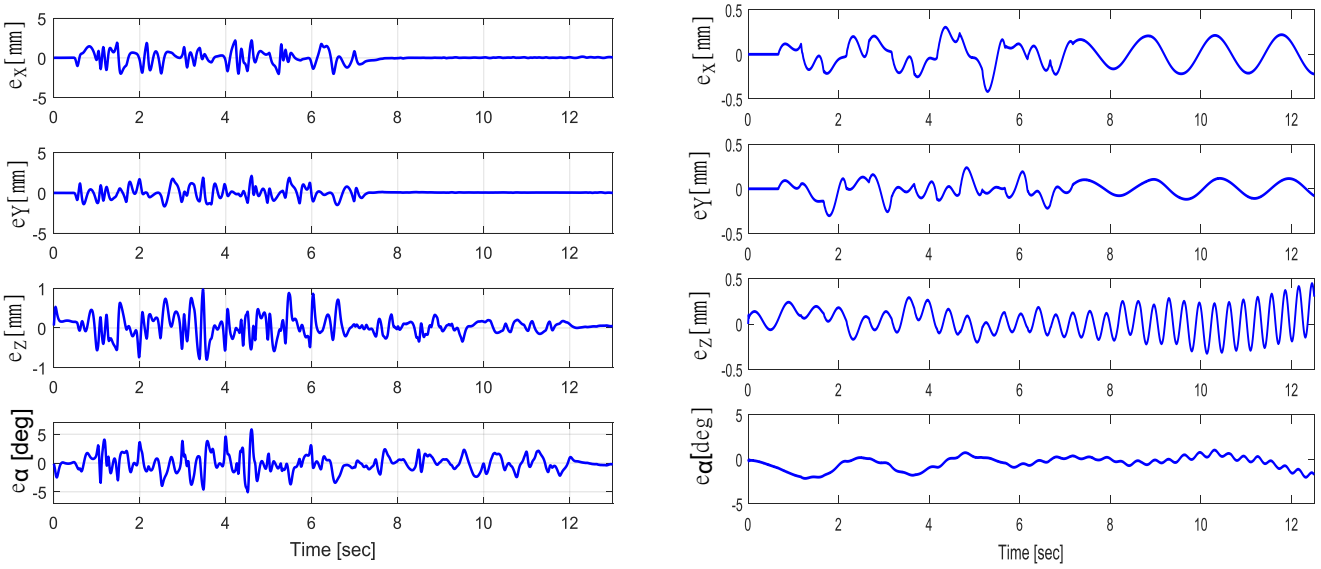
Parameterized NMPC		Ex-Fast NMPC	
Parameter	Value	Parameter	Value
		N_p	4
N_p	2	N_c	3
N_c	2	λ	180
t_r	0.054	L	1.6
λ	100	β	1
γ	4	γ	4
		t_r	0.0300

Table 3: Tracking performance comparison in the numerical simulation.

	RMSE$_T[mm]$	RMSE$_R[deg]$
Parameterized NMPC	0.8808	1.4495
Ex-Fast NMPC	0.2562	0.3416
Improvements	70.91%	76.433%

Table 4: Contribution 1: Control performance evaluation in terms of RMSE errors.

	RMSE$_J[deg]$	RMSE$_T[mm]$	RMSE$_R[deg]$
Standard PID	0.0391	0.1210	0.4275
Ex-Fast NMPC-PID	0.0339	0.1029	0.3754
Improvements	13.29%	14.95%	12.39%



(a) Parameterized NMPC (b) Ex-Fast NMPC

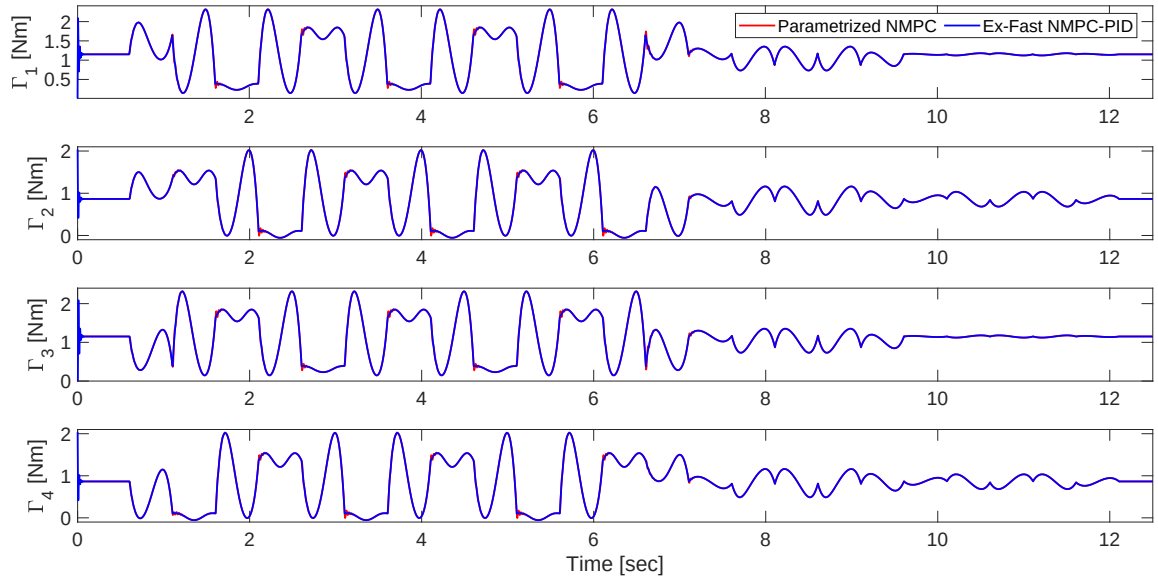
Figure 9: Evolution of the Cartesian tracking errors versus time in numerical simulation. (a) Parameterized NMPC and (b) Ex-Fast NMPC.

Table 5: Summary on the control design parameters of contribution 2.

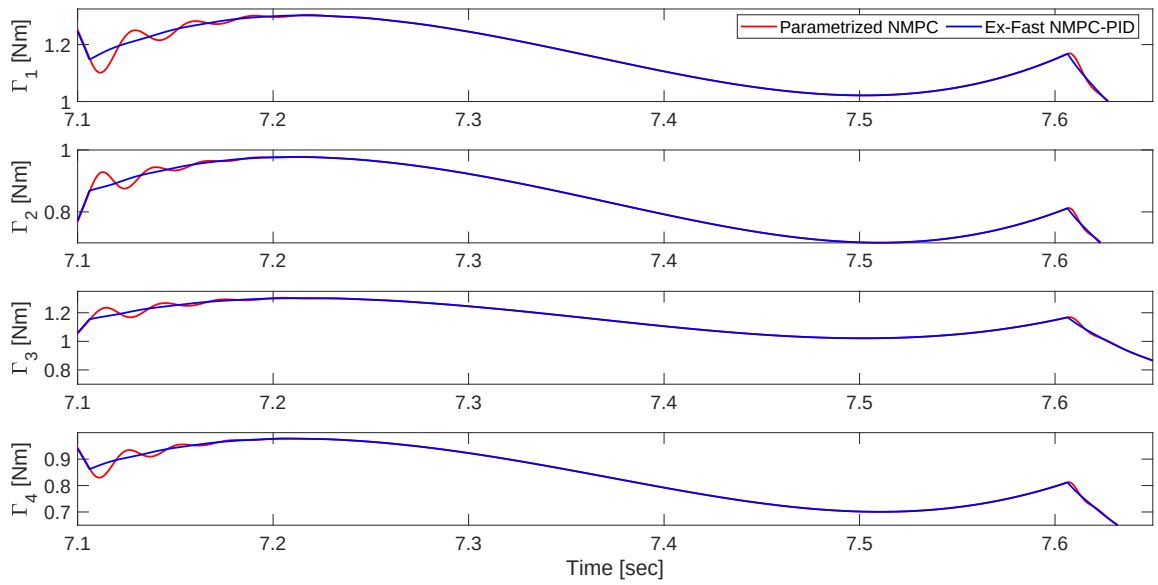
Standard PID		Ex-Fast NMPC-PID	
Parameter	Value	Parameter	Value
		N_p	3
K_P	4000	N_c	3
K_D	110	λ	0.333
K_I	6	K_P	4000
Γ	[-8,8]	K_D	110
		K_I	6
		Γ	[-8,8]

Table 6: Contribution 2: Control performance evaluation in terms of RMSE errors.

	RMSE$_J[deg]$	RMSE$_T[mm]$	RMSE$_R[deg]$
Standard PID	0.0391	0.1210	0.4275
Ex-Fast NMPC-PID	0.0211	0.0826	0.2568
Improvements	46.03%	46.48%	43.97%

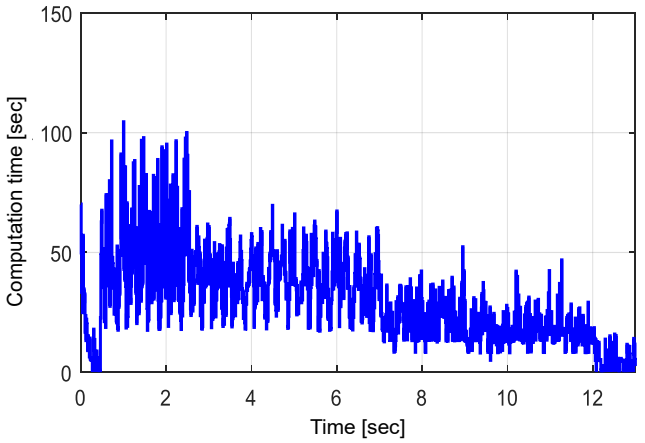


(a) Parameterized NMPC and Ex-Fast NMPC.

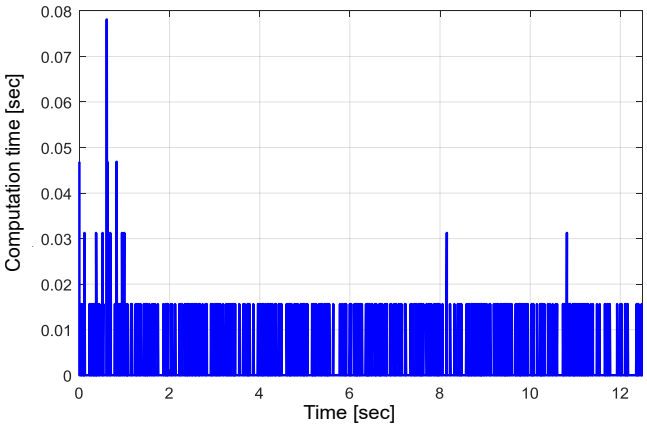


(b) An illustrative zoom.

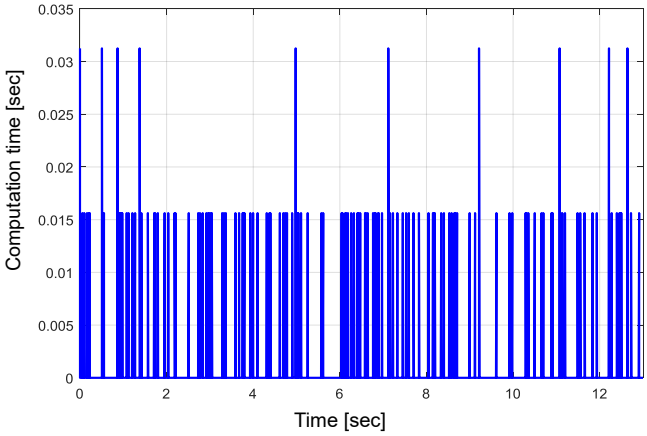
Figure 10: Evolution of the control inputs versus time in the numerical simulation. (a) Superimposed curves and (b) An illustrative zoom.



(a) Parameterized NMPC



(b) Ex-Fast NMPC (contribution 1)

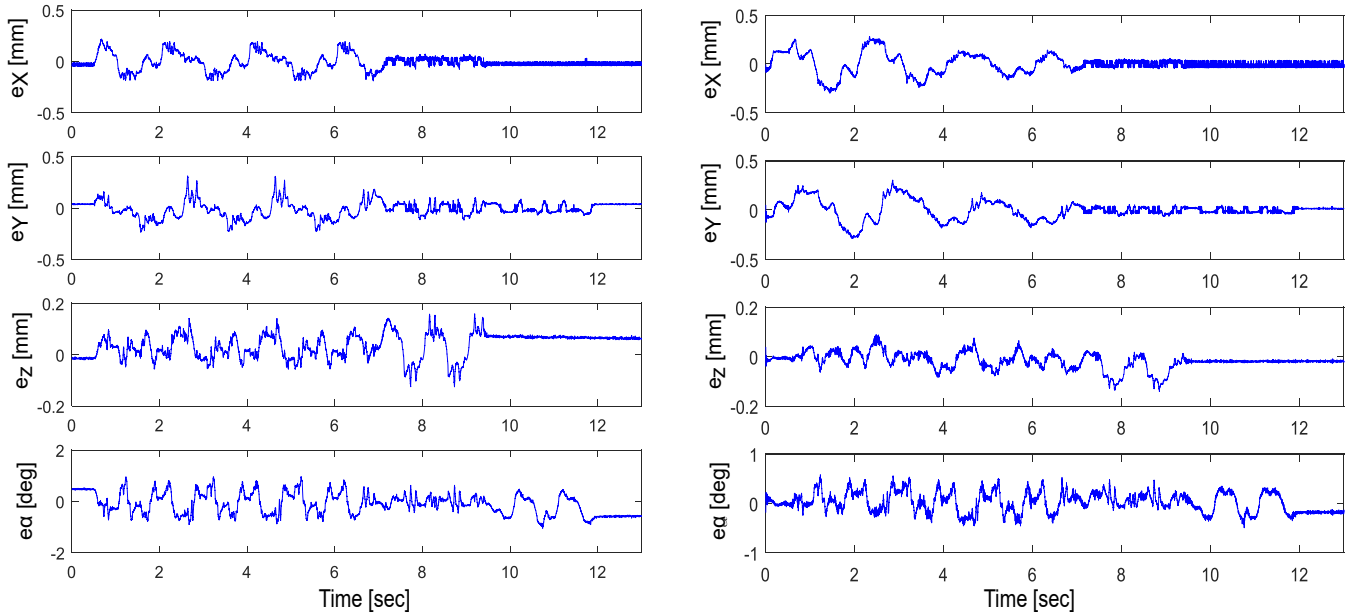


(c) Ex-Fast NMPC (contribution2)

Figure 11: Evolution of the computation time of the simulated three controllers, including (a) Parameterized NMPC, (b) Contribution 1 using nonlinear plant model, and (c) Contribution 2 using ANN.

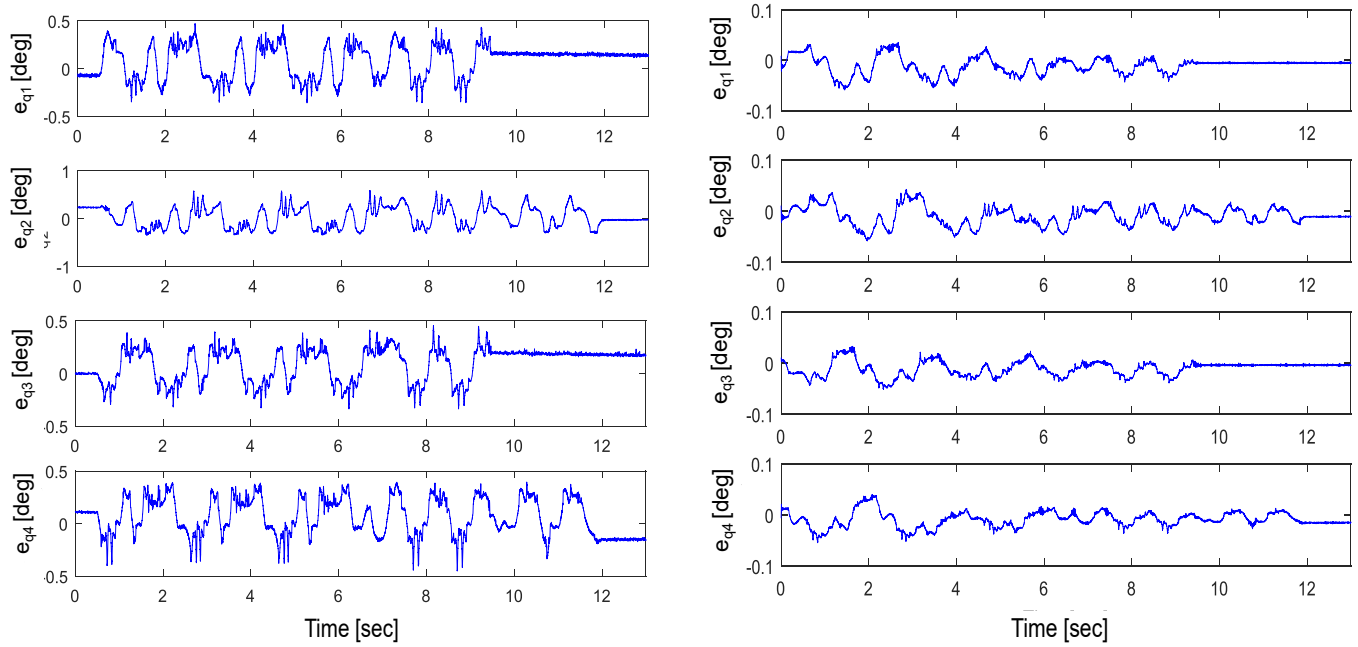
Table 7: Summary of the control design parameters of contribution 1.

Standard PID		Ex-Fast NMPC-PID	
Parameter	Value	Parameter	Value
		t_r	0.0540
		β	1
		γ	1
K_P	4000	λ	180
K_I	6	K_P	4000
K_D	110	K_I	6
Γ	[-8,8]	K_D	110
		N_p	1
		Γ	[-8,8]



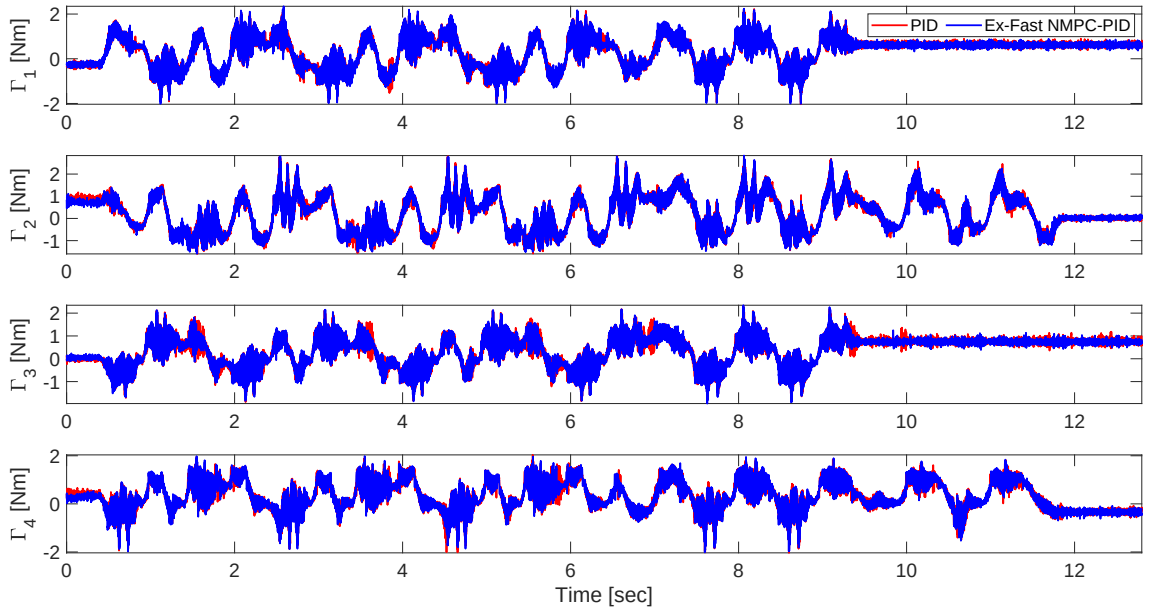
(a) PID (b) Ex-Fast NMPC-PID

Figure 13: Contribution 1: Evolution of the Cartesian tracking errors versus time, (a): PID, (b): Ex-Fast NMPC-PID.

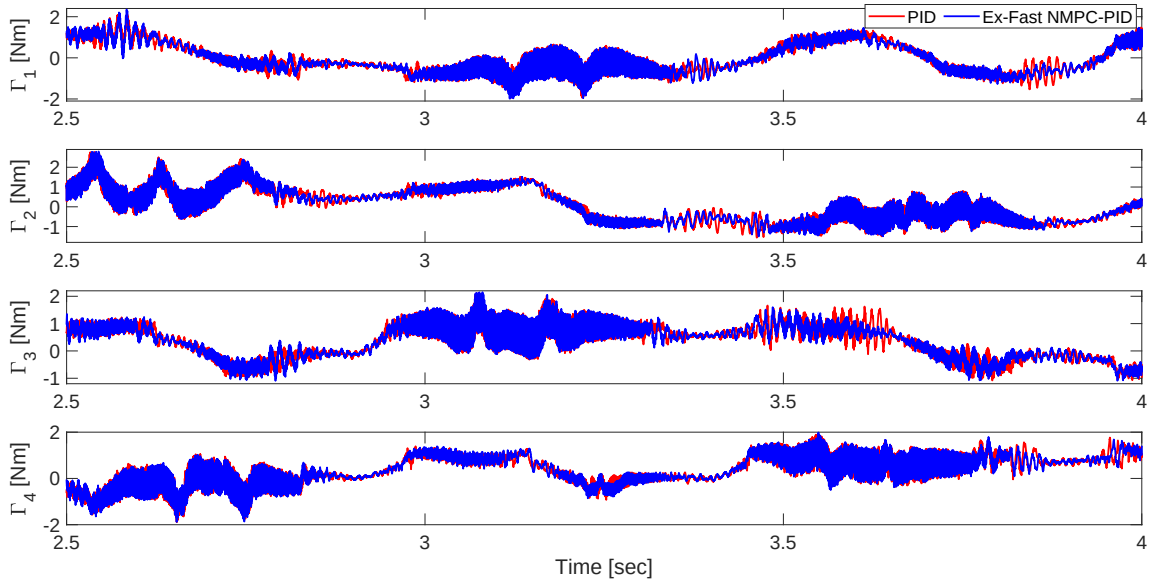


(a) PID (b) Ex-Fast NMPC-PID

Figure 14: Contribution 1: Evolution of the joint tracking errors versus time, (a): PID, (b): Ex-Fast NMPC-PID.



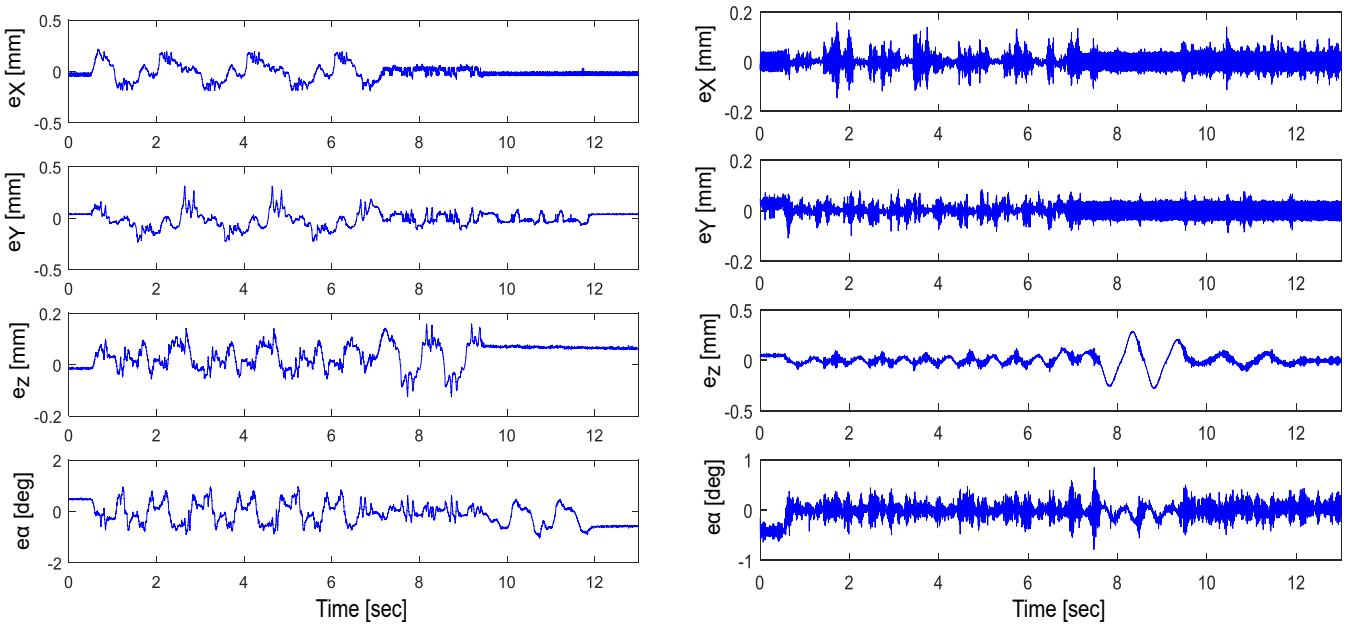
(a) PID and Ex-Fast NMPC-PID.



(b) An illustrative zoom.

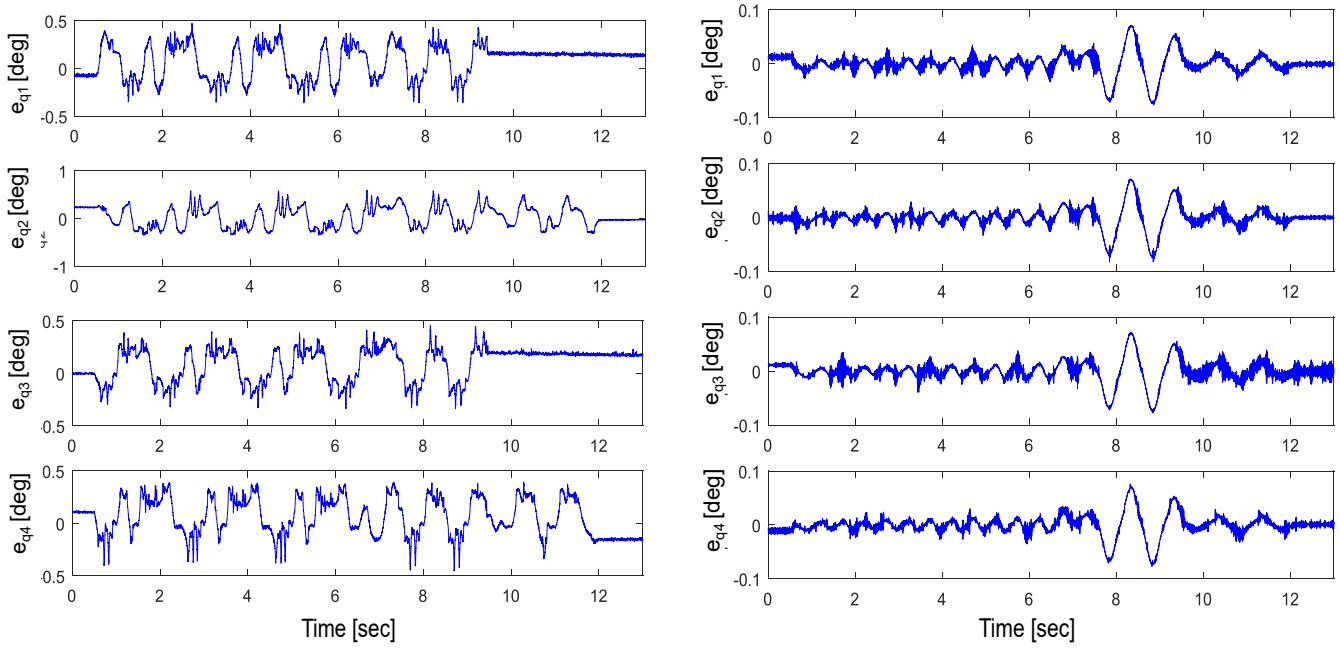
Figure 15: Contribution 1: Evaluation of the control input torques versus time, (a) PID and Ex-Fast NMPC-PID, (b) An illustrative zoom

.



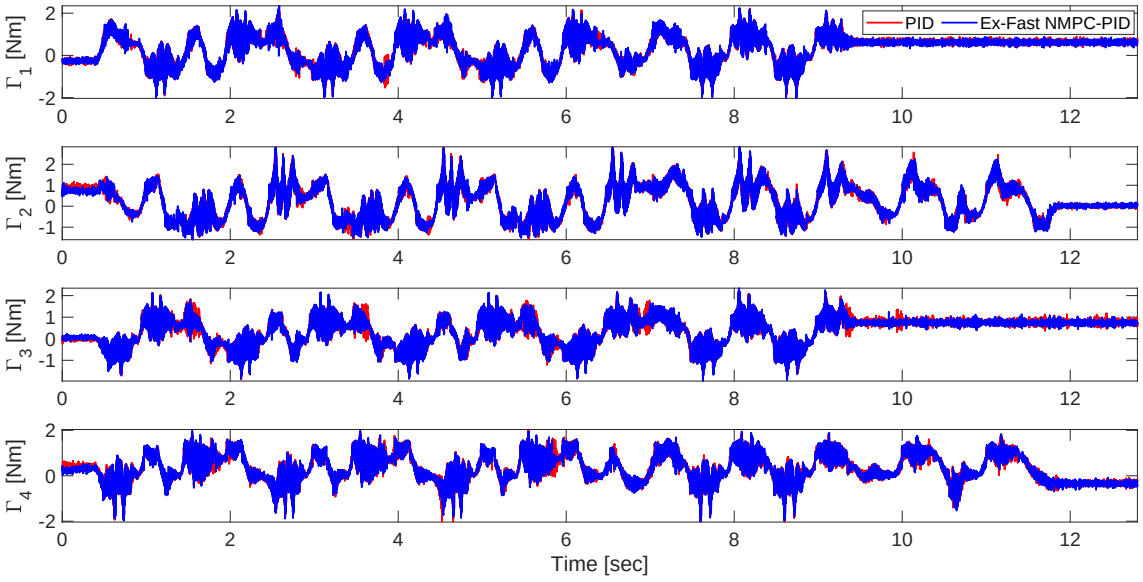
(a) PID (b) Ex-Fast NMPC-PID

Figure 16: Contribution 2: Evolution of the Cartesian tracking errors versus time. (a) PID and (b) Ex-Fast NMPC-PID.

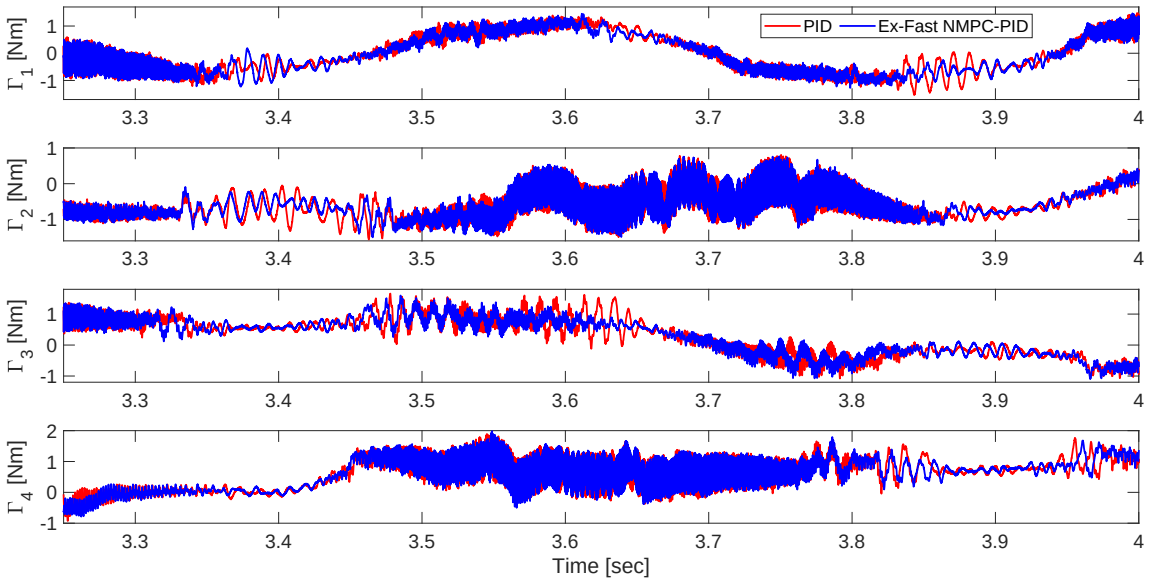


(a) PID (b) Ex-Fast NMPC-PID

Figure 17: Contribution 2: Evolution of the joint tracking errors versus time. (a) PID and (b) Ex-Fast NMPC-PID.



(a) PID and Ex-Fast NMPC-PID.



(b) An illustrative zoom.

Figure 18: Contribution 2: Evolution of the control input torques versus time. (a) PID and Ex-Fast NMPC-PID, (b) An illustrative zoom.