HAL
open science

A New Fast NMPC Scheme for Parallel Kinematic Manipulators: Design and Real-Time Experiments

Rihab Kouki, Ahmed Chemori, Faouzi Bouani

To cite this version:

Rihab Kouki, Ahmed Chemori, Faouzi Bouani. A New Fast NMPC Scheme for Parallel Kinematic Manipulators: Design and Real-Time Experiments. SCC 2019 - 1st IEEE International Conference on Signal, Control and Communication, Dec 2019, Hammamet, Tunisia. pp.69-75, 10.1109/SCC47175.2019.9116179 . lirmm-02426278

HAL Id: lirmm-02426278

https://hal-lirmm.ccsd.cnrs.fr/lirmm-02426278

Submitted on 2 Jan 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A New Fast NMPC Scheme for Parallel Kinematic Manipulators: Design and Real-Time Experiments

Rihab Kouki[1], Ahmed Chemori[2] and Faouzi Bouani[1]

***Abstract*— This paper adresses the design of a new extension of fast nonlinear model predictive control (NMPC) for parallel manipulators. The developed controller is based on a parameterized NMPC, a fast gradient solver and a proportional derived controller (PD). The main motivation behind the proposed approach is to improve the tracking performance of fast parallel manipulators and reduce the computation time per control iteration. This control scheme is faster, in terms of computing time, than the classical NMPC and ensures the robustness of the resulting closed-loop system. To demonstrate the effectiveness of the proposed controller, real time experiments are performed on a 4-DOF parallel kinematic manipulator called VELOCE.**

I. INTRODUCTION

Nonlinear Model Predictive Control (NMPC) is an advanced control technique that has been widely applied in industry due to its ability to handle multivariable nonlinear problems and deal with hard constraints. In the recent years, many efforts have been made to extend the application of NMPC strategies from slow dynamic systems to fast dynamic systems such as robotics, automotive and drive processes [1]–[4]. In order deal with high-speed systems, same In fast NMPC approaches have been developed such as the Real-time iteration scheme Multiple Shooting [5], Continuation/Generalized Minimum RESidual (C/GMRES) [6], Parameterized NMPC [7] and Control Updating Period (CUP) [8]. This class of fast controllers have been designed to accelerate the computation time to be able to handle the real-time control requirements of fast systems. However, despite its successful application, it is typically limited to a certain class of simple fast systems. This idea motivates the use of fast NMPC approach to control more complex systems, such as Parallel Kinematic Manipulators (PKMs). PKMs have been used for a wide range of potential applications [9]–[11]. This interest in parallel manipulators is increasing for various reasons, such as high accuracy and high acceleration capabilities. The development of a reliable NMPC approach for this kind of robots is a complex and difficult task due to their complex dynamics needing a high computation time. In this context, several ideas for designing predictive control schemes have been proposed to drive parallel manipulators [12]–[15]. These controllers are mainly based on simplified linear models to achieve the expected tracking performance. Nevertheless, such linear models are insufficient to describe all the complexity of parallel robots. In the case where a nonlinear dynamics should be taken into account in the control design, an NMPC approach becomes more appropriate. Another relevant work proposed by [16] uses the NMPC approach to control a 2-DOFs parallel robot actuated by pneumatic muscles. However, the effectiveness of this approach has been shown through numerical simulations. To the best authors' knowledge, NMPC controller has never been applied in real-time to PKMs. In order to resolve the online implementation problem of the conventional NMPC, fast control strategies have been used in this paper. The main objective of this work was to improve the performance of the conventional NMPC controller applied in particular to parallel robots. A predictive optimization framework based on the parameterized NMPC and the fast gradient solver is developed to achieve a high processing speed. The proposed control scheme has been experimentally validated on a 4-DOF parallel manipulator called VELOCE. The paper is organized as follows. Section II, describes the VELOCE robot and its simplified dynamic modeling. The new fast NMPC proposed solution for parallel manipulators is detailed in Section III. Section IV is dedicated to real-time experimental results and their interpretation. Finally, the conclusions and future work are given in Section V.

*This work was supported by the Erasmus+MIC mobility project

[1]Rihab Kouki and Faouzi Bouani are with Université de Tunis El Manar, Ecole Nationale d'Ingénieurs de Tunis Laboratoire Analyse, Conception et Commande de Systèmes, LR11ES20 Tunis, Tunisie rihab.kouki@enit.rnu.tn and faouzi.bouani@enit.utm.tn

[2]Ahmed Chemori is with LIRMM, Univ. Montpellier - CNRS, 161 rue Ada, 34095 Montpellier, France chemori@lirmm.fr

II. DESCRIPTION AND DYNAMIC MODELING OF VELOCE PKM

A. Description of the Veloce

The proposed experimental platform was developed at LIRMM Laboratory (France) for pick and place tasks. Each kinematic chain of the VELOCE robot is a serial arrangement of an actuator, an arm (including the rotor part of the actuator) and a forearms fixed to the moving platform by spherical joints (as illustrated in Fig.1). The moving platform is capable of performing three translations along x, y and z axes and one rotation α about the z axis. Its platform is composed of two parts, the upper and the lower part. Each pair of opposite kinematic chains is responsible for the motion of one part of the traveling plates. The relative movement, thanks to a single screw between the lower and upper part, generates the rotational DOF of the platform. The actuated joint coordinates are described by vector $q = [q_1, q_2, q_3, q_4]^T$ and completely define the configuration of the entire mechanism.

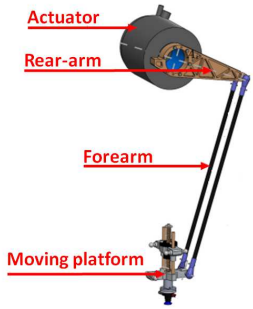

Fig. 1. Schematic view for one kinematic chain of VElOCE robot.

B. Veloce PKM Dynamics

To compute the simplified direct dynamic model of VELOCE, according to [17] the following assumptions are considered:

- The rotational inertia of the forearms is neglected and their mass is split up into two equivalent parts, one is added to the mass of the rear-arm, while the other one is added to the mass of the relevant part of the moving platform.
- Dry and viscous frictions in all passive and active joints are neglected.

The dynamic parameters of VELOCE are summarized in Table I and its dynamic model is established by separately analyzing the equilibrium of the arms and the traveling plate. For further details on the mechanical design of the VELOCE robot, the reader is referred to [18]. Then, the dynamic model expression in the Cartesian space can be expressed by

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

where $\dot{X} \in \mathbb{R}^4$ and $\ddot{X} \in \mathbb{R}^4$ are the Cartesian velocities and accelerations, respectively; $M_{tot} = M_{tp} + 4\frac{M_{forearm}}{2}$ is the total mass matrix of the manipulator, $M_{tp} \in \mathbb{R}^{4\times4}$ is the mass matrix of the traveling plate and $M_{foream} \in \mathbb{R}^{4\times4}$ denotes the mass matrix of the forearms. $I_{tot} = I_{act} + I_{arm} + \frac{l^2 M_{forearm}}{2}$ is its total inertia matrix, $I_{act} \in \mathbb{R}^{4\times4}$ and $I_{arm} \in \mathbb{R}^{4\times4}$ are the inertia matrices of the motor drives and the arms, respectively. The dynamic model of VELOCE can be expressed in a standard joint-space form as follows

$$M(q)\ddot{q} + C(q,\dot{q}) + G(\dot{q}) = \Gamma \tag{2}$$

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 centrifugal and the Coriolis matrix and $G(q) \in \mathbb{R}^4 = \Gamma_{flood}$ is the vector of gravitational forces. The torque vector generated by the actuators, defined by $\Gamma \in \mathbb{R}^4$.

TABLE I
DYNAMIC PARAMETERS OF VELOCE

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 ($kg \cdot m^2$)

III. PROPOSED CONTROL SCHEME: A FAST NMPC BASED ON FAST SOLVER

A. Background on Fast NMPC formulation

To handle fast systems, a fast NMPC approach based on the parameterization technique was developed to achieve the objective of real-time control [19]. The main idea of the parameterized NMPC is to reduce the computation time by reducing the size of the optimization problem. In this context, the decision control sequence u can be defined using a low-dimension vector p rather than a future control sequence. More precisely, according to [19], the optimal set value $\hat{p}$ is obtained by minimizing a J cost function, as follows:

$$\hat{p} = arg \min_p [J(p, X(k))] \tag{3}$$

where $\hat{p}$ is the optimal parameter vector, $X(k)$ is the current value of the state at time k and $J(.)$ is the cost function to minimize expressed by

$$J(p,x(k)) = \sum_{i=0}^{N_p-1} [(\hat{Y}(i,X(k),p) - Y_f(Y^d(k),i))]^2 + \lambda[(\hat{X}(N_p,X(k),p) - \hat{X}_f(\hat{X}(N_p,X(k),p),\Gamma^{(N_p)}(p))]^2 \tag{4}$$

where N_p is the prediction horizon, $\hat{Y}(i,X(k),p)$ is the future output predictor vector defined by p over $[k, k+N_{p-1}]$ and started from the state $X(k)$. $Y_f(Y^d,i))$ is the filtered version of the desired trajectory enabling to decouple the response time from overshoots, as specified by

$$Y_f(Y^d,i)) = Y^d + e^{-3T_e i/t_r}(Y(k) - Y^d(k)) \tag{5}$$

where T_e denotes the sampling time, t_r is the desired setting time of the closed-loop system and $Y(k)$ is the output system at present instant. λ is the weighting factor, $\hat{X}(N_p,X(k),p)$ is the state value at the end of the prediction horizon and the vector $\hat{X}_f(\hat{X}(N_p,X(k),p),\Gamma^{N_p}(p))$ is computed according to the state prediction $\hat{X}(N_p,X(k)$ and the control low signal Γ.

Based on the new optimal value $\hat{p}$ obtained by solving the optimization problem (3), the following exponential parameterized structure of the future control sequence, over the sampling period $[k-1,k]$, can be defined by:

$$\Gamma(k+i) = Sat_{\Gamma_{max}}^{\Gamma_{min}}(\Gamma^* + \alpha_1 e^{-\lambda i T_e} + \alpha_2 e^{q\lambda i T_e}) \tag{6}$$

where $i \in \{0,\cdots,N_p-1\}$, the stationary control is defined by Γ^* and the tuning parameters are denoted by $\lambda > 0$ and $q \in \mathbb{N}$. Sat is the saturation map, with $Sat : \mathbb{R}^{n_\Gamma} \mapsto \mathbb{R}^{n_\Gamma}$ defined as follows for $i \in \{1,\cdots,n_\Gamma\}$:

$$Sat_{\Gamma_{max}}^{\Gamma_{min}}(\Gamma_i) = \begin{Bmatrix} \Gamma_{min} & \text{if } \Gamma_i \le \Gamma_{min} \\ \Gamma_{max} & \text{if } \Gamma_i \ge \Gamma_{max} \\ \Gamma_i & \text{otherwise} \end{Bmatrix} \tag{7}$$

The coefficients of the exponential parameterization $\alpha_1, \alpha_2 \in \mathbb{R}^{n_u}$ are obtained by solving a simple linear equations system as follows:

$$\Gamma^* + \alpha_1 + \alpha_2 = \Gamma(k-1) \tag{8}$$

$$\alpha_1(e^{-\lambda T_e}-1)+\alpha_2(e^{-q\lambda T_e}-1)=p\delta_{max} \tag{9}$$

Equation (8) is the first condition guarantees the continuity of the control sequence, while (9) defines the constraints on the variation rates of the control input $(\Gamma(k)-\Gamma(k-1))=\delta_{max})$ that should not exceed a fraction $p\in[-1,+1]^{n_\Gamma}$ of some maximal allowable values $\delta_{max}\in\mathbb{R}^{n_\Gamma}$. According to (6) one can consider that the control value Γ^* can be defined as a part of the decision variable p with $p\in\mathbb{R}^{2\times n_\Gamma}$. Therefore, the parametrization of the control sequence is defined as

$$\Gamma(k+i)=Sat_{\Gamma_{max}}^{\Gamma_{min}}(P_\Gamma+\alpha_1(p)e^{-\lambda iT_e}+\alpha_2(p))e^{q\lambda iT_e}) \tag{10}$$

where $P_\Gamma=[p_{n_\Gamma+1},\cdots,p_{2\times n_\Gamma}]^T$ is the latest value of the optimal control sequence p. Note that the parameter vector p is defined as $[p_1,\ldots,p_{n_\Gamma}]^T\in[-1,+1]^{n_\Gamma}$ and $[p_{n_\Gamma+1},\ldots,p_{2\times n_\Gamma}]^T=P_\Gamma\in\mathbb{R}^{n_\Gamma}$. This constraints is chosen to define the stationary control value Γ^* and to computes the parameters of the control law. Then, the coefficients $\alpha_1=\left[\begin{array}{ccc}\alpha_1^{\Gamma_1}; & \cdots; & \alpha_1^{u_{\Gamma_u}}\end{array}\right]$ and $\alpha_2=\left[\begin{array}{ccc}\alpha_2^{u_1}; & \cdots; & \alpha_2^{u_{n_u}}\end{array}\right]$ are determined by solving the following system:

$$\Psi(p)=M^{-1}U(p) \tag{11}$$

where,

$$\Psi(p)=\begin{bmatrix}\alpha_1^{\Gamma_1}(p)\\ \alpha_2^{\Gamma_2}(p)\\ \vdots\\ \vdots\\ \alpha_1^{\Gamma_{n_\Gamma}}(p)\\ \alpha_2^{u_{n_\Gamma}}(p)\end{bmatrix},\ U(p)=\begin{bmatrix}\Gamma_1(k-1)-P_{\Gamma_1}\\ p_1\delta_{max}^1\\ \vdots\\ \vdots\\ u_{n_\Gamma}(k-1)-P_{\Gamma_{n_\Gamma}}\\ p_{n_\Gamma}\delta_{max}^{n_\Gamma}\end{bmatrix} \tag{12}$$

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

As a result, the generation of the parameterized control low Γ depends on the optimal vector $p\in\mathbb{R}_{n_\Gamma}^n$ which is determined from equation (3). The first part of the resulting control sequences is injected into the system at each sampling time. The n-degree of freedom of the parameter setting p are used to improve the transient behavior of the tracking error, represented by exponential terms.

It important to emphasize that the exponential parameterization approach for NMPC can be formalized by several ways. However, to solve the real-time problem of fast PKM robots, it is necessary to understand that simply changing the overall shape of the NMPC controller based on the exponential parameterization technique would not be enough. Consequently, the next step is to use a fast optimization solver to obtain the set parameters p at a reduced computation time compared to conventional solvers such as *fmincon function of Matlab*.

B. Proposed Extension

A new nonlinear extension of the predictive control architecture for fast robots is proposed in this section. This developed control scheme essentially consists of a parametric approach, a fast gradient solver and an additional PD control term to improve the tracking performance as illustrated in Fig. 2. The following control law is then proposed to control the manipulator:

$$\Gamma=\Gamma_{Fast\,NMPC}+\Gamma_{PD} \tag{13}$$

where $\Gamma_{Fast\,NMPC}$ is the fast NMPC control signal and Γ_{PD} is the control vector of the proportional derivative controller.

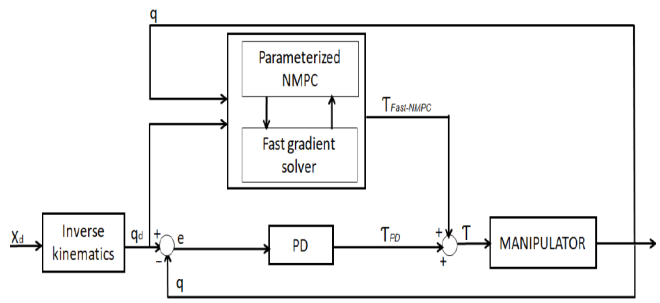

Fig. 2. Block diagram of the proposed control scheme.

For real-time implementation purpose, several optimization techniques were tested such as the Nlopt library and the CPU approach to enable fast NMPC implementation for PKM robots. However, not all these solutions were able to solve the real-time control problem of this parallel manipulator. Then comes the idea to extend the state of the art of fast NMPC controllers. This solution enables an explicit handling of nonlinear robot model and a reduction in the computation time. The proposed optimization problem solver is a fast gradient method that works in the millisecond range. The description of this solver is detailed in Algorithm hereafter. In fact, the solution of the minimization step consists of the parameter vector p. This algorithm starts with an initialization step of the control vector $\tilde{p}$. The next steps in algorithm 1 are the updates of the control form to define the parameter vector p of the parameterized NMPC approach and then to compute $p(k+1)=\tilde{p}(k+1)+\beta(\tilde{p}(k+1)-\tilde{p}(k))$, leading to the control equation $\tilde{p}(k+1)=p(k)-\frac{1}{L}\nabla J(\tilde{p},(Y(i,x(k),p))$, where L is the Lipshitz constant of the gradient ∇J and $\beta\in[0,1[$ is the problem conditioning.

Algorithm 1 Fast Gradient Solver

1: **Intialize** $\tilde{p}(k)\leftarrow p(k)\in\mathbb{R}^{n_u}$
2: $\tilde{p}(k+1)=p(k)-\frac{1}{L}\nabla J(\tilde{p},(Y(i,x(k),p))$
3: $p(k+1)=\tilde{p}(k+1)+\beta(\tilde{p}(k+1)-\tilde{p}(k))$

IV. EXPERIMENTAL VALIDATION OF THE PROPOSED CONTROL SCHEME

The objective of this section is to present the experimental testbed and discus the real-time results obtained by the

application of the proposed control schemes described in section III to the parallel robot described in section II.

A. Experimental Testbed and implementation issues

In order to apply the proposed NMPC strategy extended to fast robots in real time, VELOCE PKM prototype will be used. Its mechanical structure is illustrated in Fig. 3. The arms of this robot are mounted on four direct-drive motors TMB0140-100-3RBS from ETEL Motion Technology. These actuators are capable of reaching 550 rpm speed and can deliver a maximum peak torque of 127 Nm. Each actuator is equipped with a non-contact incremental optical encoder providing a total number of 5000 pulses per revolution. The global structure of the manipulator is able of reaching a maximum velocity of 10 m/s, a maximum acceleration of 20 G and can support a maximum payload of 10 Kg. For the real-time experimental tests, the control schemes

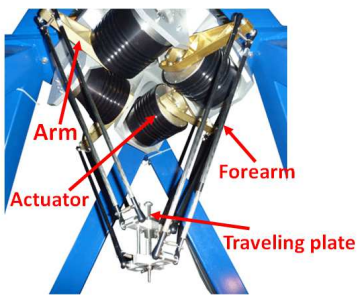

Fig. 3. View of the experimental setup of VELOCE PKM.

should be implemented using Mathworks Simulink/Matlab and compiled with XPC/Target computer and Real-Time toolbox. The generated code is uploaded to the target PC, an industrial computer running at a frequency of 10 KHz. Therefore, the real-time implementation of the proposed nonlinear controller was performed with a sampling time of 0.1 ms. In order to quantify the improvement brought by the proposed control scheme described in section III, we introduce the following Root Mean Square of the tracking errors (RMSE) based criteria:

$$RMSE_{tra} = (\frac{1}{N}\sum_{i=1}^{N} e_x^2(i) + e_y^2(i) + e_z^2(i))^{\frac{1}{2}} \tag{14}$$

$$RMSE_{rot} = (\frac{1}{N}\sum_{i=1}^{N} e_\alpha^2(i))^{\frac{1}{2}} \tag{15}$$

$$RMSE_q = (\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{4} e_{q_j}^2(i))^{\frac{1}{2}} \tag{16}$$

In these formulas, $RMSE_{tra}$, $RMSE_{rat}$ denote the RMSE criteria based on the computed Cartesian tracking errors where the translational movements are separated from the rotational ones, N is the number of samples recorded, e_x, e_y, e_z denote the tracking error along the x, y and z, respectively and e_α is the tracking error of the platform's rotations. $RMSE_q$ define the RMSE criteria based on the joint tracking errors and e_{q_j} $j = 1,...,4$ denotes the tracking error of the ith joint.

B. Real-time Experimental Results

A combination of the parametrization technique and the fast gradient solver is developed to obtain the best computational performance required by the VELOCE robot. The computing time is reduced without any linearization technique. The Extended-Fast NMPC-PD based control structure design (Ex-Fast NMPC-PD) is an extended version of Fast NMPC approach. It has a computation time (up to 0.0059 ms) compared to the standard NMPC strategy (greater than 117240 ms), which is unsuitable for real-time applications to control fast PKMs. We can see that the average execution time for the different operating modes remains within the range of $[0.0047, 0.0059]$ ms (see Fig. 4).

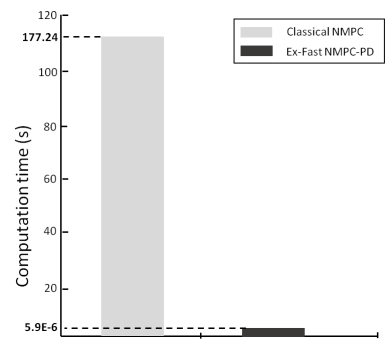

Fig. 4. Summary of the computation time achieved by both controllers (Classical NMPC and Ex-Fast NMPC-PD).

The desired trajectory followed by the VELOCE robot is a sequence of point-to-point motions with a duration of each motion T= 0.5 s. The corresponding Cartesian reference trajectory is illustrated in a 3D view in Fig. 5. The obtained

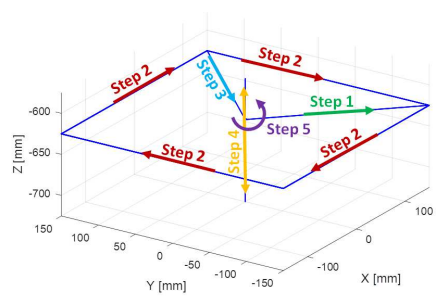

Fig. 5. Desired trajectory in Cartesian space for VELOCE robot (3D point-to-point).

results by the proposed regulator are shown in Fig. 6, where the robot's platform starts from an arbitrary position to the desired initial position (0,0,-625,0) mm and from the initial one to the final Cartesian position (0,0,-625,300) mm and then return to the initial position. The motivation behind this scenario is to assess the tracking capabilities in terms of Cartesian and articular tracking errors. In order to highlight the outcomes of the extended fast NMPC-PD controller, its performance is proposed to be compared with the one of a standard PD controller. The used parameters of both controllers are summarized in Table II. These parameters

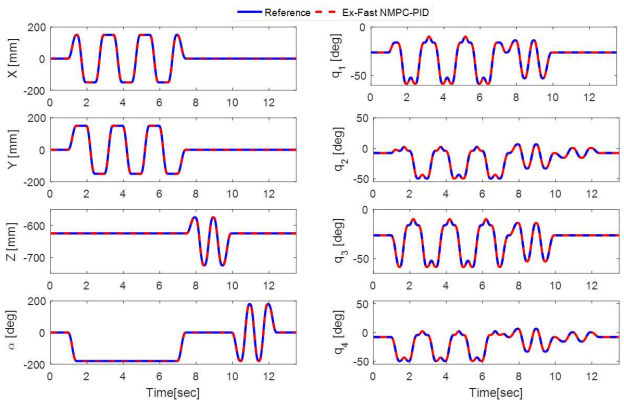

Fig. 6. Trajectory tracking versus time, left: Cartesian space, right: joint space.

have been adjusted so that each controller can provide its best performance. The obtained results by the proposed controller are illustrated in Fig. 7 and 8. For the seek of clarity, these results are zoomed around the interval [7.5, 12] seconds, as shown in Fig. 9 and 10. The performance comparison of both controllers is summarized in Table III, based on the proposed performance-evaluation-criteria. By analyzing Fig. 9, a performance comparison between both controllers in terms of Cartesian tracking errors is illustrated. It can be clearly see that the performance of the proposed controller is significantly improved compared to the standard PD control in terms of tracking errors (up to 35.81%). Also, the rotation movements errors of the traveling plat is improved by 32.68% than that of the PD scheme. The joint tracking errors, with a zoomed interval, for this scenario are displayed in Fig. 10. As with previous results, the extended Fast NMPC controller outperforms the PD controller in terms of joint tracking errors (improvement of approximately 32.36%). The torques generated by both controllers are illustrated in Fig. 11, it is possible to notice that the standard PD is delayed compared to the extended fast NMPC-PD controller and the amplitudes of both inputs are roughly similar. It should be also noted that all torques remain within the admissible limits of each actuator (127 N.m). According to the obtained results, we can conclude that the proposed controller significantly improves the trading performance of the closed-loop system compared to the controller PD.

TABLE II
SUMMARY ON CONTROL DESIGN PARAMETERS FOR REAL-TIME EXPERIMENTS

Parameter	Value	Parameter	Value
T_e	0.1 ms	t_r	0.0540
N_p	1	K_p	4000
λ	180	K_d	110
q	1	Γ	[-8, 8]

TABLE III
PERFORMANCE COMPARISON BETWEEN THE PROPOSED CONTROL APPROACH AND THE STANDARD PD

Performance	Standard PD	Ex-Fast-FNMPC-PD	Improvements
$RMSE_c[mm]$	3.63	2.33	35.81%
$RMSE_r[deg]$	242.18	163.04	32.68%
$RMSE_q[deg]$	13.29	8.99	32.36%

V. CONCLUSIONS AND FUTURE WORK

In this paper, a fast NMPC control theory is developed and applied to control PKMs. In order to improve the tracking performance and achieve the objective of real-time control, a new extended fast NMPC strategy has been proposed for parallel manipulators. To this aim, a combination of a parameterized approach, a fast nonlinear solver and a proportional derivative control is proposed. The extended controller benefits from the advantages of the control sequence configuration approach and the fast gradient optimization. This proposed idea reduces the computation time compared to the classical NMPC and allows high processing speeds. To evaluate the effectiveness of the proposed controller, real-time experiments were conduced on a 4-DOF parallel robot. The experimental results show that the proposed controller has good computational capabilities and achieves satisfactory control performance for parallel manipulators. For future work, a detailed stability analysis of the proposed control for VELOCE PKM together with more extensive results can be investigated.

ACKNOWLEDGMENT

This paper has been supported by the Erasmus+MIC mobility project between the University of Montpellier and the Tunis El Manar University.

REFERENCES

[1] M. Neunert, C. De Crousaz, F. Furrer, M. Kamel, F. Farshidian, R. Siegwart, and J. Buchli, "Fast nonlinear model predictive control for unified trajectory optimization and tracking," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 1398–1404.

[2] M. Kamel, K. Alexis, M. Achtelik, and R. Siegwart, "Fast nonlinear model predictive control for multicopter attitude tracking on so (3)," in *2015 IEEE Conference on Control Applications (CCA)*. IEEE, 2015, pp. 1160–1166.

[3] J. Albersmeyer, D. Beigel, C. Kirches, L. Wirsching, H. G. Bock, and J. P. Schlöder, "Fast nonlinear model predictive control with an application in automotive engineering," in *Nonlinear Model Predictive Control*. Springer, 2009, pp. 471–480.

[4] L. Wirsching, J. Albersmeyer, P. Kühl, M. Diehl, and H. Bock, "An adjoint-based numerical method for fast nonlinear model predictive control," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 1934–1939, 2008.

[5] M. Diehl, R. Findeisen, F. Allgöwer, H. G. Bock, and J. P. Schlöder, "Nominal stability of real-time iteration scheme for nonlinear model predictive control," *IEE Proceedings-Control Theory and Applications*, vol. 152, no. 3, pp. 296–308, 2005.

[6] *Application of Newton/GMRES Method to Nonlinear Model-Predictive Control of Functional Electrical Stimulation*, Ottawa, Canada, 2016.

[7] C. Damour, M. Benne, A. J.-J. Kadjo, S. Rosini, and B. Grondin-Perez, "Fast NMPC scheme of a 10 kW commercial PEMFC," *International Journal of Hydrogen Energy*, vol. 38, no. 18, pp. 7407–7413, 2013. [Online]. Available: http://hal.univ-reunion.fr/hal-01202275

[8] M. Alamir, "Monitoring control updating period in fast gradient based NMPC," in *ECC*. IEEE, 2013, pp. 3621–3626.

[9] N. Seward and I. A. Bonev, "A new 6-dof parallel robot with simple kinematic model," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 4061–4066.

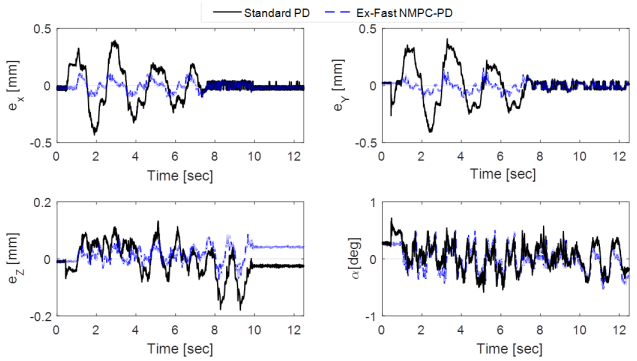

Fig. 7. Evolution of the Cartesian tracking errors versus time.

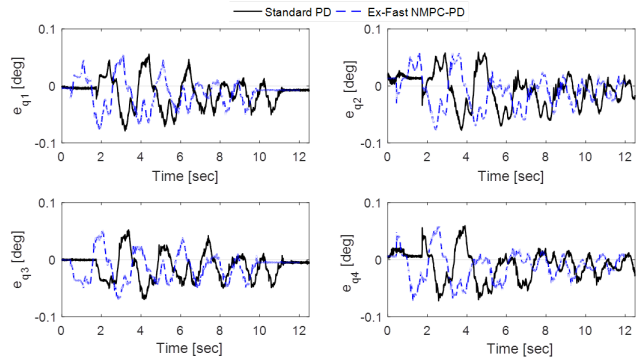

Fig. 8. Evolution of the Joint tracking errors versus time.

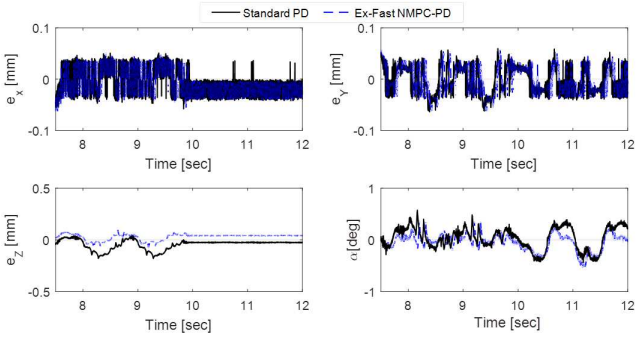

Fig. 9. Zoomed-in view of the Cartesian tracking errors versus time.

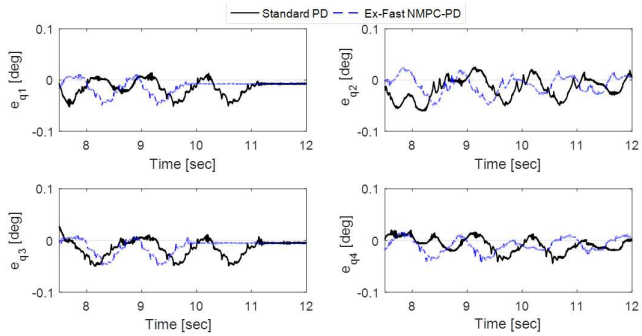

Fig. 10. Zoomed-in view of the Joint tracking errors versus time.

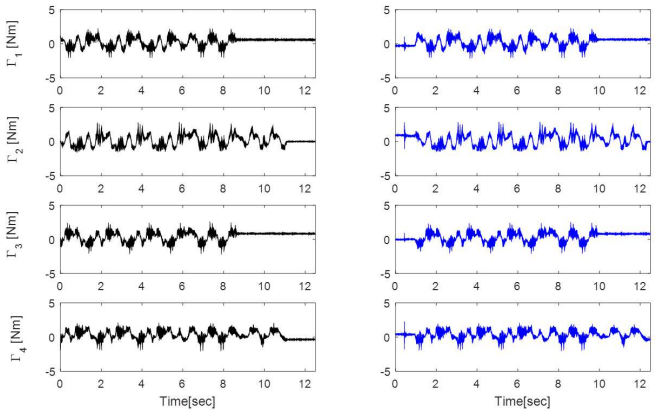

Fig. 11. Control inputs, left: Standard PD, right: Ex-Fast NMPC-PD.

[10] B. Lian, T. Sun, Y. Song, Y. Jin, and M. Price, "Stiffness analysis and experiment of a novel 5-dof parallel kinematic machine considering gravitational effects," *International Journal of Machine Tools and Manufacture*, vol. 95, pp. 82–96, 2015.

[11] J. J. Fernandes and A. S. A, "Kinematic and dynamic analysis of 3puu parallel manipulator for medical applications," *Procedia Computer Science*, vol. 133, pp. 604 – 611, 2018, international Conference on Robotics and Smart Manufacturing (RoSMa2018). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050918310433

[12] M. Katliar, J. Fischer, G. Frison, M. Diehl, H. Teufel, and H. H. Bülthoff, "Nonlinear model predictive control of a cable-robot-based motion simulator," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9833 – 9839, 2017, 20th IFAC World Congress.

[13] A. Vivas and P. Poignet, "Predictive functional control of a parallel robot," *Control Engineering Practice*, vol. 13, no. 7, pp. 863 – 874, 2005, control Applications of Optimisation.

[14] S. Wen, G. Qin, B. Zhang, H. Lam, Y. Zhao, and H. Wang, "The study of model predictive control algorithm based on the force/position control scheme of the 5-dof redundant actuation parallel robot," *Robotics and Autonomous Systems*, vol. 79, pp. 12 – 25, 2016.

[15] K. Belda, J. Böhm, and P. Píša, "Concepts of model-based control and trajectory planning for parallel robots," in *Proceedings of the 13th IASTED International Conference on Robotics and Applications*, ser. RA '07. Anaheim, CA, USA: ACTA Press, 2007, pp. 15–20. [Online]. Available: http://dl.acm.org/citation.cfm?id=1659997.1660001

[16] D. Schindele and H. Aschemann, "Nonlinear model predictive control of a fast parallel robot actuated by pneumatic muscles," *IFAC Proceedings Volumes*, vol. 40, no. 12, pp. 591 – 596, 2007, 7th IFAC Symposium on Nonlinear Control Systems.

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

[18] M. Bennehar, A. Chemori, S. Krut, and F. Pierrot, "Adaptive control of parallel manipulators: Design and real-time experiments," 2016.

[19] M. Alamir, "A framework for real-time implementation of low-dimensional parameterized nmpc," *Automatica*, vol. 48, no. 1, pp. 198 – 204, 2012.