Control Engineering Practice 39 (2015) 1–11

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

Enlarging parallel robot workspace through Type-2 singularity crossing

Georges Pagis [a,b], Nicolas Bouton [a], Sebastien Briot [b,*], Philippe Martinet [b,c]

[a] Institut Pascal, IFMA, MMS Department, UMR CNRS 6602, 63000 Clermont-Ferrand, France
[b] Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN), UMR CNRS 6597, 44321 Nantes, France
[c] LUNAM, Ecole Centrale de Nantes, 44321 Nantes, France

ARTICLE INFO

Article history:
Received 18 December 2013
Accepted 31 January 2015

Keywords:
Parallel robots
Computed torque control
Multi-model approach
Singularities

ABSTRACT

In order to increase the reachable workspace of parallel robots, a promising solution consists of the definition of optimal trajectories that ensure the non-degeneracy of the dynamic model in the Type 2 (or parallel) singularity. However, this assumes that the control law can perfectly track the desired trajectory, which is impossible due to modeling errors.

This paper proposes a robust multi-model approach allowing parallel robots to cross Type 2 singularities. The main idea is to shift near singularities to a simplified dynamic model that can never degenerate. The two main contributions are the definition of an optimal trajectory crossing Type 2 singularities and the multi-model control law allowing to track this trajectory. The proposed control law is validated experimentally through a Five-bar planar mechanism.

1. Introduction

Contrary to serial robots, which are largely used in industry, parallel robots are under-represented despite having many advantages, such as higher acceleration capacities and a better payload-to-weight ratio. The small number of parallel mechanisms in factories can be explained by the relative complexity of their model and by the presence of singularities (Arakelian, Briot, & Glazunov, 2008; Conconi & Carricato, 2009; Gosselin & Angeles, 1990), which divide their workspace into different aspects (each aspect corresponding to one or more assembly modes, Merlet, 2006). The manipulator workspace is therefore usually reduced to only one of these aspects, resulting in a greatly reduced reachable workspace size. The main idea of this paper is to propose a control law allowing parallel manipulators to move between those different aspects.

Various types of singularity exist, and for a global overview of the singularity problem the reader is referred to Conconi and Carricato (2009). However, since Type 2 (Gosselin & Angeles, 1990) (or parallel) singularities are probably the most constraining ones, this paper will focus only on this type. In these singularities, one (or more) manipulator's degree of freedom becomes uncontrollable. In order to increase the workspace size several approaches have been envisaged in the literature, such as:

- The design of parallel robots without singularities. This can be done by using the optimal design approach (Briot & Arakelian, 2010; Liu, Wang, & Pritschow, 2006) or by creating decoupled mechanisms (Gogu, 2004; Kong & Gosselin, 2002). This solution is the most usual one, but it usually leads to the design of robots with a small workspace size or robot architectures with very low practicability.
- The use of redundancy (Kurtz & Hayward, 1992; Nahon & Angeles) or, to reduce costs, the use of mechanisms with variable actuation modes (Arakelian et al., 2008; Rakotomanga et al., 2006). These mechanisms can change the way they are actuated without adding additional actuators, but this change can only be carried out when the mechanism is stopped, thus increasing the time necessary to perform the task.
- Planning assembly mode changing trajectories. A first way to do this is to bypass a cusp point (Zein, Wenger, & Chablat, 2008). However, this solution is hardly practical for two main reasons: (i) it forces the mechanism to follow a particular trajectory, which can be very different from the desired one; (ii) only a few mechanisms have cusp points. A second solution is to go directly through a Type 2 singularity (Briot, Arakelian, & Chablat, 2008; Ider, 2005). In Briot et al. (2008), a physical criterion, obtained through the analysis of the dynamic model, is presented. It enables the computation of a trajectory which can cross a singularity without the dynamic model degenerating, by respecting the criterion in question on the singularity locus.

This last solution is promising, since it can considerably increase the workspace size of any parallel mechanism. However, in previous studies it was considered that the controller allowed

* Corresponding author.
E-mail addresses: Georges.Pagis@ifma.fr (G. Pagis), Nicolas.Bouton@ifma.fr (N. Bouton), Sebastien.Briot@irccyn.ec-nantes.fr (S. Briot), Philippe.Martinet@irccyn.ec-nantes.fr (P. Martinet).

http://dx.doi.org/10.1016/j.conengprac.2015.01.009
0967-0661/

the mechanism to perfectly track the desired trajectory. This is obviously impossible due to modeling uncertainties. In order to fill this gap, the aim of the present paper is to propose an advanced control law dedicated to Type 2 singularity crossing.

Since Type 2 singularities have an impact on the dynamic of the mechanism, the use of a geometric/kinematic controller would not allow taking into account this dynamic degeneracy. Moreover, in Briot et al. (2008) it has been shown that, in order to cross a Type 2 singularity, the mechanism has to track a trajectory that respects a specific criterion on the singularity locus. This criterion gives a relation between the singular position, the mechanism's speed and its acceleration when crossing the singularities. However, only dynamic controllers can perform tracking of velocity and acceleration (Khalil & Dombre, 2004). Most of the different dynamic control loop algorithms can be considered as special cases of the computed torque control (*CTC*) (Craig & Hall, 2005; Khalil & Dombre, 2004; Spong, Hutchinson, & Vidyasagar, 2006). This technique consists of an inner nonlinear compensation loop and an outer loop with an exogenous control signal u. However, this control law is sensitive to modeling errors, so the dynamic model must be well identified (Briot & Gautier, 2012; Gautier, 1997).

When applying a *CTC* control law for singularity crossing, the degeneracy of the dynamic model near the singularity results in computing infinite torques, thus leading to the instability of the controller. No controller has ever been developed for singularity crossing,[1] and most studies concentrate on solutions in order to *avoid* the singularities. In order to be used when crossing a Type 2 singularity, the dynamic model used by the *CTC* must not degenerate near singularities, even if the trajectory does not perfectly respect the physical criterion mentioned above. As a result, in this paper, a new multi-model *CTC* (e.g. see Craig & Hall, 2005; Spong et al., 2006) is proposed, which guarantees that the robot dynamic model of the mechanism does not degenerate near a singularity. This multi-model control law was developed thanks to the definition of a new dynamic criterion based on Briot et al. (2008). The contribution of this paper is to propose a complete methodology, from the trajectory planning to the achievement of singularity crossing on an experimental robot without path restriction.

This paper is organized as follows: first the approach used to compute the criterion for crossing Type 2 singularities is recalled, and a method developed to increase the robustness of the planned trajectory is proposed. Then, in Section 3, the multi-model *CTC* control law developed for crossing singularities is presented. Section 4 introduces the robot used to validate the Type 2 singularity crossing approach proposed. Finally, the relevancy of this controller is demonstrated through full-scale experiments on a Five-bar mechanism.

2. Trajectory generation for crossing a Type 2 singularity

2.1. Dynamic modeling of parallel mechanisms

This section will briefly recall the dynamic equations of a parallel manipulator composed of m links, n degrees of freedom (*dof*) and driven by n actuators. The manipulator is composed of legs attached to the base and to the mobile platform (for a more detailed analysis of closed-loop kinematic chain, the reader is turned to Merlet, 2006). The position and the speed of the manipulator can be fully described using:

- $\mathbf{q}=[q_1,q_2,\ldots,q_n]^T$ and $\dot{\mathbf{q}}=[\dot{q}_1,\dot{q}_2,\ldots,\dot{q}_n]^T$ which represent respectively the vectors of active joint variables and active joint velocities,
- $\mathbf{x}=[x,y,z,\phi,\psi,\theta]^T$ and $\mathbf{t}=[\dot{x},\dot{y},\dot{z},\dot{\phi},\dot{\psi},\dot{\theta}]^T$ which are the mobile platform pose parameters and their derivatives with respect to time; x, y and z represent the position of the platform controlled point and ϕ, ψ and θ represent the orientation of the platform about three axes $\mathbf{a}_\phi$, $\mathbf{a}_\psi$ and $\mathbf{a}_\theta$ (*Briant* angles).

Since the mechanism is moving, all of these terms depend on the current time t. However, for purposes of clarity, this dependency will not be written in all equations, and only not time-dependent terms will be specified. Those generalized coordinates are not independent. Indeed, let us consider the vector $\mathbf{v}$ regrouping the independent elements of $\mathbf{t}$. The matrix $\mathbf{D}$ relates the platform twist $\mathbf{t}$ (expressed in the base frame) to the vector $\mathbf{v}$ by (Merlet, 2006)

$$\mathbf{t}=\mathbf{D}\mathbf{v} \tag{1}$$

Note that for mechanism with 6 degrees of freedom, the matrix $\mathbf{D}$ is the identity $[6\times 6]$ matrix.

Relations between the platform coordinates are found by writing the closed-loop equations. Using Lagrangian formalism, the dynamic model of the mechanism can be written as

$$\tau=\mathbf{w}_b+\mathbf{B}^T\lambda, \tag{2}$$

$$\mathbf{w}_p=\mathbf{A}^T\lambda \tag{3}$$

where

- τ is the $[n\times 1]$ vector of the input efforts,
- λ is the $[n\times 1]$ vector of the Lagrange multipliers,
- $\mathbf{A}$ and $\mathbf{B}$ are two $[n\times n]$ matrices deduced from the mechanism loop-closure equations, such that $\mathbf{A}\mathbf{v}=\mathbf{B}\dot{\mathbf{q}}$ (Merlet, 2006),
- $\mathbf{w}_b$ and $\mathbf{w}_p$ are $[n\times 1]$ terms related to the Lagrangian L of the system by

$$\mathbf{w}_b=\frac{d}{dt}\left(\frac{\partial L}{\partial\dot{\mathbf{q}}}\right)-\frac{\partial L}{\partial\mathbf{q}},\quad \mathbf{w}_p=\frac{d}{dt}\left(\frac{\partial L}{\partial\mathbf{v}}\right)-\frac{\partial L}{\partial\mathbf{x}} \tag{4}$$

In this expression, $\mathbf{w}_p$ is the wrench applied to the platform by the legs and the external forces (Briot et al., 2008) and t is the time.

Then, assuming that matrix $\mathbf{A}$ can be inverted and by substituting (3) into (2), the general dynamic model of parallel manipulators is obtained (Khalil & Dombre, 2004):

$$\tau=\mathbf{w}_b+\mathbf{J}^{T0}\mathbf{w}_p, \tag{5}$$

where

- ${}^0\mathbf{w}_p$ is the expression of the wrench $\mathbf{w}_p$ in the base frame, i.e. ${}^0\mathbf{w}_p=\mathbf{D}\mathbf{w}_p$,
- $\mathbf{J}={}^0\mathbf{A}^{-1}\mathbf{B}$ is the matrix relating the platform twist $\mathbf{t}$ and $\dot{\mathbf{q}}$, with ${}^0\mathbf{A}$ the expression of matrix $\mathbf{A}$ in the base frame, i.e. ${}^0\mathbf{A}=\mathbf{A}\mathbf{D}^{-1}$.

2.2. Type 2 singularity crossing

Based on the analysis of the kinematic model, a classification of singularities into three different types is proposed in Gosselin and Angeles (1990):

- *Type 1 singularities or serial singularities* occur when the mechanism is in a position such that the kinematic matrix $\mathbf{B}$ becomes rank deficient. In such configurations, the mechanism loses its ability to move in one given direction.

[1] Note that a possible solution for crossing singularities is to plan a fast trajectory toward the singularity locus. Once the mechanism is close enough from the singularity, the controller could declutch the actuators, and couple them back once the mechanism is far enough from the singularity. This solution is obviously not robust at all and presents many disadvantages.

- *Type 2 singularities or parallel singularities* occur when the kinematic matrix ${}^0\mathbf{A}$ becomes rank deficient. In Type 2 singularities, one or more of the robot's degrees of freedom become uncontrollable. Such singularities divide the workspace into different aspects, resulting in a reduction in the manipulator's workspace. Moreover, in the presence of these singularities, the robot may also not be able to resist an external wrench applied to the platform, and the reactions in its joints grow to infinity.
- *Type 3 singularities* are configurations where both Type 1 and Type 2 singular configurations appear at the same time. They are disregarded in the rest of the paper as they appear if both Type 1 and Type 2 singularities exist.

Finally, parallel mechanisms with fewer than 6 *dof* can have another type of singularity: the constraint singularity (Conconi & Carricato, 2009; Zlatanov & Bonev).

If a parallel mechanism is in a singular Type 2 position, matrix $\mathbf{A}^T$ cannot be inverted in Eq. (3). The dynamic model degenerates and therefore cannot be solved.

However, as explained in the Introduction, it has been proven in Briot et al. (2008) that a mechanism can cross a Type 2 singularity without a torque discontinuity. Indeed, on a Type 2 singularity, the columns of ${}^0\mathbf{A}$ are linearly dependent, i.e. a vector $\mathbf{t}_s$ exists such that

$${}^0\mathbf{A}\mathbf{t}_s = 0 \Leftrightarrow \mathbf{t}_s^{T0}\mathbf{A}^T = 0 \tag{6}$$

The $[n \times 1]$ vector $\mathbf{t}_s$ represents the twist of the uncontrollable motion of the platform at the singularity locus (Merlet, 2006). Thus, multiplying (3) by $\mathbf{t}_s^T$ leads to

$$\mathbf{t}_s^{T0}\mathbf{A}^T\lambda = 0 \tag{7}$$

In this case, the following condition must also be satisfied:

$$\mathbf{t}_{s\,0}^{Tp}\mathbf{w} = 0 \tag{8}$$

which is the condition for the non-degeneracy of the dynamic model (Briot et al., 2008).

As a result, if the desired manipulator motion does not guarantee the achievement of a wrench $\mathbf{w}_p$ that respects condition (8), the dynamic model is degenerated and the desired manipulator input strains must grow to infinity to produce the desired platform motion. Physically, this condition means that the parallel manipulator can cross the Type 2 singularity if and only if the wrench ${}^0\mathbf{w}_p$ exerted by the legs and external efforts on the platform is reciprocal to the twist $\mathbf{t}_s$ of the uncontrollable motion in the Type 2 singularity.

In order to better understand this phenomenon, let us consider the Five-bar mechanism depicted in Fig. 1. A Five-bar mechanism is a planar parallel mechanism composed of two actuators located at the revolute joints located at points A_1 and A_2 and three passive revolute joints at points B_1, B_2 and C.

It is considered that the mechanism is not moving and that the gravity effects are canceled. A force $\mathbf{f}$ is applied on the mobile platform. A simple analysis of the effort transmission shows that the reactions in the passive joints located at points B_1 and C (B_2 and C, resp.) must be colinear to the vector $\overrightarrow{B_1C}$ ($\overrightarrow{B_2C}$, resp.) for any mechanism configurations and that $\mathbf{f} = \mathbf{r}_1 + \mathbf{r}_2$ (with $\mathbf{r}_i$ the force in the joint of the leg i).

Fig. 2 represents the same mechanism in a singular configuration. $\overrightarrow{B_1C}$ is colinear to $\overrightarrow{B_2C}$ and, as a result, $\mathbf{r}_1$ is colinear to $\mathbf{r}_2$. It can be proven that, in such a case, the robot gets an uncontrollable motion along the vector $\mathbf{t}_s$ which is perpendicular to $\overrightarrow{B_1C}$ and $\overrightarrow{B_2C}$ (Fig. 2). To compensate a force $\mathbf{f}$ *which is not colinear* to $\mathbf{r}_1$ and $\mathbf{r}_2$ (i.e. for which the criterion (8) is not respected as $\mathbf{t}_s^T\mathbf{f}$ will be different from zero in this case), the reactions $\mathbf{r}_1$ and $\mathbf{r}_2$ must have infinite norms. If the force $\mathbf{f}$ is colinear to $\mathbf{r}_1$ and $\mathbf{r}_2$ (i.e. the criterion (8) is respected as $\mathbf{t}_s^T\mathbf{f} = 0$ in this case), the reactions $\mathbf{r}_1$ and $\mathbf{r}_2$ will have finite norms.

This simplified problem gives an insight onto the general theory presented in this section.

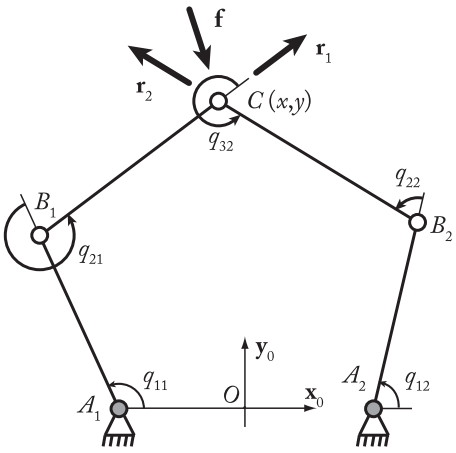

Fig. 1. Kinematic chain of the Five-bar mechanism.

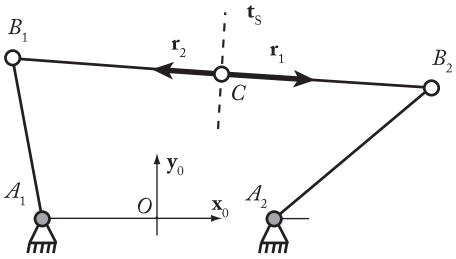

Fig. 2. 5R mechanism in a singular position.

2.3. Generation of a robust trajectory for crossing a Type 2 singularity

In order to cross a singularity without a torque discontinuity, the mechanism has to follow a trajectory which respects criterion (8) on the singularity locus. Theoretically, the dynamic model degenerates only at the singularity locus; however, *numerically* matrix $\mathbf{A}$ is singular in a space around the singularity locus. In order to prevent this degeneracy around the singularity, the trajectory has to respect the criterion (8) not only on the singular point, but also around it.

Trajectory generation is achieved using polynomials, whose degree can vary. Indeed, in order to guarantee that criterion (8) is respected around the singularity locus (contrary to what was done previously), it is proposed *in this work* to vanish the criterion (8) and n of its derivatives:

$$\mathbf{t}_s^T\frac{d^i\mathbf{w}_p}{dt^i} = 0, \quad i = 1, \ldots, n, \tag{9}$$

where d^i/dt^i represents the derivative to the ith degree.

To the best of our knowledge, this is the first time that such criteria have been proposed. Generating a trajectory based on these criteria (9) enables us to increase robustness to model uncertainties and control error around the singularity.

The determination of the derivative order is directly linked to the length of the portion of the trajectory which should respect the criterion (8). The higher this order is, the longer will be the portion of the trajectory impacted by this new criterion. On the other hand, if this order is too low the controller may switch back to the complete dynamic model before exiting the area where $\mathbf{A}$ is numerically bad conditioned. Experimental results show that nullifying the first two derivatives of the criterion is sufficient on our prototype (the singularity crossing is robust and yet the trajectory is not impacted a lot by this criterion).

The next section will present the control law used to enable singularity crossing.

3. Control law dedicated to Type 2 singularity crossing

3.1. Computed torque control

The computed torque control (*CTC*) (Paccot, Andreff, & Martinet, 2009; Spong et al., 2006) is an advanced control law which computes the input torques that the actuators must apply to the mechanism in order to follow a given trajectory. It is based on the dynamic model presented in Section 2.

As for any type of control law in robotics, the aim of the *CTC* controller is to minimize the error in either joint or task space. Since near to a Type 2 singularity the kinematic matrix **A** is singular, it is not possible to compute the Cartesian velocities **v** from the joint velocities $\dot{\mathbf{q}}$ using the *DKM* (Direct kinematic Model). Furthermore, on industrial robots on-board sensors usually measure joint space values, and therefore only the joint space control law should be used to cross Type 2 singularities.

According to Spong et al. (2006) and Ghorbel, Chételat, and Longchamp (1994), the dynamic model of the mechanism satisfies

$$\boldsymbol{\tau} = \mathbf{w}_b + \mathbf{J}^T \mathbf{w}_p = \mathbf{M}\ddot{\mathbf{q}} + \mathbf{H} \tag{10}$$

where

- **H** is the $[n \times n]$ positive-definite inertia matrix. This matrix depends on the actuator's coordinates **q** and the mobile platform Cartesian coordinates **x**,
- **H** is the $[n \times 1]$ matrix regrouping the gravitational terms and the centrifugal and Coriolis terms. This matrix depends on the actuator's coordinates and speed **q** and $\dot{\mathbf{q}}$,

This expression of the dynamic model is obtained by substituting every term depending on the platform coordinates by terms depending only on the actuators' coordinates (using the closed-loop equations). Even though this model is not linear regarding the position and the velocities of the mechanism, it is linear regarding its acceleration. Therefore, by replacing the angular acceleration $\ddot{\mathbf{q}}$ in Eq. (10) by an adapted control signal **u**, the dynamics of the system is linear with respect to the control variable :

$$\begin{aligned} \tau &= \mathbf{M}\mathbf{u} + \mathbf{H} \\ \mathbf{u} &= \ddot{\mathbf{q}} \end{aligned} \tag{11}$$

A double integrator between the control signal and the joint variables appears and thus, only a *PD* control law is used to impose the control signal:

$$\mathbf{u} = \ddot{\mathbf{q}}_\mathbf{d} + \mathbf{K}_d(\dot{\mathbf{q}}_\mathbf{d} - \dot{\mathbf{q}}) + \mathbf{K}_p(\mathbf{q}_\mathbf{d} - \mathbf{q}) \Rightarrow \ddot{\mathbf{e}} + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e} = 0 \tag{12}$$

where

- **u** is a $[n \times 1]$ vector.
- $\mathbf{q}_d$ (respectively $\dot{\mathbf{q}}_d$ and $\ddot{\mathbf{q}}_d$) is the desired joint position (respectively velocity and acceleration),
- **q** (respectively $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$) is the current measured joint position (respectively velocity and acceleration).
- **e** (respectively $\dot{\mathbf{e}}$ and $\ddot{\mathbf{e}}$) is the position error (respectively speed and acceleration) in the joint space ($\mathbf{e} = \mathbf{q_d} - \mathbf{q}$).
- $\mathbf{K_p}$ and $\mathbf{K_d}$ are the two gain tunings that do not depend on the time.

This control signal is a classic second-order control law.

This control input converts a complicated nonlinear controller design problem for a linear system consisting of n decoupled subsystems. Of course, this controller is based on the dynamic model of the mechanism. If this model is not accurate, the tracking error can therefore be important, but the control signal still guarantees that this tracking error respects Eq. (12) and therefore tends to zero with the desired second order dynamics.

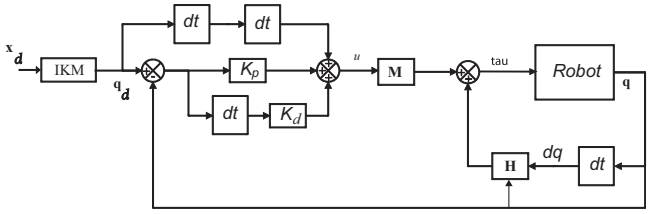

Fig. 3. Classic computed torque control law.

Consequently, *CTC* computes the input torques that ensure the second order dynamics on the tracking error:

$$\boldsymbol{\tau} = \mathbf{M}(\ddot{\mathbf{q}}_d + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}) + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) \tag{13}$$

Fig. 3 represents a classic computed torque control law applied to a parallel mechanism whose dynamic model can be expressed by Eq. (10).

The robustness and stability of the *CTC* control law have been studied in Samson (1987). In this paper, the authors show that the *CTC* control law is stable if the matrix **H** is positive definite. However, in our case, **H** represents the positive definite inertia matrix of the robot (Merlet, 2006), which is always positive definite. Therefore, the proposed *CTC* control law is stable.

It should be noted that the vector of positions in the task space **x** is necessary to compute matrices **M** and **H**. However, in industrial cases sensors only measure the vector of positions in the joint space **q**. Therefore the *DKM* is required. Unfortunately, when planning a Type 2 singularity crossing trajectory, the mechanism changes its assembly mode, and so the solution of the *DKM* has to change. To do so, the controller needs the information that the mechanism has changed its assembly mode, which cannot be deduced from the joint positions. Experimentally, the most reliable solution is to choose the *DKM* solution based on the desired trajectory.

It should be mentioned that, when using the proposed *CTC* for crossing Type 2 singularities, the trajectory planned in order to respect the crossing criterion (Section 2) will be different from the real one, due to errors in the dynamic model. As a result, numerically speaking, the *crossing criterion (8) will never be respected* and the computed torque control could send infinite torques to the robot, preventing singularity crossing due to the inversion of the matrix **A**. The proposed control law avoids this problem and allows parallel mechanism to safely cross singularities.

3.2. Multi-model control law

3.2.1. Robot dynamic models dedicated to the multi-model control law

As explained previously, the complete dynamic model can numerically degenerate when the mechanism approaches a singularity. To avoid these numerical issues, the proposed solution is to plan a trajectory respecting *around the singularity locus* the criterion

$$\begin{aligned} &\mathbf{w}_p = 0 \Rightarrow \mathbf{t}_s^T 0 \mathbf{w}_p = 0 \\ &\frac{d^i}{dt^i}\mathbf{w}_p = 0 \Rightarrow \mathbf{t}_s^T \frac{d^i}{dt^i}\mathbf{w}_p = 0, \quad i = 1, \ldots, n \end{aligned} \tag{14}$$

This new criterion still guarantees that (9) and (12) are satisfied. Vanishing $\mathbf{w}_p$ leads to the deletion of the part of the dynamic model which is directly linked to the kinematic coupling and which degenerates near the Type 2 singular configurations. Therefore if this new criterion is perfectly guaranteed, the

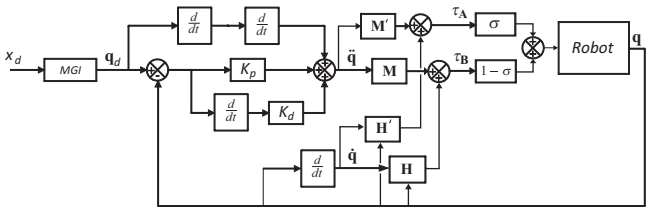

Fig. 4. Multi-model computed torque control law.

mechanism can cross the singularity without having its dynamic model degenerating and with its platform being controlled.

Of course during the real robot displacement, numerically $\mathbf{w}_p$ will not be null, but such a new criterion enables the implementation of a multi-model control law. The multi-model *CTC* law presented in this paper consists of using two models (Fig. 4):

- *Model* 1: The complete dynamic model, as long as matrix **A** introduced in Section 2.1 is invertible:

$$\boldsymbol{\tau}_\mathbf{A} = \mathbf{w}_b(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}) + \mathbf{J}^T \mathbf{w}_p(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}) \tag{15}$$

  In order to control the system, the control law presented in Eq. (13) is applied:

$$\boldsymbol{\tau}_\mathbf{A} = \mathbf{M}(\ddot{\mathbf{q}}_d + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}) + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) \tag{16}$$

- *Model* 2: A reduced dynamic model which cannot degenerate when the mechanism is close to a singular position:

$$\boldsymbol{\tau}_\mathbf{B} = \mathbf{w}_b(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}) \tag{17}$$

  Once again, this second model is used to control the mechanism using the control law presented in Eq. (13):

$$\boldsymbol{\tau}_\mathbf{B} = \mathbf{M}'(\ddot{\mathbf{q}}_d + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}) + \mathbf{H}'(\mathbf{q}, \dot{\mathbf{q}}) \tag{18}$$

By definition, this new matrix $\mathbf{M}'$ is still positive definite. Therefore, this control law is stable (Samson, 1987).

The second dynamic model is used to compute input torques only when the trajectory has been planned in order to have $\mathbf{w}_p = 0$. Considering that the control law is correctly adjusted, the effective trajectory is close enough to the desired one and therefore the hypothesis $\mathbf{w}_p = 0$ is acceptable. Once the mechanism is far enough from the singularity locus, i.e. the criterion (14) is no longer respected, the control switches back to the complete dynamic model and the mechanism can finish its trajectory. The index choice that determines when the control law can switch from one model to another is discussed in the next paragraph.

3.2.2. *Index choice for model switching*

In order to cross the singularity locus, the controller requires an index which defines the moment when *Model* 2 has to be used. The discussion about the best indicator of singularity proximity is a well-known problem (Merlet, 2006; Glazunov et al., 2012; Voglewede & Ebert-Uphoff, 2004). However, since Model 2 is valid only when $\mathbf{w}_p$ is null, here the index used will be based on the value of $\|\mathbf{w}_p\|$.

Therefore, based on the assumption that the control error is well regulated, the desired value of $\mathbf{w}_p$ (computed according to the reference trajectory) is used as the switching variable. This prevents issues due to the computation of the actuators' accelerations.

3.2.3. *Switching function: σ*

In order to guarantee torque continuity when the control switches from one model to another, the input torques are computed

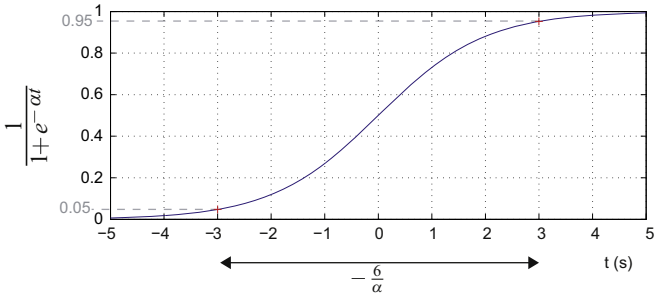

Fig. 5. Variation of the logistic function and influence of parameter α.

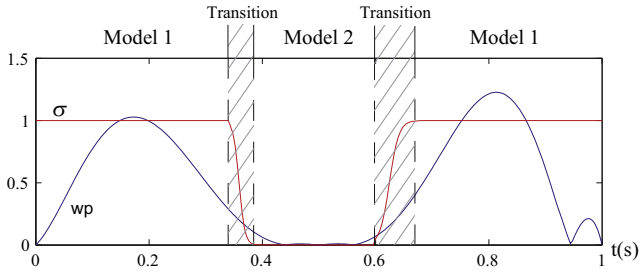

Fig. 6. Variation in sigma based on the value of $\mathbf{w}_p$ when crossing a singularity.

using the logistic function σ (shown in Fig. 5) such that

$$\boldsymbol{\tau} = \sigma\boldsymbol{\tau}_\mathbf{A} + (1-\sigma)\boldsymbol{\tau}_\mathbf{B} \tag{19}$$

This function σ is equal to:

- 1 when the first model must be used,
- 0 when the second model must be used, i.e. when the mechanism is close to a singular position,
- $1/(1+e^{-\alpha t})$ or $1-1/(1+e^{-\alpha t})$ when switching between the two models, where t is the current time, with α denoting the length of the transition phase.

The logistic function σ varies continuously between 0 and 1, which prevents any torque discontinuity. This α parameter is computed based on the value of the derivative of $\mathbf{w}_p$ such that σ is null when $\mathbf{w}_p$ and its derivatives are null, i.e. when the mechanism is in a singular position, as presented in Fig. 6.

The resulting control law can be written as

$$\begin{aligned}\tau &= (\sigma\mathbf{M} + (1-\sigma)\mathbf{M}')(\ddot{\mathbf{q}}_d + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}) + \sigma\mathbf{H} + (1-\sigma)\mathbf{H}' \\ &= \underline{\mathbf{M}}(\mathbf{q}, \sigma)\mathbf{u} + \underline{\mathbf{H}}(\dot{\mathbf{q}}, \mathbf{q}, \sigma)\end{aligned} \tag{20}$$

In this equation, the matrix $\underline{\mathbf{M}}$ is defined as $\underline{\mathbf{M}} = \sigma\mathbf{M} + (1-\sigma)\mathbf{M}'$, where the matrices $\mathbf{H}$ and $\mathbf{M}'$ are both positive-definite. Moreover, since σ and $(1-\sigma)$ are real positive numbers, this new matrix $\underline{\mathbf{M}}$ is also positive-definite. Therefore, based on the stability proof of the computed torque control proposed in Samson (1987), the proposed multi-model control law is stable.

The next section presents experimental results for Type 2 singularity crossing using this multi-model control law on a planar Five-bar mechanism.

4. Case study

This section presents the 5R mechanism designed in IFMA. This mechanism was used in order to validate the proposed approach. Experimental results are detailed in Section 4.5.

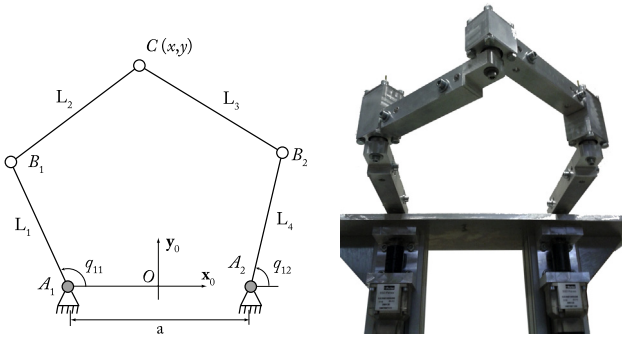

Fig. 7. Five-bar mechanism designed and manufactured at IFMA.

Table 1
Five-bar mechanism: geometric parameters.

Parameter	a	L_1	L_2	L_3	L_4
Value (m)	0.2822	0.2130	0.1888	0.1878	0.2130
Precision (m)	1×10^{-5}	1×10^{-5}	1×10^{-5}	1×10^{-5}	1×10^{-5}

4.1. Presentation of the Five-bar mechanism

A Five-bar mechanism is a planar parallel mechanism composed of two actuators located at the revolute joints positioned at points A_1 and A_2 and 3 passive revolute joints at points B_1, B_2 and C (Fig. 7). The mechanism used in this work was designed so that it can reach all the workspace positions without any collision between the proximal and the distal legs.

The mechanism and its parameters are presented in Fig. 7. The link dimensions were calibrated using a Laser Tracker (Table 1). The sight was fixed on the rotation axis of each joint. Using various positions of the mechanism, the length of each link has been computed from the average of the distances measured on each position.

The mechanism itself was made out of shaped aluminum, the joints are made of bearings which avoid having high friction effects in the passive joints. The distal and proximal legs are in different planes and therefore cannot collide with each other. The mechanism is actuated using 2 PARVEX brushless servo motors and two 1/15 reducers. The servo motors are controlled using an Adept control cabinet working with a C++ software developed by Adept FRANCE: CIDE. The dynamic control law has sampling frequency of 250 Hz. Finally, the only sensors used are those provided by the brushless motors.

4.2. Gain tuning

The proportional and derivative gains were tuned based on the mechanism's natural frequency (Khalil & Dombre, 2004). This frequency was retrieved using a ring-out procedure: the mechanism was excited using an impedance hammer, and its response was recorded using 5 accelerometers. The first natural frequency of the Five-bar mechanism in its isotropic configuration (when links BC and CD are perpendicular) is 34.2 Hz.

For a given control bandwidth fixed by a frequency ω both gains are adjusted as

$$K_p = \omega^2, \quad K_d = 2\xi\omega \tag{21}$$

where ξ is a damping coefficient, usually fixed as 1 to have a critically damped system. To guarantee that the gains do not bring the system into the neighborhood of the instability domain, the chosen frequency must be smaller than the natural resonant frequency (Khalil & Dombre, 2004). Therefore a frequency of $\omega = \omega_r/2$ was chosen, resulting in the following gain values:

$$K_p \approx 289, \quad K_d \approx 34 \tag{22}$$

4.3. Dynamic modeling and identification

A full dynamic model of the robot was computed using the methodology presented in Briot and Gautier (2012) and its identification was performed using a weighted least square method based on the use of exciting trajectories, followed by a classic geometrical control law (Gautier, 1997). The identification resulted in the following model that fully describes the robot dynamics of the studied mechanism:

$$\tau = m_3 \mathbf{J}^T \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} + \begin{pmatrix} zz_1\ddot{q}_1 \\ zz_2\ddot{q}_2 \end{pmatrix} + \begin{pmatrix} f_{v1}\dot{q}_1 \\ f_{v2}\dot{q}_2 \end{pmatrix} + \begin{pmatrix} f_{s1}\,\mathrm{sign}(\dot{q}_1) \\ f_{s2}\,\mathrm{sign}(\dot{q}_2) \end{pmatrix} \tag{23}$$

where

- m_3 is a mass equivalent located on the end effector; $m_3 = 0.40 \pm 0.02$ kg;
- zz_1 and zz_2 are rotational equivalent inertial terms, respectively on the first and second actuator; $zz_1 = 1.83\times10^{-2} \pm 6.97\times10^{-4}$ kg m^2; $zz_2 = 1.96\times10^{-2} \pm 6.60\times10^{-4}$ kg m^2;
- f_{s1} is a Coulomb friction term on the first actuator (respectively f_{s2} on the second actuator); $f_{s1} = 2.94 \pm 0.10$ N m; $f_{s2} = 2.95 \pm 0.09$ N m, and sign( ) is the function returning 1, -1 or 0 depending of the sign of the input;
- f_{v1} is a viscous friction term on the first actuator (respectively f_{v2} on the second actuator); $f_{v1} = 6.76 \pm 0.018$ N m s $f_{v2} = 6.75 \pm 0.17$ N m s.

This identified dynamic model is related to Eq. (3) by

$$\mathbf{w}_p = m_3 \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix},$$

$$\mathbf{w}_b = \begin{pmatrix} zz_1\ddot{q}_1 \\ zz_2\ddot{q}_2 \end{pmatrix} + \begin{pmatrix} f_{v1}\dot{q}_1 \\ f_{v2}\dot{q}_2 \end{pmatrix} + \begin{pmatrix} f_{s1}\,\mathrm{sign}(\dot{q}_1) \\ f_{s2}\,\mathrm{sign}(\dot{q}_2) \end{pmatrix} \tag{24}$$

It should be noted that the friction terms in both passive joints are insignificant and therefore the identification routine returned null values.

Different trajectories were computed in order to cross-validate the dynamic model identified. For each trajectory, the positions, velocities and input torques were retrieved for both actuators. Using the identified dynamic model and the measured positions and velocities, the input torques can be computed and compared to the measured ones, as illustrated in Fig. 8 which represents the input torques measured and computed along a trajectory. This trajectory, and therefore the evolution of the input torques, has been defined randomly and contains both high and low acceleration phases.

4.4. Control law implementation

The Five-bar mechanism is controlled by an industrial control architecture developed by ADEPT with an open architecture. This control architecture allows the user to control the mechanism either in position, speed or torque, using a C/C++ software developed by *ADEPT France*: CIDE. This software was designed mostly for position control; therefore safety elements preventing mostly physical damage had to be developed for the computed torque control law.

The dynamic model identified contains accelerations in both the joint space and the task space. Therefore, in order to express the dynamic model as in Eq. (10), the task space acceleration has to be expressed as a function of the joint space acceleration. This

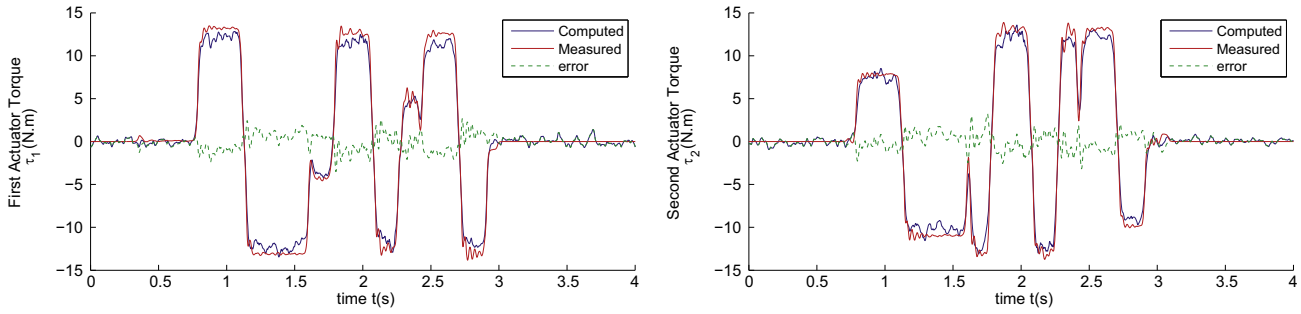

Fig. 8. Verification of the identified dynamic model.

Table 2
Coefficients of the polynomials for each trajectory used on the Five-bar mechanism.

Coefficient	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9
$x(t)=\sum_{i=1}^{9} a_i t^i$	0	0	0	−0.25173	0.80819	−0.98914	0.21007	0.59110	−0.48834
$y(t)=\sum_{i=1}^{9} a_i t^i$	0.1	0	0	4.44755	−14.27911	17.47601	−3.71157	−10.44354	8.62799

can be done by differentiating the kinematic model:

$$\mathbf{v}=\mathbf{J}\dot{\mathbf{q}} \Rightarrow \dot{\mathbf{v}}=\dot{\mathbf{J}}\dot{\mathbf{q}}+\mathbf{J}\ddot{\mathbf{q}} \tag{25}$$

By substituting (25) into (23) one can obtain the dynamic model used for the computed torque control law as presented in Eq. (10):

$$\tau=\mathbf{M}\begin{pmatrix}\ddot{q}_1\\ \ddot{q}_2\end{pmatrix}+\mathbf{H} \tag{26}$$

where

$$\mathbf{M}=m_3\mathbf{J}^T\mathbf{J}+\begin{pmatrix}zz_1 & 0\\ 0 & zz_4\end{pmatrix}$$
$$\mathbf{H}=m_3\mathbf{J}^T\dot{\mathbf{J}}\begin{pmatrix}\dot{q}_1\\ \dot{q}_2\end{pmatrix}+\begin{pmatrix}f_{v1}\dot{q}_1+f_{s1}\,\mathrm{sign}(\dot{q}_1)\\ f_{v2}\dot{q}_2+f_{s2}\,\mathrm{sign}(\dot{q}_2)\end{pmatrix} \tag{27}$$

Around the singularity locus, the *CTC* computes the input torques using the reduced dynamic model defined such as

$$\tau=\mathbf{M}'\begin{pmatrix}\ddot{q}_1\\ \ddot{q}_2\end{pmatrix}+\mathbf{H}' \quad \text{where} \quad \mathbf{M}'=\begin{pmatrix}zz_1 & 0\\ 0 & zz_4\end{pmatrix}, \quad \mathbf{H}'=\begin{pmatrix}f_{v1}\dot{q}_1+f_{s1}\,\mathrm{sign}(\dot{q}_1)\\ f_{v2}\dot{q}_2+f_{s2}\,\mathrm{sign}(\dot{q}_2)\end{pmatrix} \tag{28}$$

Finally, in order to compute the dynamic model of the robot and the *CTC* control law on-line, the actuator's positions and speeds were filtered by using the oversampling method at 1 KHz (the control law running at 250 Hz).

4.5. Experimental results

4.5.1. Generation of a crossing trajectory

Crossing trajectories were generated using two polynomials P_x and P_y such that

$$x=P_x(x_f-x_0)+x_0, \quad y=P_y(y_f-y_0)+y_0 \tag{29}$$

where

- $x(t_0)=y(t_0)=\dot{x}(t_0)=\dot{y}(t_0)=\ddot{x}(t_0)=\ddot{y}(t_0)=0$,
- $x(t_f)=y(t_f)=\dot{x}(t_f)=\dot{y}(t_f)=\ddot{x}(t_f)=\ddot{y}(t_f)=0$,

They are both 8th order polynomials, corresponding to 8 conditions on each axis: two conditions for the initial position and speed, two for the final position and speed, one for the singular position and three to guarantee that the singularity crossing criterion (14) is respected around the singularity locus (Briot et al., 2008).

Table 2 details the coefficients of those two polynomials.

Fig. 9 represents a crossing trajectory in the task space as well as the evolution of the task space coordinates along this trajectory and the evolution of the dynamic criterion (14) for

$$\begin{aligned} &x_0=0, \quad y_0=0.1,\\ &x_s=0.05434, \quad y_s=0.2,\\ &x_f=0.1, \quad y_f=0.34, \end{aligned} \tag{30}$$

4.5.2. Type 2 singularity crossing with classic computed torque control law

Theoretically, the computed torque control law can cross a singularity following a trajectory generated as explained in Section 4.5.1. However, numerically the dynamic model degenerates (e.g. Section 3.1) and the control law computes infinite torques. Fig. 10 presents the experimental results when following a crossing trajectory computed as presented in Section 4.5.1 with a classic *CTC*.

When the mechanism approaches the singularity locus, the input torques τ become discontinuous and tend to infinity. Therefore, even if the mechanism should reach the singularity locus after 0.5 s, it can be seen in Fig. 10 that the input torques tend to infinity before reaching the singularity. The result of the robot displacement is shown in Fig. 11. To avoid causing physical damage to the actuator, a security stops the mechanism, which remains blocked inside the singularity, resulting in an increase of the articular error.

4.5.3. Type 2 singularity crossing with the multi-model CTC: results and process repeatability

This section presents the results of Type 2 singularity crossing for different trajectories computed according to the method presented in Section 2.3 and in the previous paragraph.

Fig. 12 shows the input torques generated by the computed torque control law along different crossing trajectories from one assembly mode to another, as well as the desired trajectory and the control error. For each trajectory, the mechanism crosses the singularity without torque discontinuity. Fig. 13 presents different configurations of the motion of the mechanism along the first trajectory at different time instants.

The multi-model control law leads to an increase of the error around the singularity locus. Therefore, when the control law switches back to the complete dynamic model, the input torques

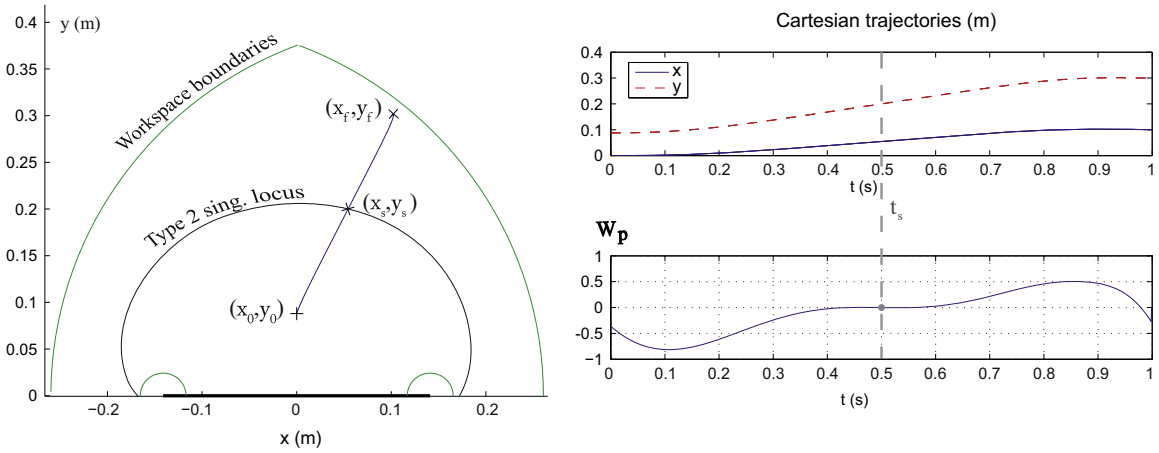

Fig. 9. Example of 8th order polynomial trajectory crossing a Type 2 singularity locus.

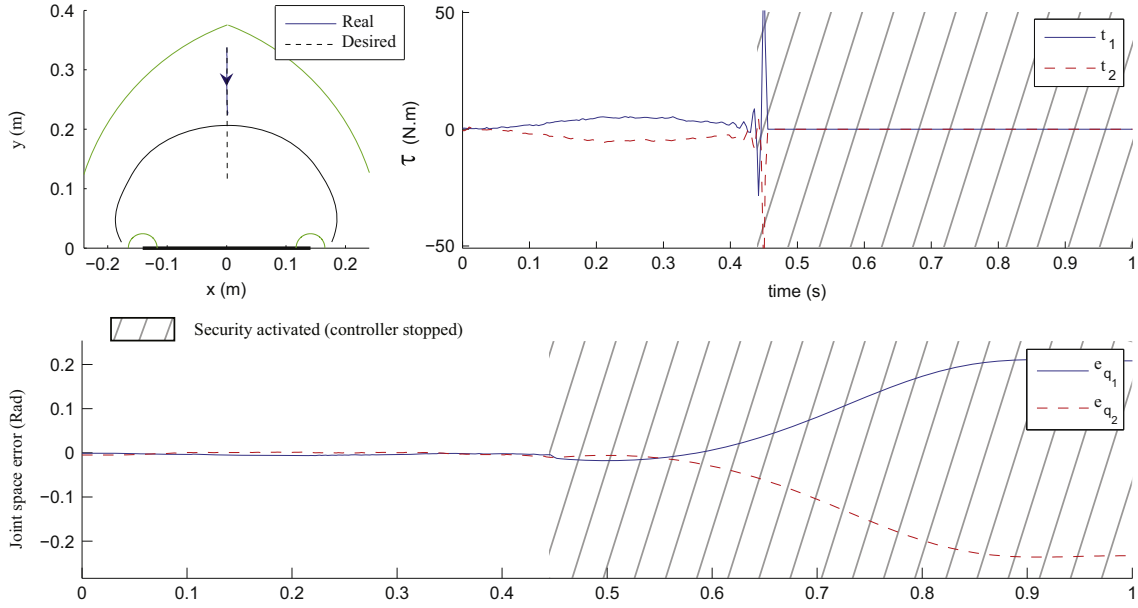

Fig. 10. Measured input torques, tracking error and trajectory when following a crossing trajectory with a classic computed torque control law.

can significantly increase in order to nullify this error. This can be seen for the first trajectory at 0.7 s.

All three trajectories represented in Fig. 12 were planned to cross the singularity at 0.5 s and end at 1 s. For each trajectory, the first figure illustrates the desired trajectory in the task space and the Type 2 singularity of the mechanism.

Each trajectory was run five times to test the robustness of the proposed controller. Moreover, the starting and ending points were chosen randomly and neither those points nor the crossing direction had any effect on the singularity crossing.

Fig. 14 represents the Cartesian coordinates of the mobile platform along 5 other examples of crossing trajectories, each of them having different starting and ending points and crossing in either direction. It can be seen that even if the starting and ending points are chosen randomly, the multi-model *CTC* is able to cross the singularity successfully.

During our different experiments using this controller, the robot has always successfully crossed the singularity locus without any difficulty. Thus the new controller proposed, coupled with the new dynamic criterion, enables a parallel mechanism to cross a Type 2 singularity without torque discontinuity. Even though the multi-model control law induces an overshoot, it does not impact the singularity crossing, which is totally robust with respect to the desired trajectory. This methodology can therefore considerably increase the mobile platform reachable space of almost any parallel mechanism.

4.5.4. *Discussion on the tracking error*

During the singularity crossing, the tracking error tends to significantly increase. This is due to the multi-model controller used. If the controller was perfect, when approaching the singularity locus the dynamic term $\mathbf{w_p}$ would be null. Therefore, the second model used in the multi-model controller would describe perfectly the dynamic of the mechanism, and the tracking error would still be null. However, the tracking error cannot be null which involves that $\mathbf{w_p}$ is not null. Considering that the controller is correctly configured, its value is negligible compared to $\mathbf{w_b}$, but still it slightly changes the dynamic of the mechanism. Therefore, the second model does not completely describe the dynamic of the mechanism which results in an increase of the tracking error.

The main objective of the singularity crossing is to access different assembly modes, and thus increase the workspace size. Despite the fact that the tracking error could increase during the singularity crossing, the mechanism has always successfully changed its assembly mode by using the multi-model control law. However, the use of an advanced controller (such as an adaptive or

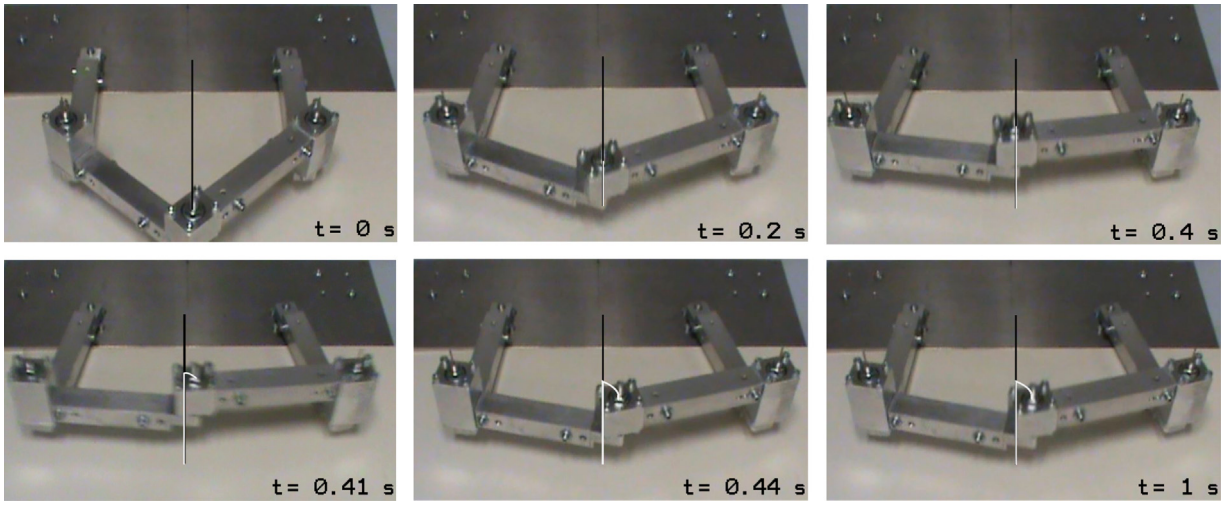

Fig. 11. Images of the mechanism trying to cross a singularity with a classic *CTC* control law.

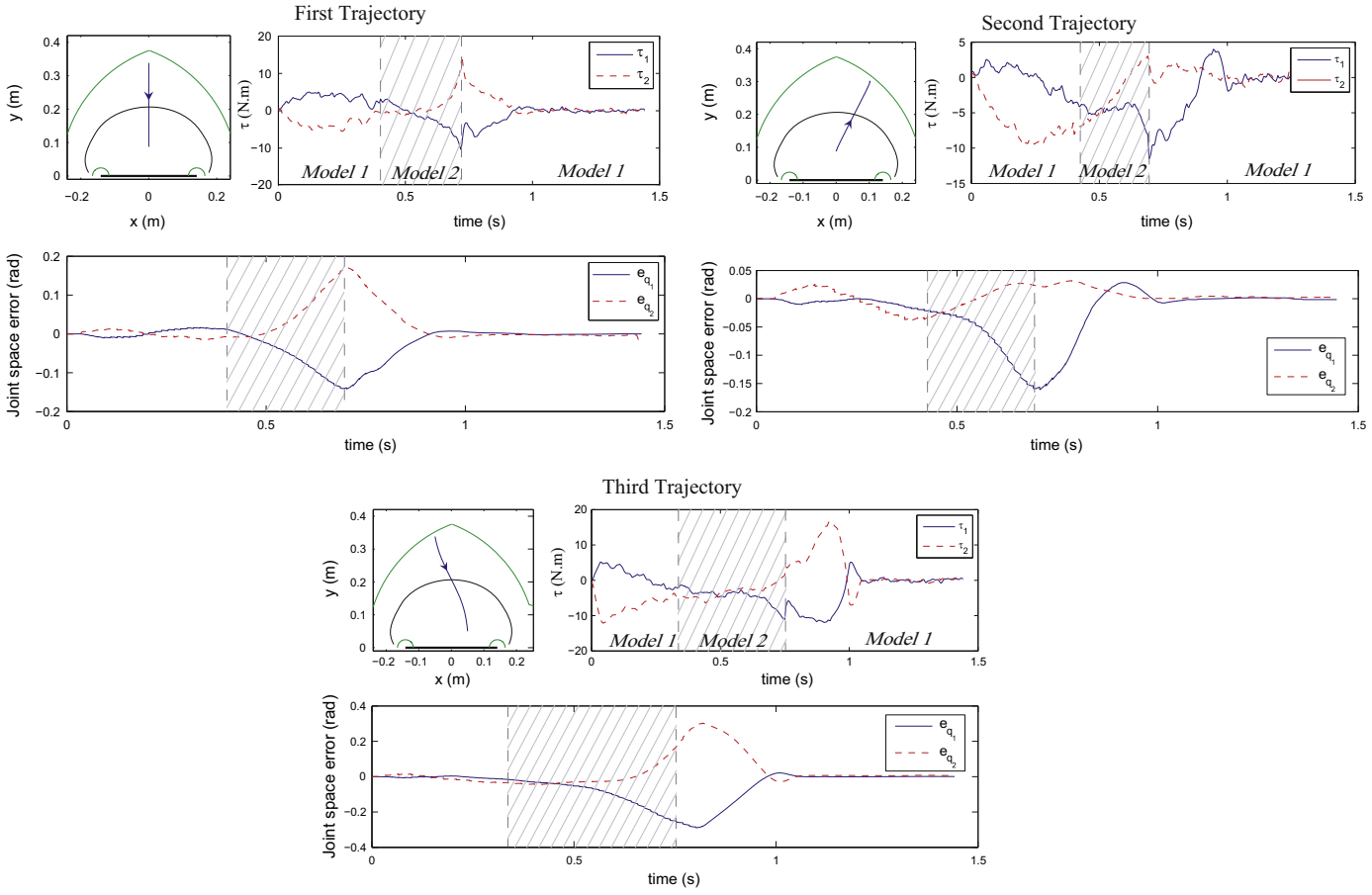

Fig. 12. Different crossing trajectories and corresponding input torque applied.

predictive control law) coupled with the proposed multi-model controller is under investigation. The first results concerning this adaptive approach are promising, however this development is out of the scope of the present paper.

5. Conclusion

The presence of singularities in the workspace of parallel robots greatly reduces their effector's reachable positions. Several solutions have been proposed to either increase the workspace size (*e.g.* changing the assembly mode) or bypass the singularity problem (*e.g.* design mechanisms without singularities). A promising solution consists of changing the assembly mode by crossing Type 2 singularities. This solution requires the definition of an optimal trajectory that must be tracked by a dynamic controller. However, the classic dynamic control laws are unsuitable. This solution requires that the crossing trajectory respect a dynamic criterion at the singularity locus, which prevents the dynamic model from degenerating. However, if this criterion is not perfectly verified (which is always the case

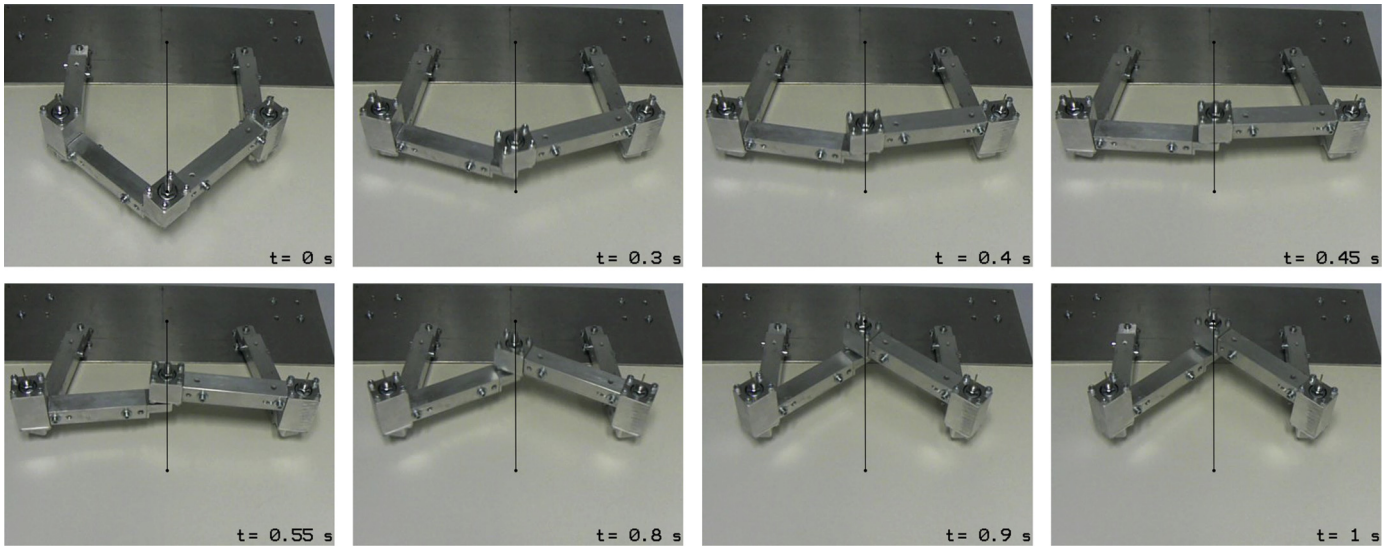

Fig. 13. Trajectory reproduction during a singularity crossing.

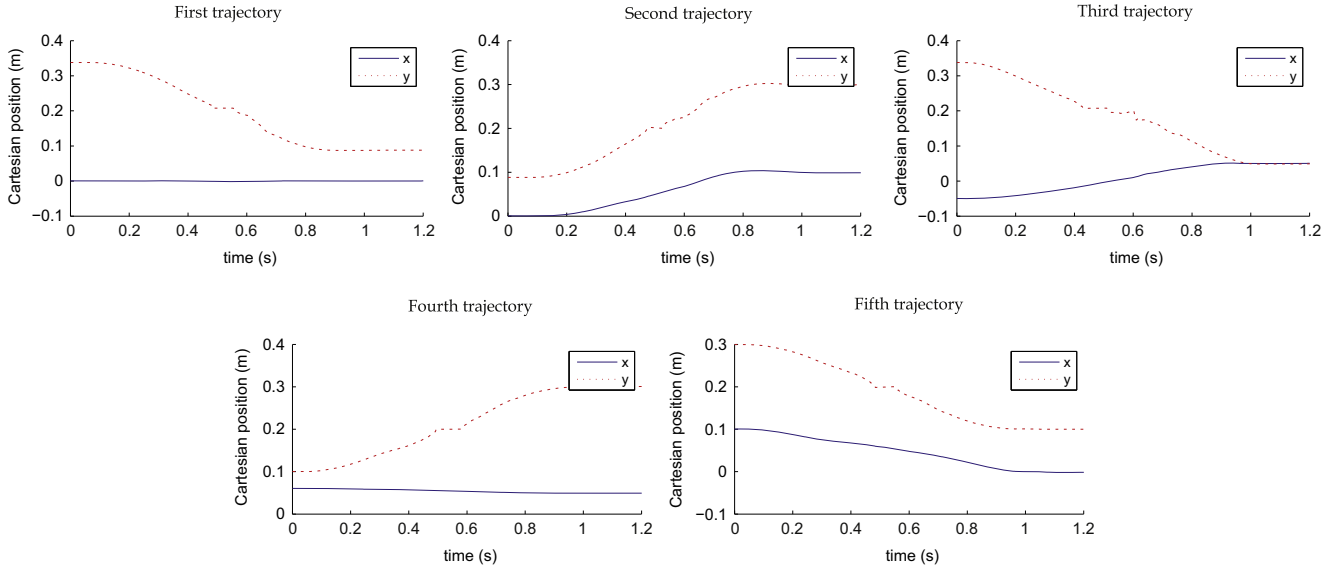

Fig. 14. Cartesian coordinates along 5 different crossing trajectories.

numerically) the kinematic and dynamic model degenerates, resulting in the computation of infinite torques.

This paper proposes a multi-model controller dedicated to Type 2 singularity crossing which avoids dynamic model degeneration near the Type 2 singularity locus. It does so by generating a trajectory that nullifies the degenerating part of the dynamic model around the singularity locus. A dynamic multi-model controller is used to follow the trajectory generated this way; the controller switches to a simplified dynamic model when the mechanism is close to a singularity. This prevents the dynamic model from degenerating even though the desired trajectory is not perfectly tracked, and therefore allows the mechanism to cross the Type 2 singularity without torque discontinuity.

This new controller was validated experimentally on a Five-bar mechanism. It was compared to a classic dynamic control law without multi-model that was not able to cross the singularity, while the multi-model *CTC* was validated on various singularity crossing trajectories, thus proving its experimental robustness.

Acknowledgments

This work was sponsored by the French government research program “Investissements d'avenir” through the RobotEx Equipment of Excellence (ANR-10-EQPX-44) and by the French Institute for Advanced Mechanics.

References

Arakelian, V., Briot, S., & Glazunov, V. (2008). Increase of singularity-free zones in the workspace of parallel manipulators using mechanisms of variable structure. *Mechanism and Machine Theory*, *43*(9), 1129–1140. http://dx.doi.org/10.1016/j.mechmachtheory.2007.09.005.

Briot, S., & Arakelian, V. (2010). On the dynamic properties of rigid-link flexible-joint parallel manipulators in the presence of Type 2 singularities. *Journal of Mechanisms and Robotics*, *2*(2), 021004. http://dx.doi.org/10.1115/1.4001121.

Briot, S., Arakelian, V., & Chablat, D. (2008). Optimal force generation in parallel manipulators for passing through the singular positions. *International Journal of Robotics Research*, *27*(8), 967–983. http://dx.doi.org/10.1177/0278364908094403.

Briot, S. & Gautier, M. Global Identification of Drive Gains and Dynamic Parameters of Parallel Robots – Part 1: Theory," Proceedings of the 19th CISM-IFToMM Symposium on Robot Design, Dynamics, and Control (ROMANSY 2012), 2012.

Conconi, M., & Carricato, M. (2009). A new assessment of singularities of parallel kinematic chains. In *Proceedings of international conference on robotics and automation* (Vol. 25(4), pp. 757–770).

Craig, J. J., & Hall, P. P. (2005). *Introduction to robotics*. Upper Saddle River, NJ, USA: Pearson/Prentice Hall.

Gautier, M. (1997). Dynamic identification of robots with power model. In *Proceedings of international conference on robotics and automation* (Vol. 3, pp. 1922–1927).

Ghorbel, F., Chételat, O. & Longchamp, R. (1994). A reduced model for constrained rigid bodies with application to parallel robots. In *Proceedings of the IFAC symposium on robot control* (3) (pp. 57–62).

Glazunov, V., Arakelian, V., Briot, S., & Rashoyan, G. V. (2012). Speed and force criteria for the proximity to singularities of parallel structure manipulators. *Journal of Machinery Manufacture and Reliability*, *41*(3), 194–199.

Gogu, G. (2004). Structural synthesis of fully-isotropic translational parallel robots via theory of linear transformations. *European Journal of Mechanics A/Solids*, *23* (6), 1021–1039.

Gosselin, C., & Angeles, J. (1990). Singularity analysis of closed-loop kinematic chains. In *Proceedings of international conference on robotics and automation* (Vol. 6(3), pp. 281–290). doi: http://dx.doi.org/10.1109/70.56660.

Ider, S. (2005). Inverse dynamics of parallel manipulators in the presence of drive singularities. *Mechanism and Machine Theory*, *40*(1), 33–44. http://dx.doi.org/10.1016/j.mechmachtheory.2004.05.007.

Khalil, W. & Dombre, E. (2004). *Modeling, identification and control of robots*, Vol. 56. Hermes, ISBN: 190399666X; 978-1903996669. doi: http://dx.doi.org/10.1115/1.1566397.

Kong, K., & Gosselin, C. (2002). A class of 3-dof translational parallel manipulators with linear input–output equations. In *The workshop on fundamental issues and future research directions for parallel mechanisms and manipulators* (pp. 3–4).

Kurtz, R., & Hayward, V. (1992). Multiple-goal kinematic optimization of a parallel spherical mechanism with actuator redundancy. In *Proceedings of International Conference on Robotics and Automation* (Vol. 8, pp. 644–651). doi: http://dx.doi.org/10.1109/70.163788.

Liu, X. J., Wang, J., & Pritschow, G. (2006). Performance atlases and optimum design of planar 5R symmetrical parallel mechanisms. *Mechanism and Machine Theory*, *41*, 119–144. http://dx.doi.org/10.1016/j.mechmachtheory.2005.05.003.

Merlet, J. (2006). *Parallel robots*. Springer ISBN: 978-1-4020-4133-4.

Nahon, M. & Angeles, J. Force optimization in redundantly-actuated closed kinematic chains. In *Proceedings of international conference on robotics and automation*. doi: http://dx.doi.org/10.1109/ROBOT.1989.100103.

Paccot, F., Andreff, F., & Martinet, P. (2009). A review on the dynamic control of parallel kinematic machine: Theory and experiments. *International Journal of Robotics Research*, *28*(3), 395–416.

Rakotomanga, N., Chablat, D., & Caro, S. (2006). Kinetostatic performance of a planar parallel mechanism with variable actuation. In *Advances in robot kinematics: Analysis and design* (pp. 311–320). arxiv:0809.3044 doi: http://dx.doi.org/10.1007/978-1-4020-8600-7_33.

Samson, C. (1987). Robust control of a class of nonlinear systems and applications to robotics. *International Journal of Adaptive Control and Signal Processing*, *1*(1), 49–68.

Spong, M., Hutchinson, S., & Vidyasagar, M. (2006). *Robot modeling and control* (Vol. 3). New York: Wiley.

Voglewede, P., & Ebert-Uphoff, I. (2004). Measuring "closeness" to singularities for parallel manipulators. *Proceedings of international conference on robotics and automation* (pp. 4539–4544).

Zein, M., Wenger, P., & Chablat, D. (2008). Non-singular assembly-mode changing motions for 3-RPR parallel manipulators. *Mechanism and Machine Theory*, *43*(4), 480–490. http://dx.doi.org/10.1016/j.mechmachtheory.2007.03.011.

Zlatanov, D. & Bonev, I. Constraint singularities of parallel mechanisms. In *Proceedings of international conference on robotics and automation.*