Guaranteed Nonlinear Model Predictive Control via Validated Simulation

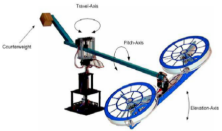
Mohamed Fnadi[1], Julien Alexandre Dit Sandretto[2], Laurent Fribourg[3]

[1]ULCO, LISIC
[2]ENSTA Paris, U2IS
[3]ENS Paris Saclay, CNRS, LSV

International Online Seminar on Interval Methods in Control Engineering

January 2022

Context



Scientific issues

- Need strong guarantees (critical systems)
- Unsafe and less reliable controllers

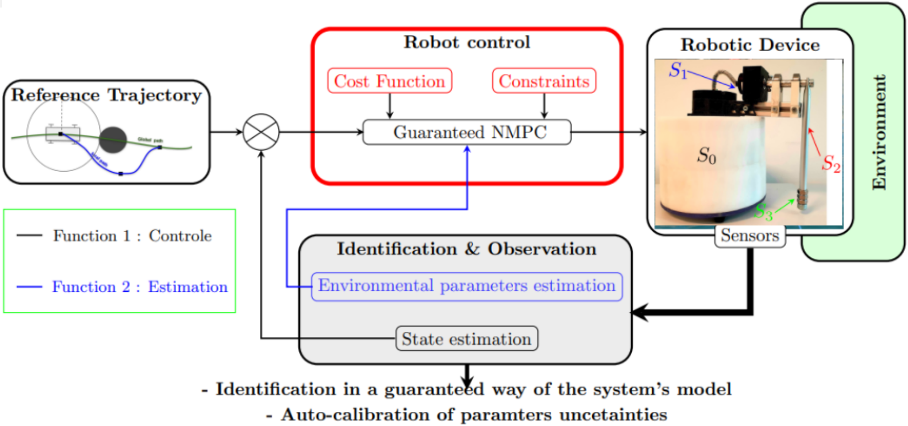
Challenges

- Complex dynamics (nonlinear ODEs)
- Uncertain models and environment

Scientific objectives of the presentation

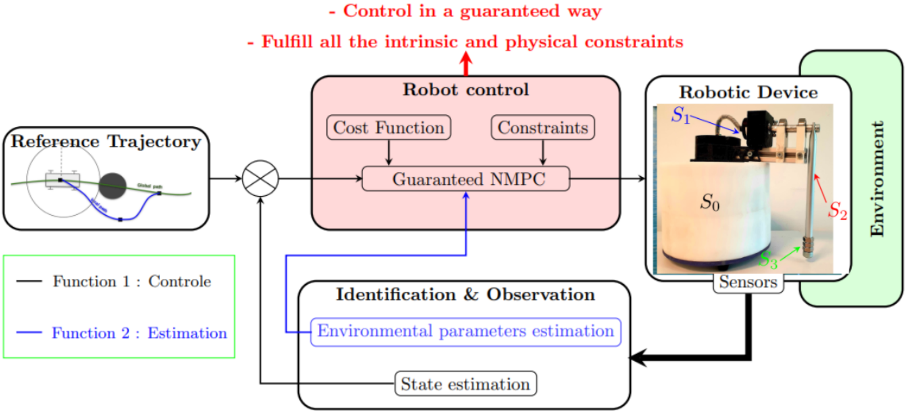
- Estimation of dynamic behavior using interval arithmetic
- Synthesis of reliable and constrained controllers robust to uncertainties

Main contributions



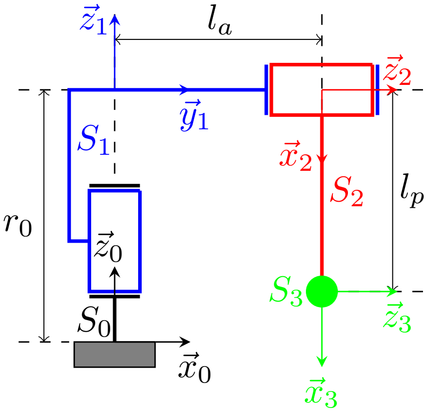
- **Addressed topics :**
  I. The accurate model requires a **guaranteed identification** handling all system modeling and design uncertainties

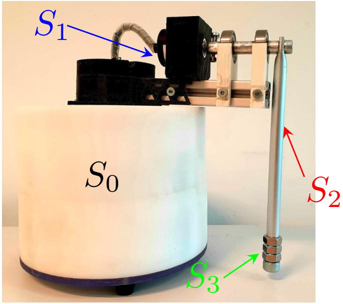
Main contributions



- **Addressed topics :**
  I. The accurate model requires a **guaranteed identification** handling all system modeling and design uncertainties
  II. Synthesis of a **guaranteed and validated NMPC control** based on this well-identified dynamic model

Preliminary Experimental Device

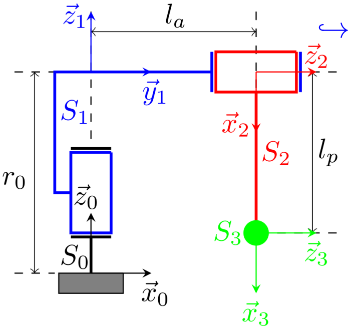




Outline

1. Part I : Guaranteed Dynamic Parameters Identification
2. Part II : Reliable NMPC via Validated Simulation
3. Conclusion and Future Works

Recall of the Dynamic Modeling



↪ The dynamics of the robot is given by the nonlinear equation

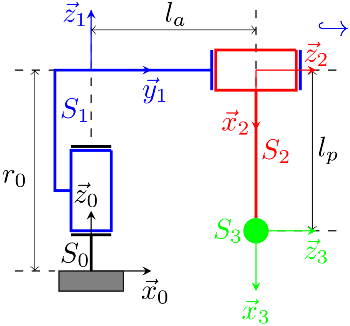
$$\Gamma = M\ddot{q} + B(q,\dot{q}) + Q(q),$$

$$M(q) = \begin{bmatrix} \mu_1 \sin(q_2)^2 + \mu_2 & \mu_3 \cos(q_2) \\ \mu_3 \cos(q_2) & \mu_4 \end{bmatrix},$$

$$\Gamma = \begin{bmatrix} \tau - f_{v_1}\dot{q}_1 \\ -f_{v_2}\dot{q}_2 \end{bmatrix}, \quad Q(q) = \begin{bmatrix} 0 \\ \mu_g \sin(q_2) \end{bmatrix},$$

$$B(q,\dot{q}) = \begin{bmatrix} -\mu_3 \sin(q_2)\dot{q}_2^2 + 2\mu_1 \cos(q_2)\sin(q_2)\dot{q}_1\dot{q}_2 \\ -\mu_1 \cos(q_2)\sin(q_2)\dot{q}_1^2 \end{bmatrix},$$

Recall of the Dynamic Modeling



↪ The dynamics of the robot is given by the nonlinear equation

$$\Gamma = M\ddot{q} + B(q,\dot{q}) + Q(q),$$

$$M(q) = \begin{bmatrix} \mu_1 \sin(q_2)^2 + \mu_2 & \mu_3 \cos(q_2) \\ \mu_3 \cos(q_2) & \mu_4 \end{bmatrix},$$

$$\Gamma = \begin{bmatrix} \tau - f_{v_1}\dot{q}_1 \\ -f_{v_2}\dot{q}_2 \end{bmatrix}, \quad Q(q) = \begin{bmatrix} 0 \\ \mu_g \sin(q_2) \end{bmatrix},$$

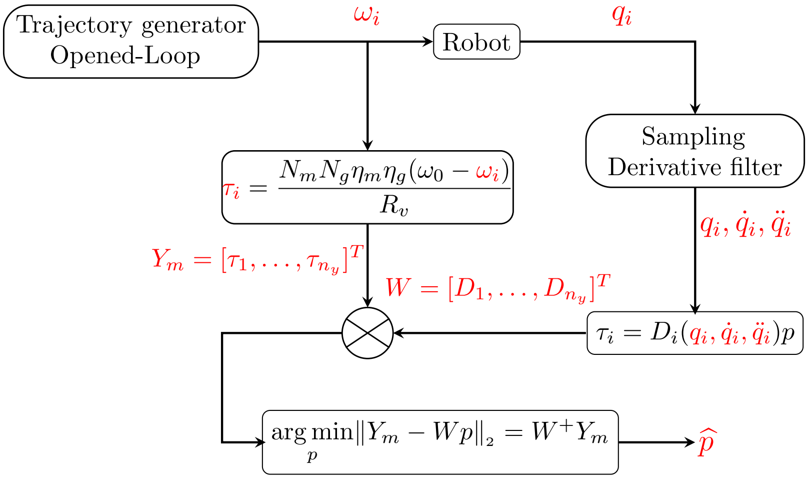
$$B(q,\dot{q}) = \begin{bmatrix} -\mu_3 \sin(q_2)\dot{q}_2^2 + 2\mu_1 \cos(q_2)\sin(q_2)\dot{q}_1\dot{q}_2 \\ -\mu_1 \cos(q_2)\sin(q_2)\dot{q}_1^2 \end{bmatrix},$$

↪ This inverse dynamic model can simply be written as,

$$y_m = f(q,\dot{q},\ddot{q},p),$$

$$p = [\mu_1, \mu_2, \mu_3, \mu_4, \mu_g, f_{v_1}, f_{v_2}] \in \mathbb{R}^{n_p=7}$$

(1) Identification with Classical Least Square Method (LSMI)



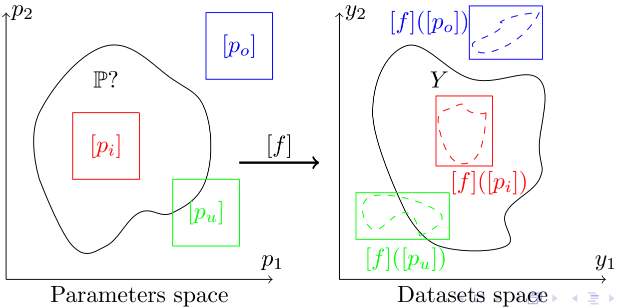
(2) Guaranteed Identification with bounded-error framework

Identification based on Interval Analysis and Set-Inversion tools

- ↪ Interval analysis yields methods to compute intervals in place of real numbers.
- ↪ Enclosing uncertainties coming from the system modeling and manufacturing.
- ↪ SIVIA algorithm is used to find the set of all possible viscous friction coefficients.

Hypothesis

Uncertainties and errors are bounded with known prior bounds



(2) Guaranteed Identification with bounded-error framework

Identification based on Interval Analysis and Set-Inversion tools

- ↪ Interval analysis yields methods to compute intervals in place of real numbers
- ↪ Enclosing uncertainties coming from the system modeling and manufacturing
- ↪ SIVIA algorithm is used to find the set of all possible viscous friction coefficients

Hypothesis

Uncertainties and errors are bounded with known prior bounds

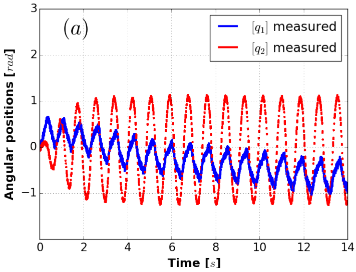
$$\mathbb{P}_i = \Big\{ p \in [\mathbf{p}] \;\mid\; \exists q(i) \in [\mathbf{q}](i),\; \exists \dot{q}(i) \in [\dot{\mathbf{q}}](i),\; \exists \ddot{q}(i) \in [\ddot{\mathbf{q}}](i) \;\; \text{s.t.} \;\; f(q(i), \dot{q}(i), \ddot{q}(i), p) \in [\mathbf{y}](i) \Big\}$$

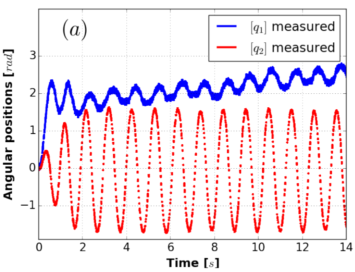
A set of feasible dynamic parameters

$$\widehat{\mathbb{P}} = \bigcap_{i=1}^{n} \mathbb{P}_i = f^{-1}([\mathbf{Y}]) \cap [\mathbf{p}]$$

Experiments

To allow high accurate identification, **two data acquisition were recorded** (#1 and #2), with **significant dynamics**, at the sampling period of $16ms$.



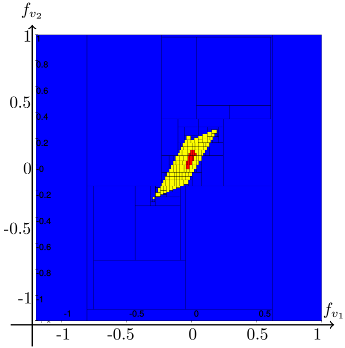


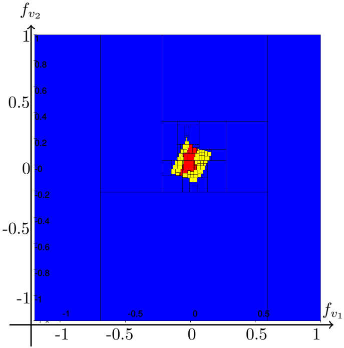
Symbol	Values with #1	Values with #2	CAD Values
$\mu_1[kg.m^2]$	$1.04082e^{-3}$	$1.06171e^{-3}$	$1.18152e^{-3}$
$\mu_2[kg.m^2]$	$2.56211e^{-3}$	$2.37309e^{-3}$	$2.55064e^{-3}$
$\mu_3[kg.m^2]$	$8.21e^{-4}$	$8.29e^{-4}$	$7.59e^{-4}$
$\mu_4[kg.m^2]$	$1.31781e^{-3}$	$1.19171e^{-3}$	$1.18152e^{-3}$
$\mu_g[kg.m^2.s^{-2}]$	$7.4175e^{-2}$	$7.2591e^{-2}$	$7.3575e^{-2}$
$f_{v_1}[N.m.s]$	$8.13e^{-2}$	$7.02e^{-2}$	—
$f_{v_2}[N.m.s]$	$6.42e^{-4}$	$5.93e^{-4}$	—

Identification results by LSMI method

Experiments

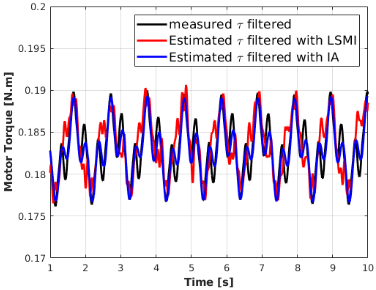
Symbol	Mean value	Uncertainty
$\mu_1 [kg.m^2]$	$1.09468e^{-3}$	$\pm 12\%$
$\mu_2 [kg.m^2]$	$2.49528e^{-3}$	$\pm 12\%$
$\mu_3 [kg.m^2]$	$8.03e^{-4}$	$\pm 10\%$
$\mu_4 [kg.m^2]$	$1.23035e^{-3}$	$\pm 12\%$
$\mu_g [kg.m^2.s^{-2}]$	$7.3447e^{-2}$	$\pm 15\%$





$$[\mathbf{f_{v1}}] \times [\mathbf{f_{v2}}] = [0.043012, 0.13002] \times [0.000454, 0.001174]$$

Results Validation - via Cross-Validation



↪ The RMSE is around 6.4% with LSMI against 2.6% with IA method, which indicates a reliable fit and a good coherence when the variables uncertainties are accounted.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\tau_i - \widehat{\tau}_i\right)^2}$$

Results Validation - via DynIbex Library

$\hookrightarrow$ Consider an IVP for nonlinear ODEs, over the time interval $[0, T]$

$$\begin{cases} \dot{x}(t) = \mathbf{f}(t, x(t), u(t), \color{red}{[\mathbf{p}]}) \\ x_0 \in [\mathbf{x}_0] \subseteq \mathbb{IR}^4 \\ u_0 \in [\mathbf{u}_0] \subseteq \mathbb{IR} \end{cases}$$

This IVP (Cauchy problem) has a unique solution $x(t; x_0; u_0)$ if $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ is Lipschitz in x and u, but for our purpose we suppose $\mathbf{f}$ smooth enough, i.e., of class C^k

Purpose

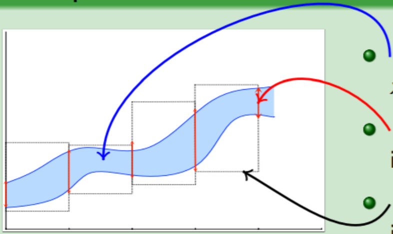
Solve in a guaranteed way ODEs from sets of initial values and bounded parameters

Results Validation - via DynIbex Library

Goal of validated numerical integration (DynIbex solver)

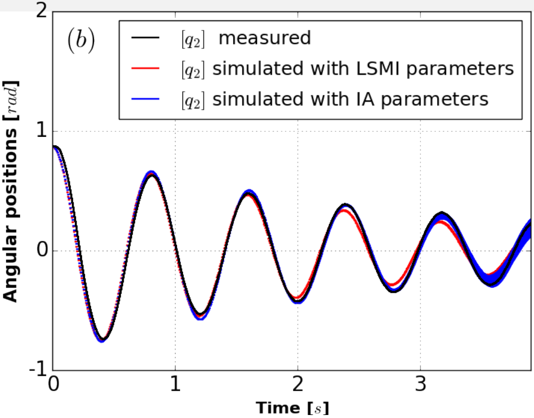
- Compute a sequence of time instants : $t_0 = 0 < t_1 < \cdots < t_n = T$
- Compute a sequence of values : $[\mathbf{x}_0], \ldots, [\mathbf{x}_n]$ such that $\forall i \in [0, n],\ x(t_i; x_0; u_0) \in [\mathbf{x}_i]$
- and a sequence $[\tilde{\mathbf{x}}_0], \ldots, [\tilde{\mathbf{x}}_{n-1}]$ such that $\forall i \in [0, n-1],\quad x(t; x_0; u_0) \in [\tilde{\mathbf{x}}_i],\quad \forall t \in [t_i, t_{i+1}]$

Example



- Exact solution of $\dot{x} = f(x)$ with $x(0) \in \mathcal{X}$.
- Safe approximation at discrete time instants.
- Safe approximation between time instants.

Results Validation - via DynIbex Library

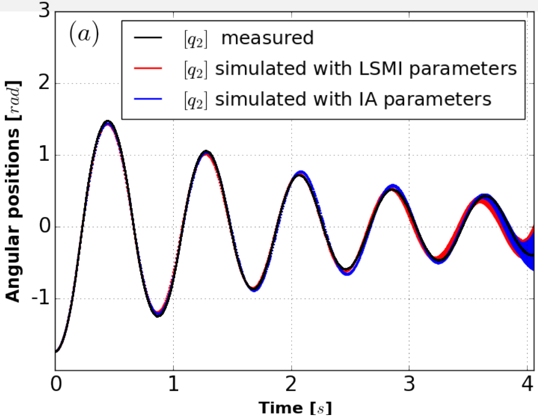


$$[\mathbf{X}_0] = \underbrace{[0^\circ, 0^\circ]}_{[x_1]} \times \underbrace{[0,0]}_{[x_2]} \times \underbrace{[46^\circ, 50^\circ]}_{[x_3]} \times \underbrace{[0,0]}_{[x_4]} \quad \text{and} \quad [\mathbf{U}_0] = [0.15Nm, 0.18Nm]$$

$$\text{Coverage ratio} = \frac{N_q}{N_t} \times 100$$

Scenario	(b)
With IA method	61%
With LSMI method	34%

Results Validation - via DynIbex Library



$$[\mathbf{X}_0] = \underbrace{[0^\circ, 0^\circ]}_{[x_{1_0}]} \times \underbrace{[0, 0]}_{[x_{2_0}]} \times \underbrace{[-98^\circ, -100^\circ]}_{[x_{3_0}]} \times \underbrace{[0, 0]}_{[x_{4_0}]} \quad \text{and} \quad [\mathbf{U}_0] = [0Nm, 0Nm]$$

$$\text{Coverage ratio} = \frac{N_q}{N_t} \times 100$$

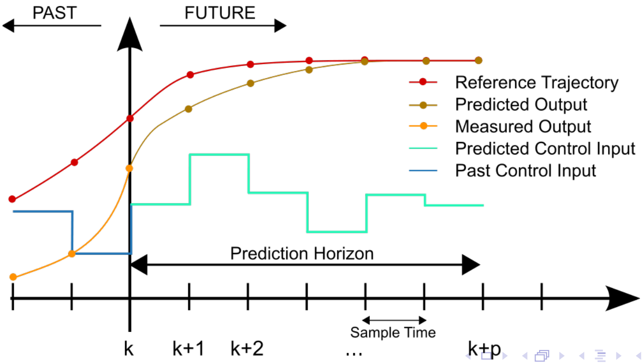
Scenario	(a)
With IA method	51%
With LSMI method	38%

Outline

General Concept of NMPC Control

Starting from measurements at time t :

1. Prediction over an horizon $T_p = N_p \times T_c$ (N_p the number of pre-computed inputs and T_c the control sampling time)
2. Computation of optimal inputs $U = \{u_1, \ldots, u_{N_p}\}$
3. Only the first input u_1 is injected into the system
4. $t = t + T_p$ (sliding) and goto 1



Optimization Cost Function

The mathematical formulation is :

$$\text{Find} \quad \hat{u}(.) \quad = \quad \underset{u(.)}{\text{argmin}}\ J(x(t), u(.))$$

Subject to :

$$\begin{aligned} \dot{x}(t) &= \mathbf{f}(t, x(t), u(t)) \\ u(t) &\in \mathbb{U}, \quad \forall t \geq 0 \\ x(t) &\in \mathbb{X}, \quad \forall t \geq 0 \end{aligned}$$

Optimization Cost Function

The mathematical formulation is :

$$\text{Find} \quad \hat{u}(.) \quad = \quad \underset{u(.)}{\text{argmin}}\ J(x(t), u(.))$$

Subject to :

$$\begin{aligned} \dot{x}(t) &= \mathbf{f}(t, x(t), u(t)) \\ u(t) &\in \mathbb{U}, \quad \forall t \geq 0 \\ x(t) &\in \mathbb{X}, \quad \forall t \geq 0 \end{aligned}$$

The continuous cost function can be derived as,

$$J(x(t), u(.)) = \int_{t}^{t+T_p} F(x(\tau), u(\tau)) d\tau$$

F is in general a quadratic function such that :

$$F(x, u) = (x - x_r)^T \, \mathbf{Q} \, (x - x_r) + u^T \mathbf{R} u$$

Guaranteed numerical integration : Runge-Kutta methods

↪ Consider an IVP for nonlinear ODEs, over the time interval $[0, T]$

$$\begin{cases} \dot{x}_t = \mathbf{f}(t, x_t, u, [\mathbf{p}]) \\ x_0 \in [\mathbf{x}_0] \subseteq \mathbb{IR}^4 \\ u \in [\mathbf{u}] \subseteq \mathbb{IR} \end{cases}$$

- A standard numerical integration method computes a sequence of values $(t_j, x_j)_{j \in \mathbb{N}}$ approximating the solution of the ODE such that $x_{j+1} \approx \mathbf{f}(t_j, x_j, u)$
- s-stage Runge-Kutta methods is defined by the following recurrence (h : time-step) :

$$k_i = \mathbf{f}(t_0 + c_i h, x_j + h \sum_{k=1}^{s} a_{ik} k_i, u) \qquad x_{j+1} = x_j + h \sum_{k=1}^{s} b_k k_i$$

The coefficient c_i, a_{ik} and b_k characterize the Runge-Kutta methods and their are usualy synthesized in a *Butcher tableau*

Guaranteed numerical integration : Runge-Kutta methods

↪ Consider an IVP for nonlinear ODEs, over the time interval $[0, T]$

$$\begin{cases} \dot{x}_t = \mathbf{f}(t, x_t, u, [\mathbf{p}]) \\ x_0 \in [\mathbf{x}_0] \subseteq \mathbb{IR}^4 \\ u \in [\mathbf{u}] \subseteq \mathbb{IR} \end{cases}$$

- A standard numerical integration method computes a sequence of values $(t_j, x_j)_{j\in\mathbb{N}}$ approximating the solution of the ODE such that $x_{j+1} \approx \mathbf{f}(t_j, x_j, u)$
- s-stage Runge-Kutta methods is defined by the following recurrence (h : time-step) :

$$k_i = \mathbf{f}(t_0 + c_i h, x_j + h \sum_{k=1}^{s} a_{ik} k_i, u) \qquad x_{j+1} = x_j + h \sum_{k=1}^{s} b_k k_i$$

  The coefficient c_i, a_{ik} and b_k characterize the Runge-Kutta methods and their are usualy synthesized in a *Butcher tableau*
- The purpose of a validated or guaranteed numerical integration method is to compute the sequence of boxes $(t_j, [\mathbf{x}_j])_{j\in\mathbb{N}}$ such that $[\mathbf{x}_{j+1}] \supseteq [\mathbf{f}](t_j, [\mathbf{x}_j], [\mathbf{u}])$
  $\Rightarrow$ DynIbex library is developed for this purpose

Validated NMPC - Global Approach

- Based on two stages :
  - **Filtering and branching :** uses the validated simulation methods to compute the N_p guaranteed inputs $[\mathbf{U}] = [\mathbf{u}_1] \times [\mathbf{u}_2] \times \ldots \times [\mathbf{u}_{N_p}]$ ensuring state variables constraints ($x_i \in [\mathbf{x}_i]$) and convergence to the reference set ($x_i \to [\mathbf{x}_r]$) (DynIbex)
  - **Interval optimization :** from safe input boxes, the optimization stage aims to compute the sub-optimal input interval that minimizes as much as we can the formulated interval cost-function

Require : $[\mathbf{x}_r], [\mathbf{x}_0], N_p, T_p, T_c, T_f$;
while $t \leq T_f$ **do**
  acquire ($[\mathbf{x}_t]$);
  $[\mathbf{U}] = \text{Filtering}([\mathbf{U}], [\mathbf{x}_t], [\mathbf{x}_r], T_c)$;
  $\hat{u}_1 = \text{Optimization}([\mathbf{U}], [\mathbf{x}_t], [\mathbf{x}_r])$;
  send($\hat{u}_1$);
  $t = t + T_c$;
end

(1) Filtering and branching

Main algorithm

- Inputs
  - the initial conditions interval $[\mathbf{x}_0]$
  - the set-point interval $[\mathbf{x}_r]$
  - NMPC parameters N_p, T_c, T_p
  - Simulation time T_f
  - input bounds, i.e., $\forall k, \quad u_{\min} \leq u_k \leq u_{\max}$
  - state bounds, i.e., $\forall t, \quad x_{\min} \leq x_t \leq x_{\max}$
  - initial control box (actuator's bounds) $[\mathbf{U}] = [\mathbf{u}_1] \times \ldots \times [\mathbf{u}_{N_p}]$
- The main steps are (in a loop) :
  1. validated simulation of the IVP-ODE to compute the new state domain over the sampling time T_c, $[\mathbf{x}_{t+T_c}] = [\mathbf{f}](t, [\mathbf{x}_t], [\mathbf{u}_k])$
  2. if $[\mathbf{x}_{t+T_c}] \subseteq [\mathbf{x}_r]$, so $k = k+1$, $[\mathbf{x}_t] = [\mathbf{x}_{t+T_c}]$, $t = t + T_c$ and goto 1
  3. else successive bisections of the input interval $[\mathbf{u}_k]$ to minimize its width
  4. Re-starting validated simulation, one side of bisected intervals is kept by considering these criteria :
     - (i) a branch leading to unsafe state is removed (i.e., $[\mathbf{x}_{t+Tc}] \not\subseteq [x_{\min}, x_{\max}]$)
     - (ii) a branch leading to a state far from the reference interval $[\mathbf{x}_r]$ is eliminated
     - (iii) a branch leading to the opposite "ways" is avoided (sensitivity analysis)
  5. $k = k+1$, $[\mathbf{x}_t] = [\mathbf{x}_{t+T_c}]$, $t = t + T_c$ and goto 1

(2) Interval Optimization

- NMPC needs to evaluate the integral of the interval cost function expressed from safe computed control box and state intervals at each prediction horizon N_p
- Using rectangle rule, we can write :

$$
\begin{aligned}
J(y,u) &= \int_{t}^{t+T_p} \left[(x(\tau)-x_r)^T \mathbf{Q} \, (x(\tau)-x_r) + u(\tau)^T \mathbf{R} u(\tau) \right] d\tau \\
&= \sum_{k=1}^{N_p} \int_{t+(i-1)T_c}^{t+iT_c} \left[(x(\tau)-x_r)^T \mathbf{Q} \, (x(\tau)-x_r) \right] d\tau + \; T_c \sum_{i=1}^{N_p} \left[u_k^T \mathbf{R} u_k \right] \\
&\in T_c \sum_{k=1}^{N_p} [\tilde{\mathbf{x}}_k]^T \, \mathbf{Q} \, [\tilde{\mathbf{x}}_k] + \; T_c \sum_{k=1}^{N_p} [\mathbf{u}_k]^T \, \mathbf{R} \, [\mathbf{u}_k] \\
&\leq ub \left\{ T_c \sum_{k=1}^{N_p} \left[[\tilde{\mathbf{x}}_k]^T \, \mathbf{Q} \, [\tilde{\mathbf{x}}_k] + \; [\mathbf{u}_k]^T \, \mathbf{R} \, [\mathbf{u}_k] \right] \right\}
\end{aligned}
$$

(2) Interval Optimization

Main algorithm of the optimization procedure :

Require : $[\mathbf{x}_r], [\mathbf{x}_1], \dots, [\mathbf{x}_{N_p}], [\mathbf{u}_1], \dots, [\mathbf{u}_{N_p}]$, Tol;
while $w([\boldsymbol{u}_1]) \geq Tol$ **do**
    $[\mathbf{U}]_{left} = [\mathbf{u}_1]_l \times [\mathbf{u}_2] \times \dots \times [\mathbf{u}_{N_p}]$;
    $[\mathbf{U}]_{right} = [\mathbf{u}_1]_r \times [\mathbf{u}_2] \times \dots \times [\mathbf{u}_{N_p}]$;
    if $J([\boldsymbol{X}], [\boldsymbol{U}]_{left}) \geq J([\boldsymbol{X}], [\boldsymbol{U}]_{right})$ **then**
        $[\mathbf{U}] = [\mathbf{U}]_{right}$;
    else
        $[\mathbf{U}] = [\mathbf{U}]_{left}$;
    end
end
$\hat{u}_1 = \min[lb([\mathbf{u}_1]),\ ub([\mathbf{u}_1])]$;
send $(\hat{u}_1)$;

Sub-optimal solution

But robust to uncertainties !

Experiments results of the Validated NMPC

Inverted pendulum constraints and NMPC parameters

- Parameters for NMPC :

$$N_p = 10, \;\; T_c = 0.01, \;\; T_p = 0.1 \;\; \text{and} \;\; T_f = 0.4$$

- State and Input constraints :

$$\forall t : \quad \begin{array}{c} x_1(t) \in [-\pi, \pi] \\ x_2(t) \in [-40, 40] \\ x_3(t) \in [-\pi, \pi] \\ x_4(t) \in [-50, 50] \\ u \in [-8.1, 8.1] \end{array}$$

- Goal :

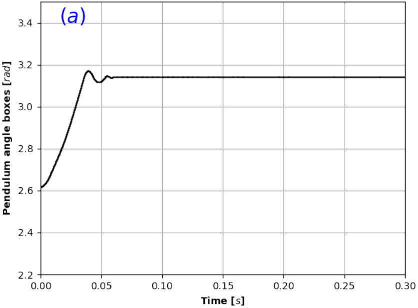
$$q_2 \in [\pi - 0.1, \pi + 0.1]$$

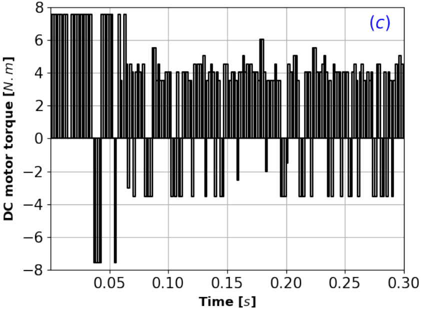
- Weighing matrices :

$$\mathbf{R} = 0.5 \quad \text{and} \quad \mathbf{Q} = diag[1000, 1000, 1000, 1000]$$

Experiments results of the Validated NMPC

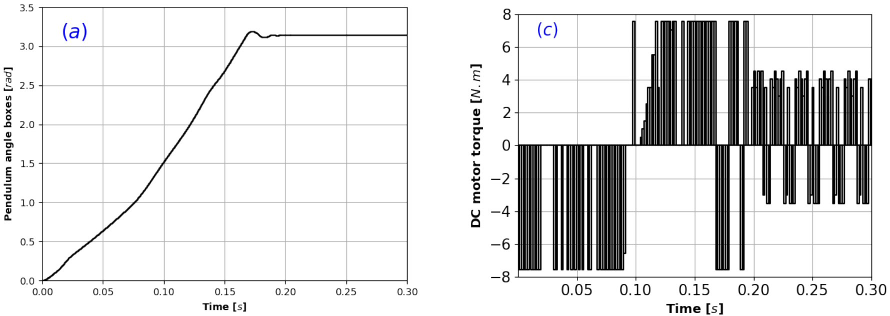
Validated NMPC results starting from $[\mathbf{x}_{3_0}] = [149^\circ, 151^\circ]$





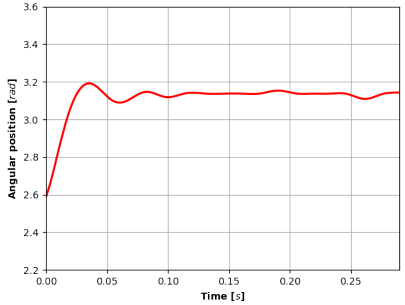
Experiments results of the Validated NMPC

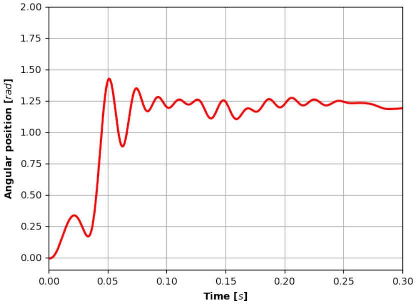
Validated NMPC results starting from $[\mathbf{x}_{3_0}] = [0°, 4°]$



Experiments results of the Validated NMPC

Experimental validation using real inverted pendulum





Advantages of the proposed validated NMPC

☺ Robust to uncertainties ☺ Constraints satisfaction ☺ Optimal controller

Drawbacks

☹ Huge computation time (real-time problem)

Outline

Conclusion and Outlook

Summary

- **Contributions in topic I : Guaranteed identification**
  - Guaranteed identification of dynamic parameters with interval analysis and set-inversion tools
  - Validated numerical integration (DynIbex) to compute tight enclosures of state variables
- **Contributions in topic II : Guaranteed NMPC**
  - New formulation of a guaranteed NMPC strategy via validated simulation
  - Robust to uncertainties with constraints satisfaction

Conclusion and Outlook

Summary

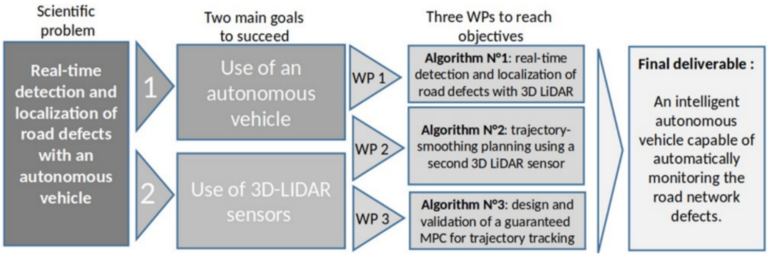
- **Contributions in topic I : Guaranteed identification**
  - Guaranteed identification of dynamic parameters with interval analysis and set-inversion tools
  - Validated numerical integration (DynIbex) to compute tight enclosures of state variables
- **Contributions in topic II : Guaranteed NMPC**
  - New formulation of a guaranteed NMPC strategy via validated simulation
  - Robust to uncertainties with constraints satisfaction

Future Works

- Real-time improvements (efficient onboard-GPU, distributed computing and relaxation methods)
- Projection of all our algorithms on a complex systems (e.g., underwater robot)
- On-line estimation and self-calibration of the environment parameters via Interval Arithmetic (i.e., aerodynamics coefficients)

Projet ANR JCJC - AutoROAD (submitted)

Autonomous RObotic System for Detection And Location of Pavement Defects : Application to Road Network State Evaluation



References

M., Fnadi, and J., Alexandre dit Sandretto. Experimental Validation of a Guaranteed Nonlinear Model Predictive Control. Algorithms, 14(8), 248, 2021.

M., Fnadi, J. Alexandre dit Sandretto, G., Ballet, and L. Fribourg. Guaranteed Identification of Viscous Friction for a Nonlinear Inverted Pendulum Through Interval Analysis and Set Inversion. 2021 American Control Conference (ACC), 3920-3926.

L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. Interval analysis. In Applied interval analysis. pp. 11-43, 2001. Springer, London.

Julien Alexandre Dit Sandretto, and Alexandre Chapoutot. Validated explicit and implicit Runge-Kutta methods. Reliable Computing electronic edition 22 (2016).

Frank Allgower, Rolf Findeisen, and Zoltan K. Nagy. Nonlinear model predictive control : From theory to application. Journal-Chinese Institute Of Chemical Engineers 35.3 (2004) : 299-316.

L. Jaulin, and E. Walter. Set inversion via interval analysis for nonlinear bounded-error estimation. 1993. Automatica, 29(4), pp.1053-1064.

Thank you for your attention ☺

medfanadi@gmail.com