



**HAL**  
open science

# Enlarging Parallel Robot Workspace through Type-2 Singularity Crossing

Georges Pagis, Nicolas Bouton, Sébastien Briot, Philippe Martinet

► **To cite this version:**

Georges Pagis, Nicolas Bouton, Sébastien Briot, Philippe Martinet. Enlarging Parallel Robot Workspace through Type-2 Singularity Crossing. Control Engineering Practice, 2015, 39, pp.1-11. <hal-01114340>

**HAL Id: hal-01114340**

**<https://hal.science/hal-01114340v1>**

Submitted on 25 Jun 2019

**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.



HAL Authorization

# Enlarging Parallel Robot Workspace through Type 2 Singularity Crossing: Controller Design and Benchmarking

Georges Pagis<sup>1,2</sup>, Nicolas Bouton<sup>1</sup>, Sebastien Briot<sup>2</sup>, Philippe Martinet<sup>2,3</sup>

{Georges.Pagis, Nicolas.Bouton}@ifma.fr; {Sebastien.Briot, Philippe.Martinet}@ircyn.ec-nantes.fr

---

## Abstract

In order to increase the reachable workspace of parallel robots, a promising solution consists in 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 modelling 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 proposed control law is validated experimentally through a Five-bar planar parallel mechanism.

*Keywords:* Parallel Robots, Computed Torque Control, Multi-Model Approach, Singularities

---

## 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 [1, 2, 3], which divide their workspace into different aspects (each aspect corresponding to one or more assembly modes [4]). The manipulator workspace is therefore usually reduced to only one of these aspects, resulting in a greatly reduced reachable workspace size.

Various type of singularity exist, and for a global overview of the singularity problem the reader is referred to [2]. However, since Type 2 [3] (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 [5, 6] or by creating decoupled mechanisms [7, 8]. 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 [9, 10] or, to reduce costs, the use of mechanisms with variable actuation modes [1, 11]. 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 [12]. 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 [13, 14]. In [13], a physical criterion, obtained through the analysis

---

<sup>1</sup>Institut Pascal, IFMA, MMS department, UMR CNRS 6602, 63000 Clermont-Ferrand, France

<sup>2</sup>Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN), UMR CNRS 6597, 44321 Nantes, France

<sup>3</sup>LUNAM, Ecole Centrale de Nantes, 44321 Nantes, France

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 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.

In order to correctly track a trajectory, only a dynamic control law can be used [15] since only can control the accelerations. Moreover, it is the only kind of control law that takes into account the dynamic properties of the system. Among the different dynamic control loop algorithms, the most efficient one is probably the Computed Torque Control (*CTC*) [15, 16, 17]. It computes the actuator torque of the robot based on its dynamic model and the desired trajectory. However, this control law is sensitive to modeling errors, so the dynamic model must be well identified [23, 24]. Furthermore, to be used when crossing a Type 2 singularity, the dynamic model must not degenerate near singularities, even if the trajectory does not perfectly respect the physical criterion mentioned above, which is unavoidable with usual *CTC*. As a result, in this paper a new multi-model *CTC* (e.g. see [16, 17]) 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 [13].

A first condensed version of this work has been submitted to ICRA 2014. The proposed paper presents detailed explanations on:

- The way to choose a metric for a multi-model approach.
- The validation of the identified dynamic model.
- New experimental results for the comparison between using the usual computed torque control (*CTC*) laws for singularity crossing and using the proposed approach.

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 III, the multi-model *CTC* control law developed for crossing singularities is presented. Section IV 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 position and the speed of the manipulator can be fully described using:

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

Using Lagrangian formalism, the dynamic model of the mechanism can be written as:

$$\boldsymbol{\tau} = \mathbf{w}_b + \mathbf{B}^T \boldsymbol{\lambda}, \quad (1)$$

$$\mathbf{w}_p = \mathbf{A}^T \boldsymbol{\lambda} \quad (2)$$

where

- $\boldsymbol{\tau}$  is the vector of the input efforts,

- $\lambda$  is the vector of the Lagrange multipliers,
- $\mathbf{A}$  and  $\mathbf{B}$  are two matrices deduced from the mechanism loop-closure equations, such that  $\mathbf{A}\mathbf{v} = \mathbf{B}\dot{\mathbf{q}}$  [4],
- $\mathbf{w}_b$  and  $\mathbf{w}_p$  are related to the Lagrangian  $L$  of the system by:

$$\mathbf{w}_b = \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial L}{\partial \mathbf{q}}, \quad \mathbf{w}_p = \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\mathbf{v}}} \right) - \frac{\partial L}{\partial \mathbf{x}} \quad (3)$$

In this expression,  $\mathbf{w}_p$  is the wrench applied to the platform by the legs and the external forces [13].

Then, assuming that matrix  $\mathbf{A}$  can be inverted and by substituting (2) into (1), the general dynamic model of parallel manipulators is obtained [15]:

$$\boldsymbol{\tau} = \mathbf{w}_b + \mathbf{J}^T {}^0\mathbf{w}_p, \quad (4)$$

where

- ${}^0\mathbf{w}_p$  is the expression of the wrench  $\mathbf{w}_p$  in the base frame, i.e.  ${}^0\mathbf{w}_p = \mathbf{D}\mathbf{w}_p$  with  $\mathbf{D}$  the matrix relating the platform twist  $\mathbf{t}$  (expressed in the base frame) to the vector  $\mathbf{v}$  by  $\mathbf{t} = \mathbf{D}\mathbf{v}$  [4],
- $\mathbf{J} = {}^0\mathbf{A}^{-1}\mathbf{B}$  is the Jacobian matrix between 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 [3]:

- **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.

- **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 [19, 2].

If a parallel mechanism is in a singular Type 2 position, matrix  $\mathbf{A}^T$  cannot be inverted in Equation (2). The dynamic model degenerates and therefore cannot be solved. However, as explained in the introduction, it has been proven in [13] 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:

$${}^0\mathbf{A}\mathbf{t}_s = 0 \Leftrightarrow \mathbf{t}_s^T {}^0\mathbf{A}^T = 0 \quad (5)$$

The vector  $\mathbf{t}_s$  represents the twist of the uncontrollable motion of the platform at the singularity locus [4]. Thus, multiplying (2) by  $\mathbf{t}_s^T$  leads to:

$$\mathbf{t}_s^T {}^0\mathbf{A}^T \boldsymbol{\lambda} = 0 \quad (6)$$

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

$$\mathbf{t}_s^T {}^0\mathbf{w}_p = 0 \quad (7)$$

which is the condition for the non-degeneracy of the dynamic model [13].

As a result, if the desired manipulator motion does not guarantee the achievement of a wrench  $\mathbf{w}_p$  that respects condition (7), 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.

### 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 (7) 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. Therefore criterion (7) has to be respected in this space around the singularity locus in order to prevent the dynamic model from degenerating.

Trajectory generation is achieved using polynomials, whose degree can vary. Indeed, in order to guarantee that criterion (7) is respected around the singularity locus (contrary to what was done previously), it is proposed *in this work* to vanish the criterion (7) and  $n$  of its derivatives:

$$\mathbf{t}_s^T \frac{d^i \mathbf{w}_p}{dt^i} = 0 \quad i = 1, \dots, n, \quad (8)$$

To the best of our knowledge, this is the first time that such criteria have been proposed. Generating a trajectory based on these criteria (8) enables us to increase robustness in order to model uncertainties and control error around the singularity. Experimental results show that nullifying the first two derivatives of the criterion might be sufficient.

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*) [17, 18] 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 Part 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  $\mathbf{A}$  is singular, it is not possible to compute the Cartesian velocities  $\mathbf{v}$  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.

Let us rewrite the dynamic model of the mechanism as [17, 20]:

$$\boldsymbol{\tau} = \mathbf{w}_b + \mathbf{J}^T \mathbf{w}_p = \mathbf{M}\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) \quad (9)$$

By replacing the angular acceleration  $\ddot{\mathbf{q}}$  in Equation (9) by an adapted control signal  $u$ , the dynamic of the system is linear with respect to the control variable. Therefore, a double integrator between the control signal and the joint variables appears. Thus, only a *PD* control law is used to impose a second-order dynamic on the error:

$$u = \ddot{\mathbf{q}} = \ddot{\mathbf{q}}_d + \mathbf{K}_d \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} \quad \Rightarrow \quad \ddot{\mathbf{e}} + \mathbf{K}_d \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} = 0 \quad (10)$$

where

- $\mathbf{q}_d$  (respectively  $\dot{\mathbf{q}}_d$  and  $\ddot{\mathbf{q}}_d$ ) is the desired joint position (respectively velocity and acceleration),

- $\mathbf{q}$  (respectively  $\dot{\mathbf{q}}$  and  $\ddot{\mathbf{q}}$ ) is the current measured joint position (respectively velocity and acceleration).
- $\mathbf{e}$  (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.

Consequently, *CTC* computes the input torques by substituting Eq. (10) into the dynamic model presented in Eq. (9):

$$\boldsymbol{\tau} = \mathbf{M}(\ddot{\mathbf{q}}_d + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}) + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) \quad (11)$$

Figure (1) represents a classic Computed Torque Control law applied to a parallel mechanism whose dynamic model can be expressed by Equation (9).

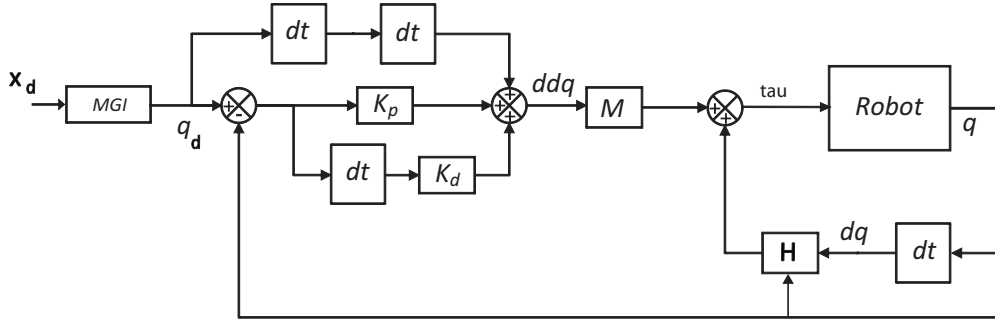


Figure 1: Classic Computed Torque Control law

It should be noted that the vector of positions in the task space  $\mathbf{x}$  is necessary to compute matrices  $\mathbf{M}$  and  $\mathbf{H}$ . However, in industrial cases sensors only measure the vector of positions in the joint space  $\mathbf{q}$ . 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 (cf. Part 2) will be different from the real one, due to errors in the dynamic model. As a result, numerically speaking, the **crossing criterion (7) 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  $\mathbf{A}$ .

The next section presents a multi-model control law which allows the controller to avoid this issue.

### 3.2. Multi-model control law

#### 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:

$$\mathbf{w}_p = 0 \text{ and } \frac{d^i}{dt^i} \mathbf{w}_p = 0, i = 1, \dots, n \quad (12)$$

This new criterion still guarantees that (8) and (10) are respected. 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. (2)):

- *Model 1* – The complete dynamic model, as long as matrix  $\mathbf{A}$  introduced in Sec. (2.1) is invertible:

$$\boldsymbol{\tau}_A = \mathbf{w}_b + \mathbf{J}^T \mathbf{w}_p = \mathbf{M}\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) \quad (13)$$

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

$$\tau_B = w_b = M' \ddot{q} + H'(q, \dot{q}) \quad (14)$$

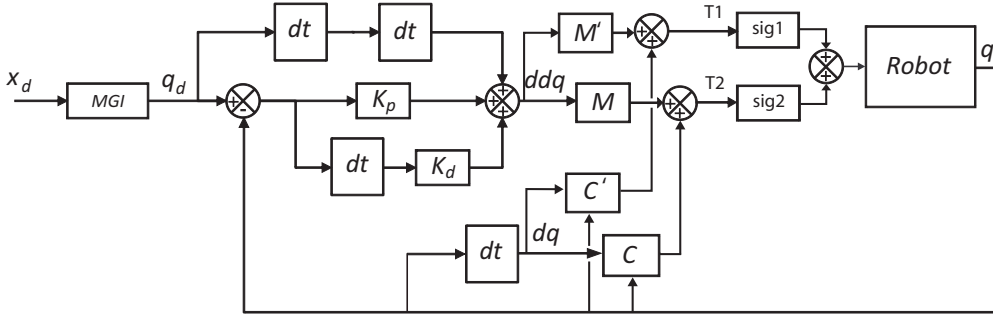


Figure 2: Multi-model computed torque control law

The second dynamic model is used to compute input torques only when the trajectory has been planned in order to have  $w_p = 0$ . Considering that the control law is correctly adjusted, the effective trajectory is close enough to the desired one and therefore the hypothesis  $w_p = 0$  is acceptable. Once the mechanism is far enough from the singularity locus, the control switches back to the complete dynamic model and the mechanism can finish its trajectory.

#### *Metric choice for model switching*

In order to cross the singularity locus, the controller requires a metric which defines the moment when *Model 2* has to be used. The discussion about the best indicator of singularity proximity is a well-known problematic [4, 21, 22]. However, since *Model 2* is valid only when  $w_p$  is null, here the metric used will be based on the value of  $w_p$ .

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

#### *Switching function: $\sigma$*

In order to guarantee torque continuity when the control switches from one model to another, the input torques are computed using the logistic function  $\sigma$  (shown in Fig. (3)) such that:

$$\tau = \sigma \tau_A + (1 - \sigma) \tau_B \quad (15)$$

This function  $\sigma$  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,
- $\frac{1}{1+e^{-\alpha t}}$  or  $1 - \frac{1}{1+e^{-\alpha t}}$  when switching between the two models, where  $t$  is the current time,

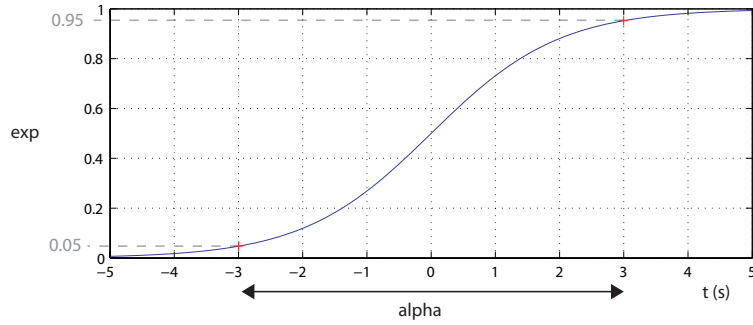


Figure 3: Variation of the logistic function and influence of parameter  $\alpha$

The logistic function  $\sigma$  varies continuously between 0 and 1, which prevents any torque discontinuity. Here, the logistic function is defined using a parameter  $\alpha$ , which can be adjusted to change the length of the transition phase. This parameter is computed based on the value of the derivative of  $w_p$  such that  $\sigma$  is null when  $w_p$  and its derivatives are null, i.e. when the mechanism is in a singular position, as presented in Fig (4).

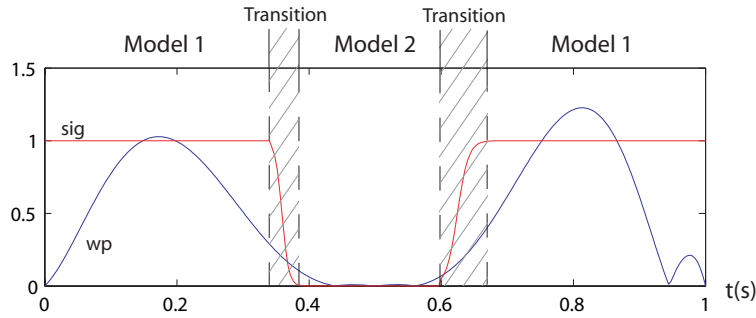


Figure 4: Variation in sigma based on the value of  $w_p$  when crossing a singularity

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

## 4. Case study

### 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  $A$  and  $E$  and 3 passive revolute joints at points  $B$ ,  $C$  and  $D$  (Fig. 5). 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.

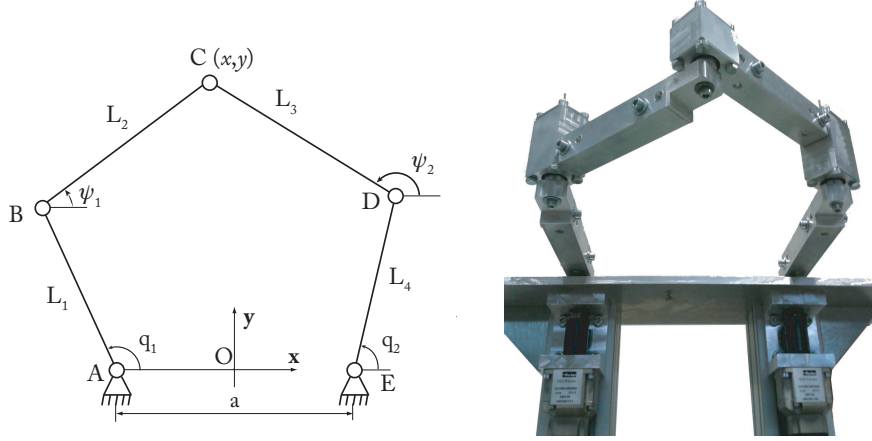


Figure 5: Five-bar mechanism designed and manufactured at IFMA

The mechanism and its parameters are presented in Fig. 5. The link dimensions were calibrated using a Laser Tracker (Table 1).

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)	$1.10^{-5}$	$1.10^{-5}$	$1.10^{-5}$	$1.10^{-5}$	$1.10^{-5}$

#### 4.2. Gain tuning

The proportional and derivative gains were tuned based on the mechanism's natural frequency [15]. 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  $\omega$  both gains are adjusted as:

$$K_p = \omega^2, \quad K_d = 2\xi\omega \quad (16)$$

where  $\xi$  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 [15]. Therefore a frequency of  $\omega = \omega_r/2$  was chosen, resulting in the following gain values:

$$K_p = 300, \quad K_d = 34 \quad (17)$$

#### 4.3. Dynamic modeling and identification

A full dynamic model of the robot was computed using the methodology presented in [24] 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 [23]. The identification resulted in the following model that fully describes the robot dynamics of the studied mechanism:

$$\tau = m_3 \mathbf{J}^T \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} + \begin{pmatrix} z z_1 \dot{q}_1 \\ z z_2 \dot{q}_2 \end{pmatrix} + \begin{pmatrix} f_{v1} \dot{q}_1 \\ f_{v2} \dot{q}_2 \end{pmatrix} + \begin{pmatrix} f_{s1} \text{sign}(\dot{q}_1) \\ f_{s2} \text{sign}(\dot{q}_2) \end{pmatrix} \quad (18)$$

where:

- $m_3$  is a mass equivalent located on the end effector;  $m_3 = 0.40 \pm 0.02\text{kg}$
- $zz_1$  and  $zz_2$  are rotational equivalent inertial terms, respectively on the first and second actuator;  $zz_1 = 1.83 \cdot 10^{-2} \pm 6.97 \cdot 10^{-4}\text{kg.m}^2$ ;  $zz_2 = 1.96 \cdot 10^{-2} \pm 6.60 \cdot 10^{-4}\text{kg.m}^2$ ;
- $f_{s1}$  is a Coulomb friction term on the first actuator (respectively  $f_{s2}$  on the second actuator);  $f_{s1} = 2.94 \pm 0.10\text{N.m}$ ;  $f_{s2} = 2.95 \pm 0.09\text{N.m}$ ;
- $f_{v1}$  is a viscous friction term on the first actuator (respectively  $f_{v2}$  on the second actuator);  $f_{v1} = 6.76 \pm 0.018\text{N.m.s}$   $f_{v2} = 6.75 \pm 0.17\text{N.m.s}$ .

This identified dynamic model is related to Eq. (2) by:

$$\begin{aligned} \mathbf{w}_p &= m_3 \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix}, \\ \mathbf{w}_b &= \begin{pmatrix} zz_1 \ddot{q}_1 \\ zz_2 \ddot{q}_2 \end{pmatrix} + \begin{pmatrix} f_{v1} \dot{q}_1 \\ f_{v2} \dot{q}_2 \end{pmatrix} + \begin{pmatrix} f_{s1} \text{sign}(\dot{q}_1) \\ f_{s2} \text{sign}(\dot{q}_2) \end{pmatrix} \end{aligned} \quad (19)$$

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. (6) which represents the input torques measured and computed along a trajectory.

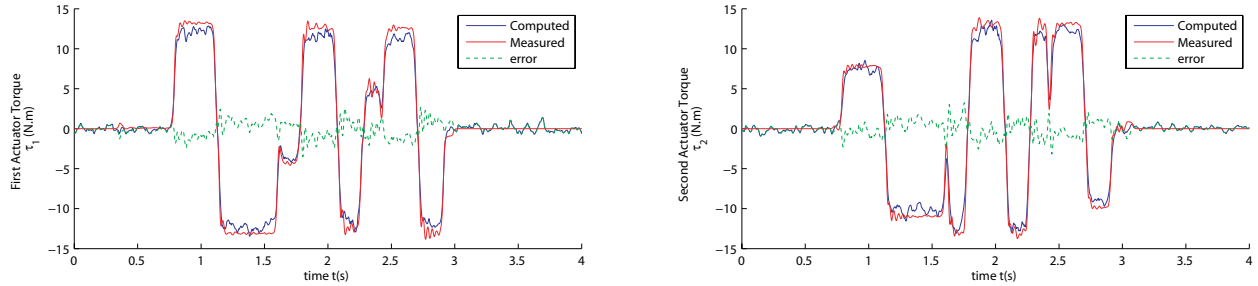


Figure 6: Verification of identified dynamic model

#### 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. (9), the task space acceleration has to be expressed as a function of the joint space acceleration. This can be done by differentiating the kinematic model:

$$\mathbf{v} = \mathbf{J}\dot{\mathbf{q}} \Rightarrow \dot{\mathbf{v}} = \dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{J}\ddot{\mathbf{q}} \quad (20)$$

By substituting (20) into (18) one can obtain the dynamic model used for the computed torque control law as presented in Equation (9):

$$\boldsymbol{\tau} = \mathbf{M} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix} + \mathbf{H}, \text{ where} \quad (21)$$

$$\mathbf{M} = m_3 \mathbf{J}^T \mathbf{J} + \begin{pmatrix} zz_1 & 0 \\ 0 & zz_4 \end{pmatrix} \quad (22)$$

$$\mathbf{H} = m_3 \mathbf{J}^T \dot{\mathbf{J}} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix} + \begin{pmatrix} f_{v1}\dot{q}_1 + f_{s1}\text{sign}(\dot{q}_1) \\ f_{v2}\dot{q}_2 + f_{s2}\text{sign}(\dot{q}_2) \end{pmatrix} \quad (23)$$

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 1Khz (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:

$$x = P_x(x_f - x_0) + x_0, \quad y = P_y(y_f - y_0) + y_0 \quad (24)$$

where

- $P_x(t_0) = P_y(t_0) = 0$ ,
- $P_x(t_f) = P_y(t_f) = 1$ ,

They are both 8<sup>th</sup> 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 (12) is respected around the singularity locus [13].

Figure 7 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 (12) for:

$$\begin{aligned} x_0 &= 0.1, & y_0 &= 0.34, \\ x_s &= 0.05475, & y_s &= 0.2, \\ x_f &= 0, & y_f &= 0.1, \end{aligned} \quad (25)$$

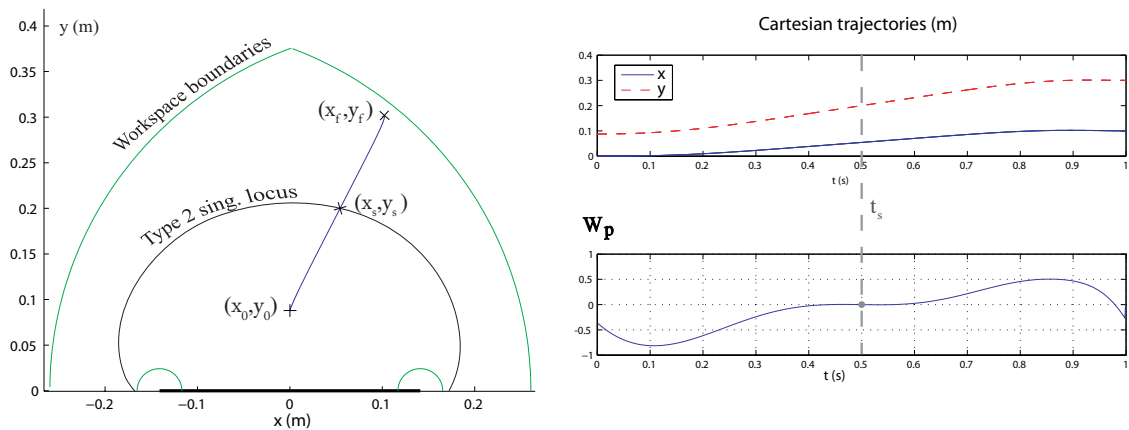


Figure 7: Example of 8<sup>th</sup> order polynomial trajectory crossing a Type 2 singularity locus

#### 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 Sec. (4.5.1). However, numerically the dynamic model degenerates (e.g. Sec. (3.1)) and the control law computes infinite torques. Fig. (8) presents the experimental results when following a crossing trajectory computed as presented in Sec. (4.5.1) with a classic CTC.

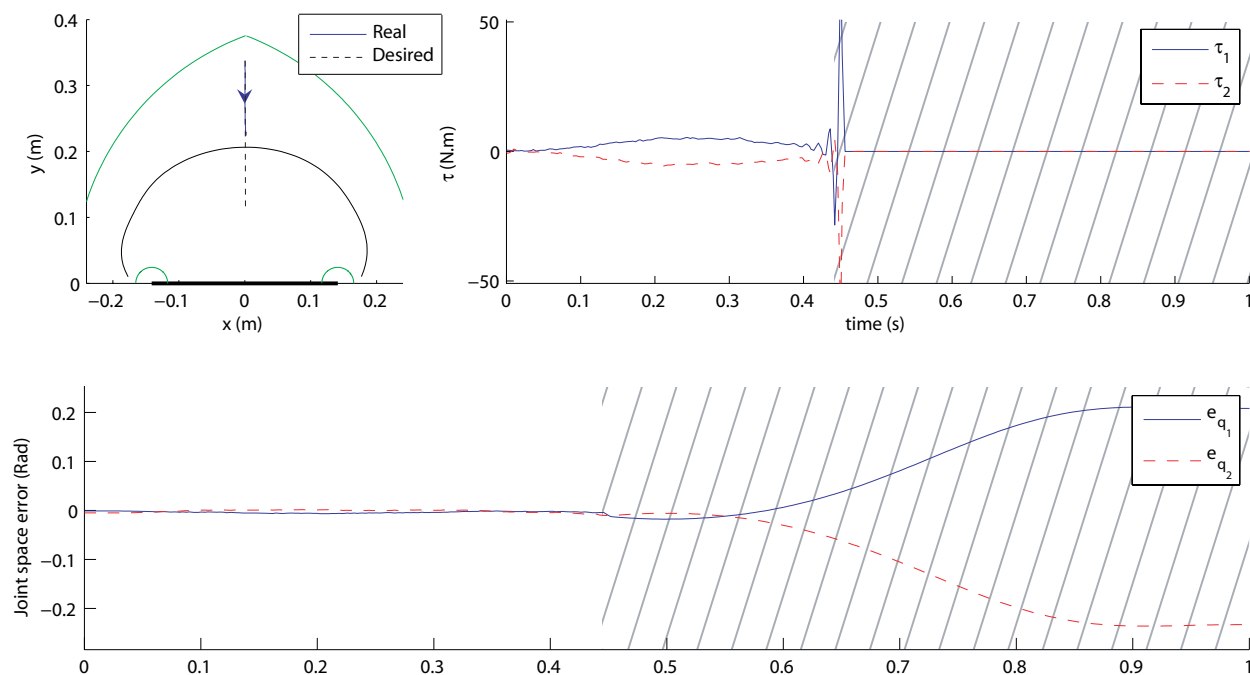


Figure 8: Measured input torques, tracking error and trajectory when following a crossing trajectory with a classic computed torque control law

When the mechanism approaches the singularity locus, the input torques  $\tau$  become discontinuous and tend to infinity. Therefore, even if the mechanism should reach the singularity locus after 0.5s, it can be seen in Fig. (8) that the input torques tend to infinity before reaching the singularity. The result of the robot displacement is shown in Fig. (9). 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.

Figure 10 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. Figure (11) 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 can significantly increase in order to nullify this error. This can be seen for the first trajectory at 0.7s.

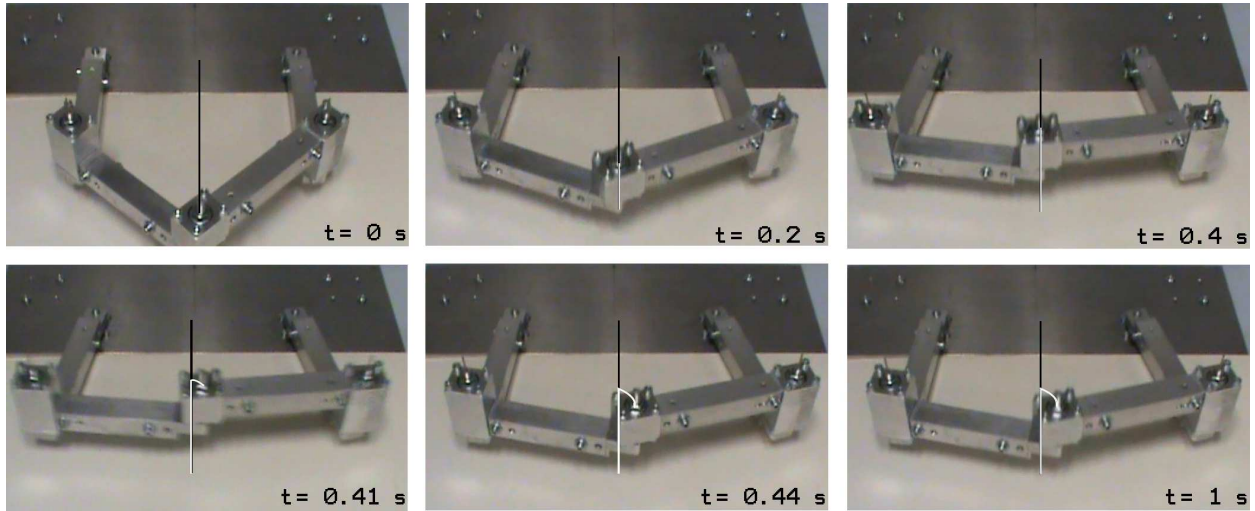


Figure 9: Images of the mechanism trying to cross a singularity with a classic CTC control law

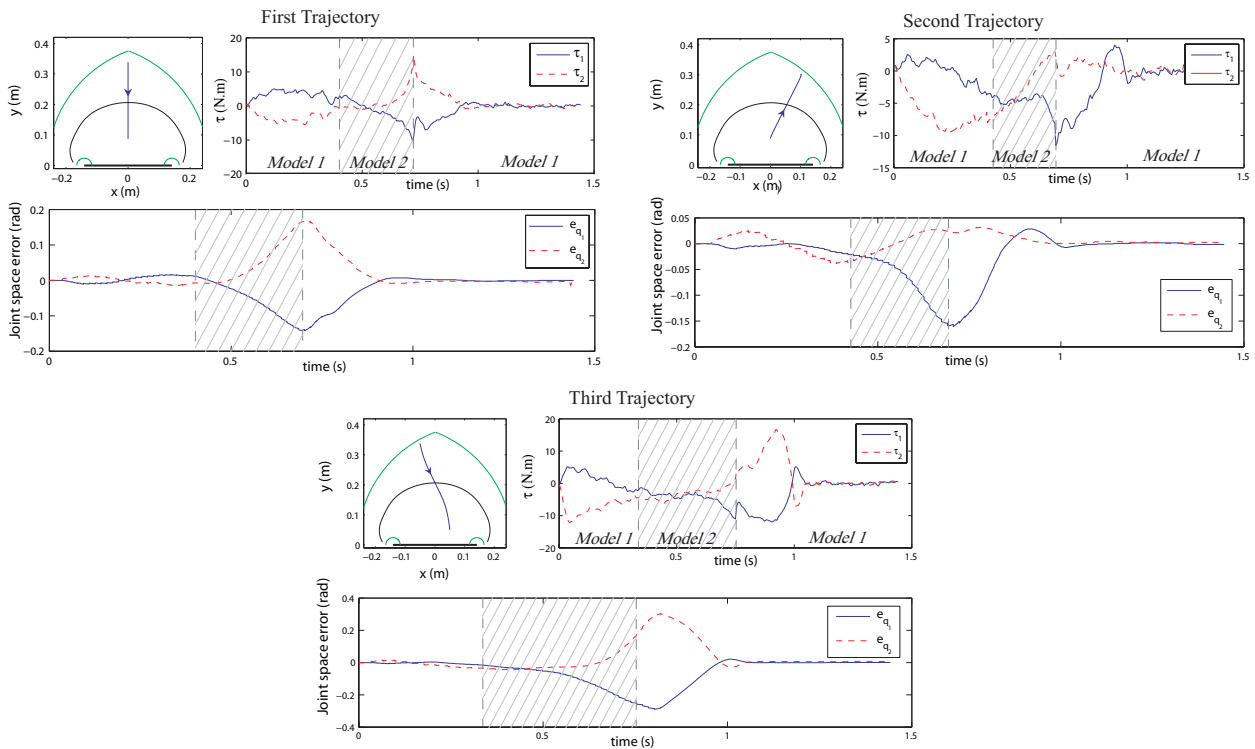


Figure 10: Different crossing trajectories and corresponding input torque applied

All three trajectories represented in Figure 10 were planned to cross the singularity at 0.5s and end at 1s. 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.

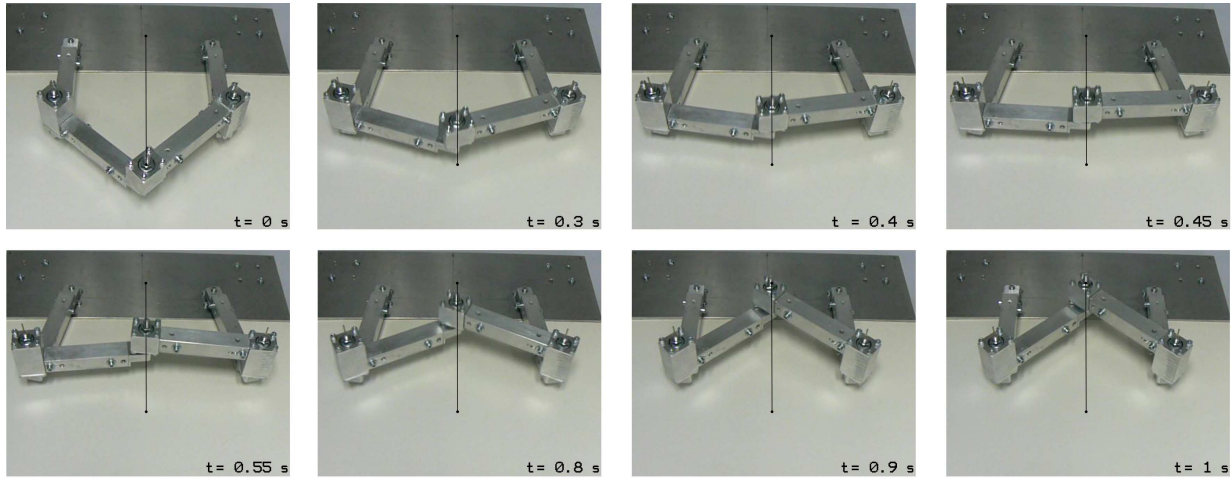


Figure 11: Trajectory reproduction during a singularity crossing

Figure (12) represents the Cartesian coordinates of the end-effector 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.

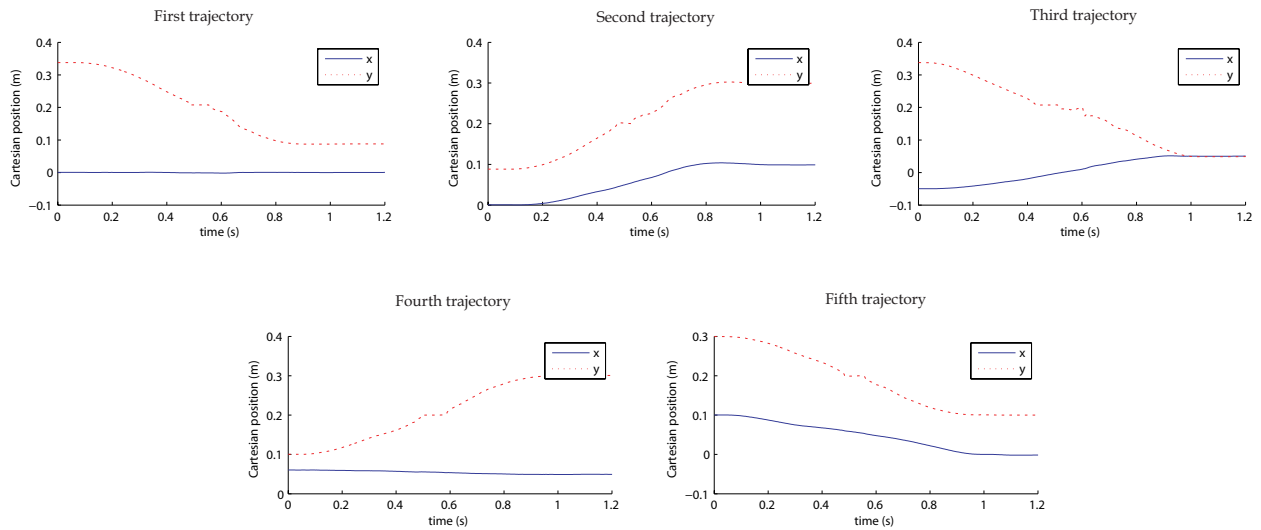


Figure 12: Cartesian coordinates along 5 different crossing trajectories

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 end-effector reachable space of almost any parallel mechanism.

## 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 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 standard dynamic control law 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.

## Acknowledgements

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

- [1] 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*.
- [2] Conconi, M., & Carricato, M., 2009. "A new assessment of singularities of parallel kinematic chains". *IEEE Transactions on Robotics*, 25(4), pp. 757-770.
- [3] Gosselin, C., & Angeles, J. (1990). "Singularity analysis of closed-loop kinematic chains". *Robotics and Automation, IEEE Transactions on*, 6(3), 281-290.
- [4] Merlet, J. P. (1993). "Closed-form resolution of the direct kinematics of parallel manipulators using extra sensors data". *Robotics and Automation, 1993. Proceedings., 1993 IEEE*, 54(2), 200-204.
- [5] Briot, S., Pashkevich, A., & Chablat, D., 2010. "Optimal technology-oriented design of parallel robots for high-speed machining applications". In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA 2010)*.
- [6] Liu, X.-J., Wang, J., and Pritschow, G., 2006. "Performance atlases and optimum design of planar 5R symmetrical parallel mechanisms". *Mechanism and Machine Theory*, 41(2), pp. 119-144.
- [7] Kong, X., & Gosselin, C., 2002. "A class of 3-dof translational parallel manipulators with linear input-output equations". In *Proceedings of the Workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators*, pp. 3-4.
- [8] Gogu, G., 2004. "Structural synthesis of fully-isotropic translational parallel robots via theory of linear transformations". *European Journal of Mechanics. A/Solids*.
- [9] Kurtz, R., & Hayward, V., "Multiple-goal kinematic optimization of a parallel spherical mechanism with actuator redundancy". *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 644-651, 1992.
- [10] Nahon, M. A., & Angeles, J., "Force optimization in redundantly actuated closed kinematic chains". In *Proceedings of the International Conference on Robotics and Automation*, 1989.
- [11] Rakotomanga, N., Chablat, D., Caro, S., 2008. "Kinetostatic performance of a planar parallel mechanism with variable actuation". *Advances in Robot Kinematics*.
- [12] Zein, M., Wenger, P., & Chablat, D., 2008. "Non singular assembly-mode changing motions for 3-RPR parallel manipulators". *Mechanism and Machine Theory*, 43(4), pp. 480-490.
- [13] Briot, S., Arakelian, V., & Chablat, D. (2008). "Optimal Force Generation in Parallel Manipulators for Passing through the Singular Positions". *The International Journal of Robotics Research*, 27(8), 967-983.
- [14] Ider, S. K., 2005. "Inverse dynamics of parallel manipulators in the presence of drive singularities". *Mechanism and Machine Theory*, 40, pp. 33-44.
- [15] Khalil, W., & Dombre, E. (2004). "Modeling, identification & control of robots".
- [16] Craig, J. J., & Hall, P. P. (n.d.). "Introduction to Robotics".
- [17] Spong, M., & Vidyasagar, M. (2004). "Robot dynamics and control".

- [18] Paccot, F., Andreff, N., Martinet, P. (2007) "A review on dynamic control of parallel kinematic machine: theory and experiments". *International Journal of Robotics Research*, Volume 28 Issue 3, March 2009, Pages 395-416.
- [19] Zlatanov, D., & Bonev, I. (2002). "Constraint singularities of parallel mechanisms". *Robotics and Automation*.
- [20] Ghorbel, F., Chetelat, O., & Longchamp, R. (1994). "A reduced model for constrained rigid bodies with application to parallel robots". *IFAC symposium on robot Control SYROCO'94*, (3), 57-62.
- [21] Glazunov, V., Arakelian, V., Briot, S., & Rashoyan, G. V. (2011). "Speed and Force Criteria for the Proximity to Singularities of Parallel Structure Manipulators". *Machinery Mechanics*.
- [22] Voglewede, P. A., & Ebert-Uphoff, I. (2004). "Measuring closeness to singularities for parallel manipulators". *IEEE Int. Conf. on Robotics and Automation*, 4539-4544.
- [23] Gautier, M. (1997). "Dynamic identification of robots with power model". *Robotics and Automation* 1997, 3, 1922-1927.
- [24] Briot, S., & Gautier, M. (2012). "Global Identification of Drive Gains and Dynamic Parameters of Parallel Robots". *Proceeding of the 19th CISM-Iftomm Symposium*, 93-100.