GDR Robotique - Journée GT6

An algorithm for assembly mode detection of parallel robots

Adrien Koessler

Institut Pascal, Sigma Clermont, IRCCyN

October 3rd, 2016

1. Methods for Assembly or Operation Mode change
2. Assembly Mode detection of parallel manipulators
3. New Algorithm for Assembly Mode detection
4. Experiments on DexTAR robot
5. Work perspectives on the presented algorithm

Introduction

Pros and cons of parallel manipulators (PMs) in an industrial context:

- Short cycle time
- High rigidity and payload capacity
- Complicated mechanical design
- Reduced operationnal workspace
- Bound to a single task

⇒ Enlargement of PM operationnal workspace
⇒ Multitasking through PM reconfiguration

Contents

Parallel manipulator workspace enlargement

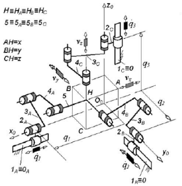
Design-based solutions:

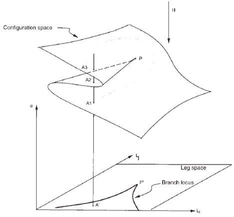
- Topological or geometrical optimization [Gog04]
- Redundant or variable actuation [Mül08] [AG08]

Trajectory-based solutions (only Type 2 singularity):

- Cusp point bypassing [MD99]
- Nondegenerating singularity crossing trajectory [BA08] [PBBM15]

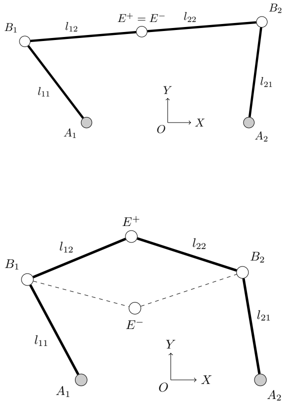
⇒ Focus on the last approach (can be generalized)



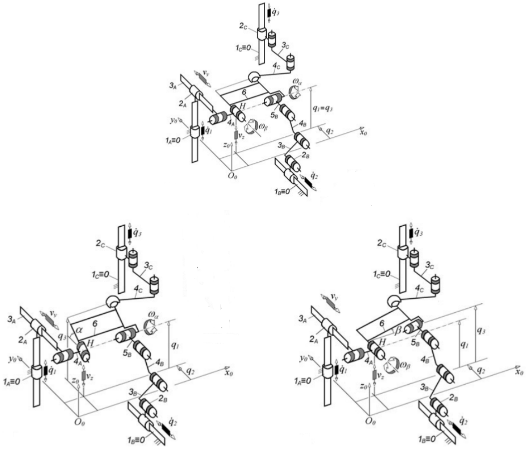


Type 2 singularities and constraint singularities

Type 2 singularity and Assembly Modes.



Constraint singularity and bifurcation in Operation Modes [Gog11].



(a) (b)

Singularity crossing

State of the art in singularity crossing:

	Type 2 sing.	Constraint sing.
Dynamic modelling of singular configurations	✓	×
Advanced control for AM/OM change	✓	×
Current AM/OM detection	in progress	×
Optimal trajectory generation to change AM/OM	×	×

Needs:

- Certify operation mode change
- Generate trajectories to ensure reconfiguration

Controlled crossing of Type 2 singularities

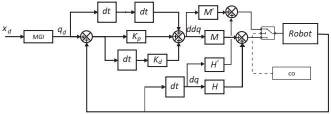
Criterion for non-degenerescence of dynamic model in Type 2 singular configuration:

$$\mathbf{t}_s^T \mathbf{w}_p = 0$$

$\mathbf{t}_s$: uncontrolled motion direction
$\mathbf{w}_p$: robot end-effector acceleration.

Crossing method [PBBM15]:

- Create a trajectory that respects the criterion near the singularity
- Suppose it is respected in the dynamic control scheme



$\Rightarrow$ Enables crossing, but inertia is needed

Contents

Usual Assembly Mode detection methods

Goal: pick the right solution to the Forward Kinematic problem, corresponding to current AM.

Sensor-based methods:

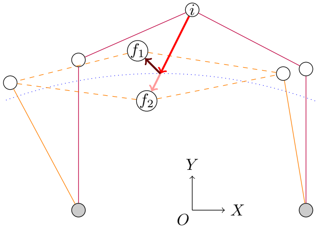
- Proprioceptive sensors:
  - Sensing of passive articulation joint value [Tan95].
  - Direct measurement using passive legs.
- Exteroceptive sensors: vision of PM's platform.

$\Rightarrow$ Additionnal cost and implementation difficulties

Iterative methods: solving the FK problem using Newton methods and an initial pose guess.
$\Rightarrow$ Diverging near singular configurations, cannot handle AM change

Iterative solver with velocity information

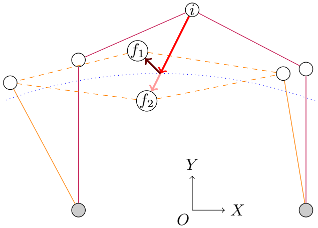
In the general case, the Forward Geometric Model gives rise to multiple solutions:



Which final configuration f is the right one...

Iterative solver with velocity information

In the general case, the Forward Geometric Model gives rise to multiple solutions:

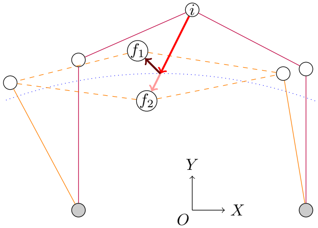


Which final configuration f is the right one...

- ... without additionnal information ?

Iterative solver with velocity information

In the general case, the Forward Geometric Model gives rise to multiple solutions:

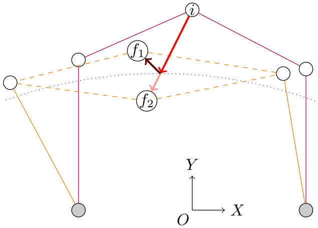


Which final configuration f is the right one...

- ... without additionnal information ?
- ... knowing that $\dot{y} < 0$?

Iterative solver with velocity information

In the general case, the Forward Geometric Model gives rise to multiple solutions:



Which final configuration f is the right one...

- ... without additionnal information ?
- ... knowing that $\dot{y} < 0$?

$\Rightarrow$ Idea: iterative end-effector pose **and velocity** tracking.

Contents

Use of Interval Analysis

Merlet [Mer04] showed the interest of using Interval Analysis (IA) in iterative FK solvers

- Numerically robust
- Takes extra constraints into account
- Computationally efficient

In IA, any variable a is represented by an interval $[a] = [\underline{a}, \overline{a}] = \{a \in \mathbb{R}; \underline{a} \leq a \leq \overline{a}\}$.
Intervals are extended to vectors through cartesian product $[\mathbf{a}] = [\underline{a}_1, \overline{a}_1] \times \cdots \times [\underline{a}_n, \overline{a}_n]$.
Interval vectors are called *boxes*.

Solving systems of equations through IA is possible.

Kinematic problem solving

Enclosures on end-effector pose $[\mathbf{x}]$ and velocity $[\dot{\mathbf{x}}]$ are sought.
Using:

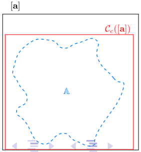
- $[\mathbf{q}]$, $[\dot{\mathbf{q}}]$ actuated joint values and rates,
- $[\boldsymbol{\xi}]$ geometric parameters,
- $[\mathbf{x}_i]$, $[\dot{\mathbf{x}}_i]$ initial guesses,

the following system is solved in $\mathbf{x}$, $\dot{\mathbf{x}}$.

$$\mathbf{f}(\mathbf{x}, \mathbf{q}, \boldsymbol{\xi}) = \mathbf{0}. \tag{1}$$

$$\mathbf{J}_x(\mathbf{x}, \mathbf{q}, \boldsymbol{\xi})\,\dot{\mathbf{x}} + \mathbf{J}_q(\mathbf{x}, \mathbf{q}, \boldsymbol{\xi})\,\dot{\mathbf{q}} = \mathbf{0}. \tag{2}$$

Solving this system relies on *contraction*.
When the system (1), (2) is well-conditionned, $\mathrm{width}([\mathbf{x}]) < \mathrm{width}([\mathbf{x}_i])$.
When the system is ill-conditionned, $[\mathbf{x}] \approx [\mathbf{x}_i]$.



Robot free evolution model

How are initial bracketings found to solve the FK problem ?

Merlet: "*any additional constraints that may limit the number of realistic solutions to the FK problem may be taken into account.*" [Mer04]
We suppose that enclosures $[\mathbf{v}_{max}]$, $[\mathbf{a}_{max}]$ on all possible end-effector velocities and accelerations exist.

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \int_0^{t_s} \mathbf{v}(t)\,dt \quad (3)$$

$$\Downarrow$$

$$[\mathbf{x}]_{k-1} + t_s[\mathbf{v}_{max}] =: [\mathbf{x}]_k. \quad (4)$$

$$\dot{\mathbf{x}}_k = \dot{\mathbf{x}}_{k-1} + \int_0^{t_s} \mathbf{a}(t)\,dt \quad (5)$$

$$\Downarrow$$

$$[\dot{\mathbf{x}}]_{k-1} + t_s[\mathbf{a}_{max}] =: [\dot{\mathbf{x}}]_k. \quad (6)$$

$\Rightarrow$ This procedure always expands the intervals

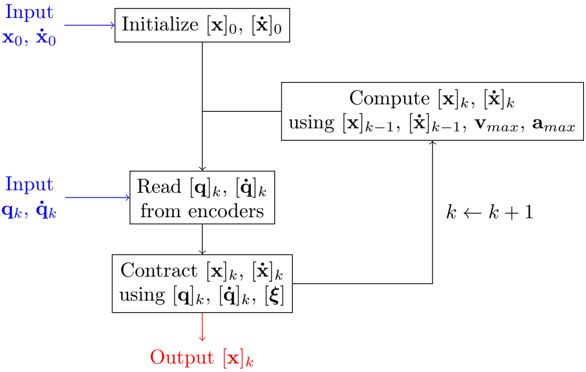
Proposed algorithm

Procedures:

- Initialization
- Expansion
- Joint reading
- Contraction

Parameters:

- $\mathbf{w}_0$
- $\mathbf{v}_{max}, \mathbf{a}_{max}$
- ϵ_q



Algorithm behaviour when crossing a singularity

How the algorithm is expected to behave:

- Far from singularities contraction is efficient, $[\mathbf{x}]$ and $[\dot{\mathbf{x}}]$ are tight.
- During the crossing contraction is inefficient, $[\mathbf{x}]$ and $[\dot{\mathbf{x}}]$ get wider.

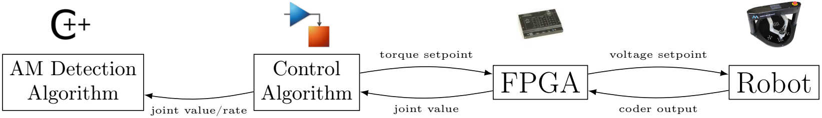
Two cases are then possible:

1. Either the bracketings remain little enough to contain a single solution to the FK problem,
2. or bracketings diverge and cannot help choosing the current assembly mode.

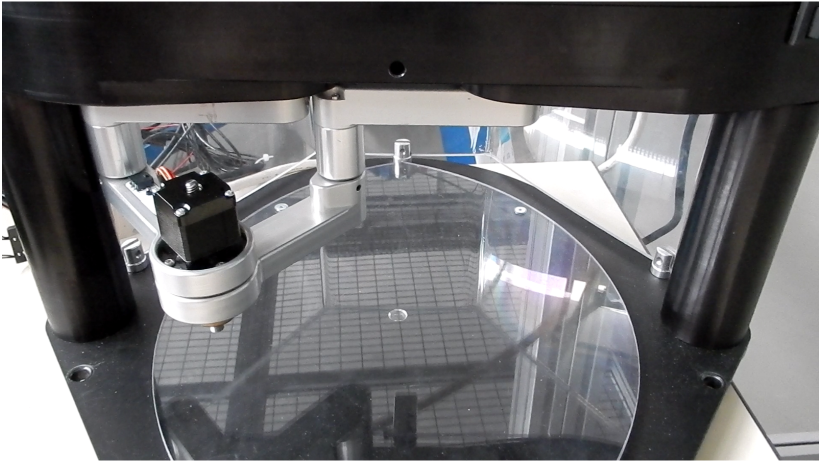
Contents

Testing robot and software

- Off-line algorithm
- Tested on DexTAR, 2-dof planar parallel robot with two Assembly Modes
- Kinematic calibration is needed
- IBEX library for interval computations



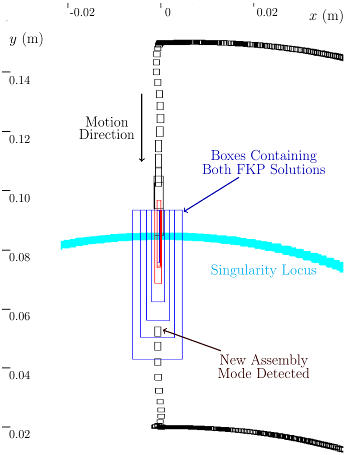
Testing trajectory



Algorithm output

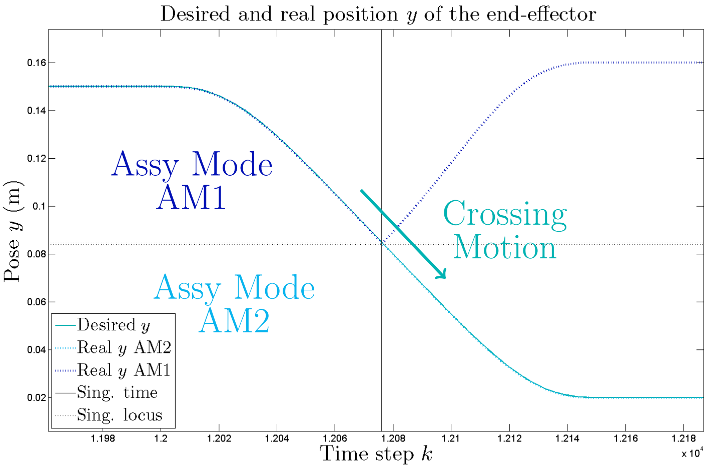
After the crossing is done, the boxes converge.

We focus on the behaviour along y-axis to explain how AM detection works.



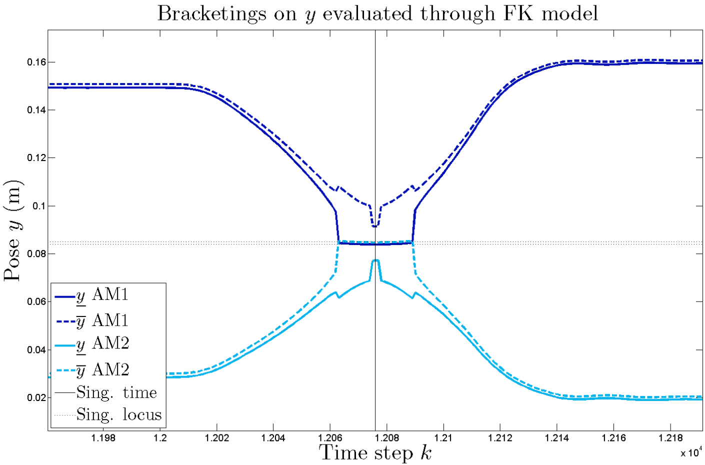
Analysis: possible y values

Recorded joint values are fed to FK model: *cross* or *bounce*.



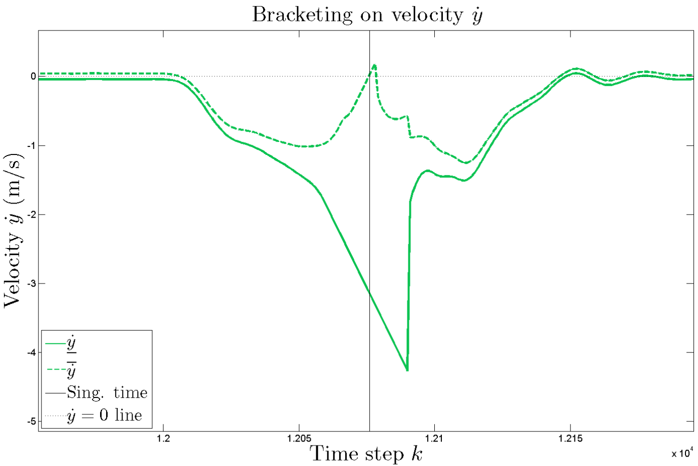
Analysis: FK model interval evaluation

Robot is in AM1 or AM2 $\Rightarrow y_k \in {}^1[y]_k \cup {}^2[y]_k$.



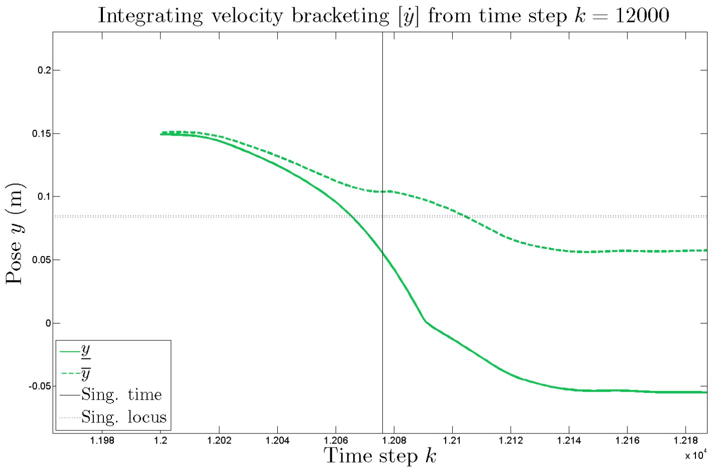
Analysis: velocity bracketing

The velocity of the end-effector is tracked too. Integrating it gives insight on end-effector pose.



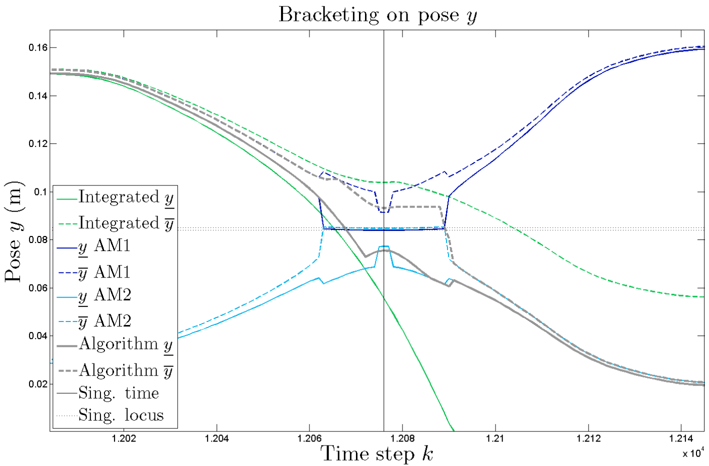
Analysis: resulting pose bracketing

Robot starts in AM1, so $y_k \in {}^{vel}[y]_k = {}^{1}[y]_i + \int_i^k [\dot{y}]_k \, dk$.



Analysis: bracketings combined and AM detected

Let us combine both bracketings: $y_k \in {}^{vel}[y]_k \cap ({}^{1}[y]_k \cup {}^{2}[y]_k)$.
Convergence of algorithm output is obtained, thus AM is detected.



Conclusion on experiments

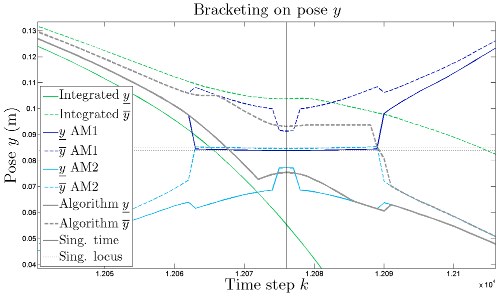
- Algorithm reliability was proven.
- How to obtain $\mathbf{v}_{max}$ and $\mathbf{a}_{max}$ is an open question.
- The algorithm is very pessimistic: it is unlikely that $\ddot{\mathbf{x}} = \mathbf{a}_{max}$ on singularity, since crossing criterion imposes $\ddot{x} = 0$ along the uncontrolled direction.
- Algorithm only needs:
  1. a kinematic model of a mechanism
  2. reasonable velocity and acceleration bounds

  It is not only fot Type 2 singularities !

Contents

Motion generation to ensure singularity crossing

Experimental fact: inertia is needed to succeed in crossing singularities.
Algorithm output analysis: higher velocity setpoint allows to eliminate parasite solutions to FK problem.



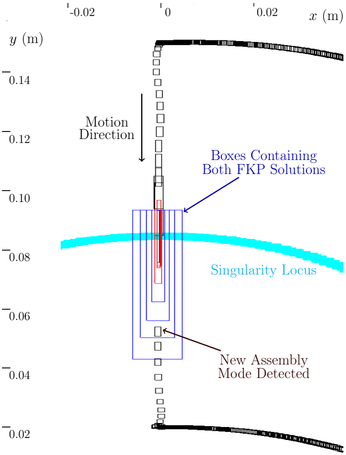
$\Rightarrow$ reverse the inputs and outputs of the algorithm to create a **motion generator with ensured crossing**.

Singularity proximity indication

Singularity locus should be seen in the $(\mathbf{x}, \mathbf{q})$-space.

The red boxes denote that the robot might be in a singular configuration.

This is an **indicator for proximity of singularity**, based on geometrical uncertainities.



Computation of operationnal velocity and acceleration limits

The goal is to get reliable values of $\mathbf{v}_{max}$ and $\mathbf{a}_{max}$.

1. Through **modelling**: a way to solve kinematic equations in singular configurations should be investigated
   $\Rightarrow$ full-cycle $\mathbf{v}_{max}$ and $\mathbf{a}_{max}$ values.
2. Through **observation**: an estimator for acceleration should be synthetized, from which reliable bounds can be deduced
   $\Rightarrow$ local $\mathbf{v}_{max}$ and $\mathbf{a}_{max}$ values.

Thanks

This PhD work is funded by Sigma Clermont.
Supervisors are Pr. Y. Mezouar, N. Bouton, B.C. Bouzgarrou and S. Briot.
We are grateful to A. Goldsztejn for his help on Interval Analysis.

Thank you for your attention.
Do you have questions ?

Robot Evolution Model

The equations of evolution of the real positions and speeds for time step k are

$$\mathbf{x}(kT_s) = \mathbf{x}((k-1)T_s) + \int_{(k-1)T_s}^{kT_s} \dot{\mathbf{x}}(\tau)d\tau$$

$$\dot{\mathbf{x}}(kT_s) = \dot{\mathbf{x}}((k-1)T_s) + \int_{(k-1)T_s}^{kT_s} \ddot{\mathbf{x}}(\tau)d\tau$$

with T_s the sampling time. Using intervals, we can write:

$$[\mathbf{x}_k] = [\mathbf{x}_{k-1}] + [\int_{(k-1)T_s}^{kT_s} \dot{\mathbf{x}}(\tau)d\tau]$$

$$[\dot{\mathbf{x}}_k] = [\dot{\mathbf{x}}_{k-1}] + [\int_{(k-1)T_s}^{kT_s} \ddot{\mathbf{x}}(\tau)d\tau]$$

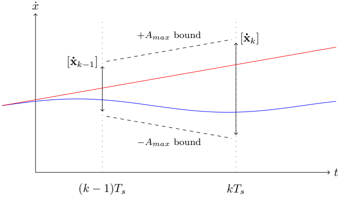
How can the integrals be bracketed ?

Robot Evolution Model

It can be seen that $T_s[\dot{\mathbf{x}}_k]$ is a valid enclosure of $\int_{(k-1)T_s}^{kT_s} \dot{\mathbf{x}}(\tau)d\tau$.

For the second integral, we have no choice for now but using **mechanical limits** for end-effector velocity V_{max} and acceleration A_{max}.

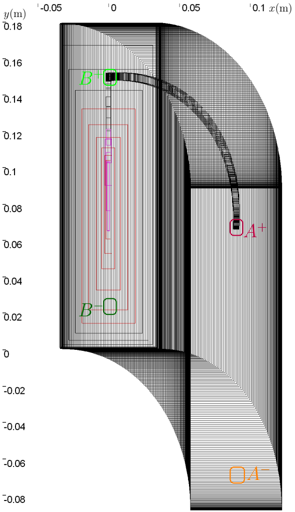
This gives us the following:



$$[\dot{\mathbf{x}}_k] = ([\dot{\mathbf{x}}_{k-1}] + T_s A_{max}[-1, 1]^n) \cap V_{max}[-1, 1]^n$$

$$[\mathbf{x}_k] = [\mathbf{x}_{k-1}] + T_s[\dot{\mathbf{x}}_k]$$

End-Effector Position Tracking



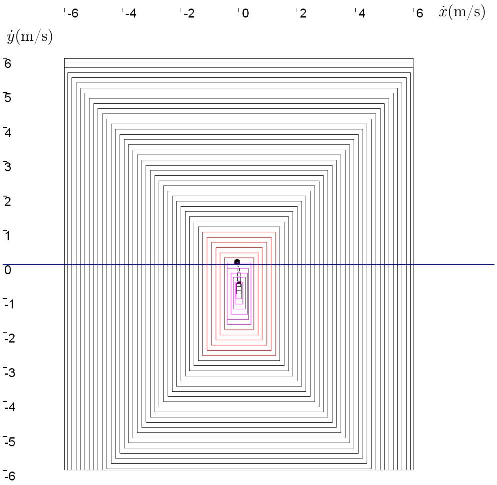
The crossing phase between B^+ to B^- is not followed well by the intervals.

Intervals grow up to the **geometrical limits**. They contain **both results** to FGM.

This result is not a bug, it's **meaningful** !

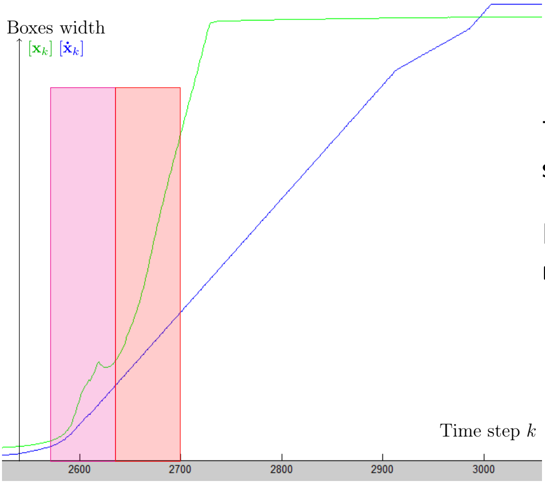
- Parameters are too imprecise or plain wrong.
- Trajectory is slow or waving around the singularity.

End-Effector Velocity Tracking



Colored boxes are not kept under the **blue line** **that marks** $\dot{y} < 0$.

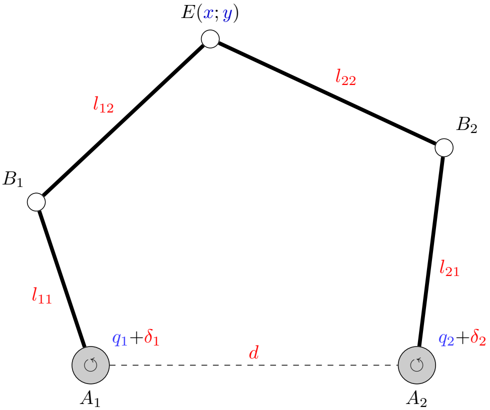
Position-Velocity Coupling



This example does not show convergence.

Boxes grow to their **maximal width**.

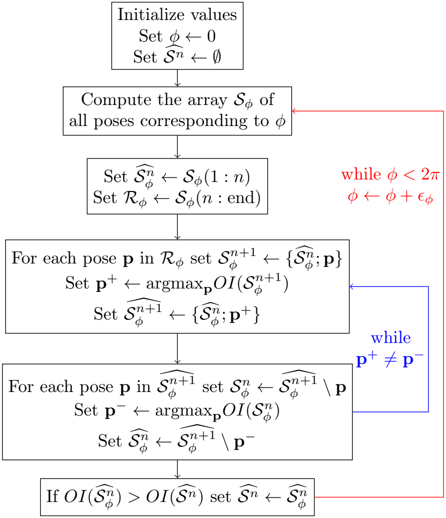
Parameters to Identify



There are seven parameters to identify : five lengths and two angles.

Joint values come from encoder measurements. End-effector pose comes from external device.

Choice of Calibration Poses

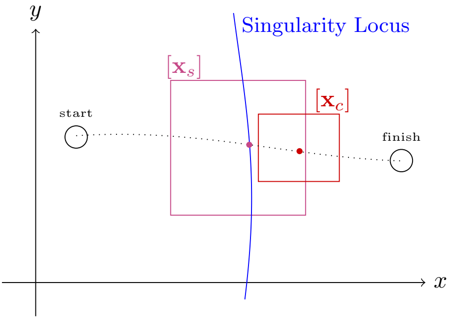


The algorithm used to maximize the observability index is **DETMAX** [JB13].

The orientation ϕ of the end-effector should be taken into account as the target needs to be always visible for the measurement appliance.

Trajectory Preprocessing

Another goal of this work is to ensure the crossing.
We saw that the speed along the trajectory was relevant to the success of the algorithm.



We have an enclosure of $\mathbf{x}$ at the singularity.
We want to ensure the crossing some time after.

Can we find **conditions on** $\dot{\mathbf{x}}$?
Are those **optimal** ?

Feasible End-effector Velocities and Accelerations

Equations for velocity and acceleration are:

$$\mathbf{A}\dot{\mathbf{x}} + \mathbf{B}\dot{\mathbf{q}} = 0 \tag{7}$$

$$\mathbf{A}_e\ddot{\mathbf{x}} + \mathbf{B}_e\ddot{\mathbf{q}} + \mathbf{b}_e = 0 \tag{8}$$

$$\boldsymbol{\tau} = \mathbf{M}\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) \tag{9}$$

Starting from $[\boldsymbol{\tau}]$ and $[\dot{\mathbf{q}}]$ we can go all the way back to $[\dot{\mathbf{x}}]$ and $[\ddot{\mathbf{x}}]$

- to find full-cycle values of V_{max} and A_{max}
- to give better enclosures on $[\dot{\mathbf{x}}]$ and $[\ddot{\mathbf{x}}]$ in real-time

(9) and (8) can be made **consistent in the singularities**. What about (7) ?

References I

- Vigen Arakelian and Victor Glazunov, *Mechanism and Machine Theory Increase of singularity-free zones in the workspace of parallel manipulators using mechanisms of variable structure*, Mechanism and Machine Theory **43** (2008), 1129–1140.
- Sébastien Briot and Vigen Arakelian, *Optimal Force Generation in Parallel Manipulators for Passing through the Singular Positions*, The International Journal of Robotics Research **27** (2008), no. 8, 967–983.
- Grigore Gogu, *Structural synthesis of fully-isotropic translational parallel robots via theory of linear transformations*, European Journal of Mechanics A/Solids **23** (2004), 1021–1039.

References II

- ______, *Bifurcation in constraint singularities and structural parameters of parallel mechanisms*, Meccanica **46** (2011), no. 1, 65–74.
- Ahmed Joubair and Ilian A. Bonev, *Comparison of the efficiency of five observability indices for robot calibration*, Mechanism and Machine Theory **70** (2013), 254–265.
- P R Mcaree and R W Daniel, *An Explanation of Never-Special Assembly Changing Motions for 3-3 Parallel Manipulators*, International Journal of Robotic Research **18** (1999), no. 6, 556–574.
- J. P. Merlet, *Solving the Forward Kinematics of a Gough-Type Parallel Manipulator with Interval Analysis*, The International Journal of Robotics Research **23** (2004), no. 3, 221–235.

References III

Andreas Müller, *Redundant Actuation of Parallel Manipulators*, Parallel Manipulators, Towards New Applications, 2008, pp. 87–108.

Georges Pagis, Nicolas Bouton, Sebastien Briot, and Philippe Martinet, *Enlarging parallel robot workspace through Type-2 singularity crossing*, Control Engineering Practice **39** (2015), 1–11.

Luc Tancredi, *De la simplification et la résolution du modèle géométrique direct des robots parallèles*, Ph.D. thesis, 1995, pp. 22–32.