Interval trajectory tracking with flatness

Olivier Mullier   Estelle Courtial

U2IS, ENSTA Paristech
olivier.mullier@polytechnique.edu

Laboratoire PRISME, Polytech Orléans
estelle.courtial@univ-orleans.fr

20 juin 2016

This work was supported by an initiative of the French Ministry of Higher Education and Research “Investissements d’Avenir”, through the labex VOLTAIRE (ANR-10-LABX-100-01).

Context: CIPEGE[1]

CIPEGE goal

Improve the employability of university students in Earth Sciences.

The CIPEGE tool

A decision making tool to satisfy qualitatively (level of study) and quantitavely (number of graduated) the labor market.

[1] International prospective employment center in Earth sciences and environment (Centre international de prospective pour l'emploi en géosciences et en environnement)

Context

Discrete time nonlinear system:

$$\begin{cases} x(k+1) = f(x(k), u(k)), \ x(0) = x_0 \\ y(k) = h(x(k)) \end{cases}$$

avec

- $x(k) \in \mathbb{R}^n$ the state;
- $y(k) \in \mathbb{R}^p$ the output;
- $u(k) \in \mathbb{R}^m$ the input (control) ;
- x_0 the initial condition ;
- $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ and $h : \mathbb{R}^n \to \mathbb{R}^p$ are two analytic functions.

Context

Discrete time nonlinear system:

$$\begin{cases} x(k+1) = f(x(k), u(k)),\ x(0) = x_0 \\ y(k) = h(x(k)) \end{cases}$$

avec

- $x(k) \in \mathbb{R}^n$ the state;
- $y(k) \in \mathbb{R}^p$ the output;
- $u(k) \in \mathbb{R}^m$ the input (control) ;
- x_0 the initial condition ;
- $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ and $h : \mathbb{R}^n \to \mathbb{R}^p$ are two analytic functions.

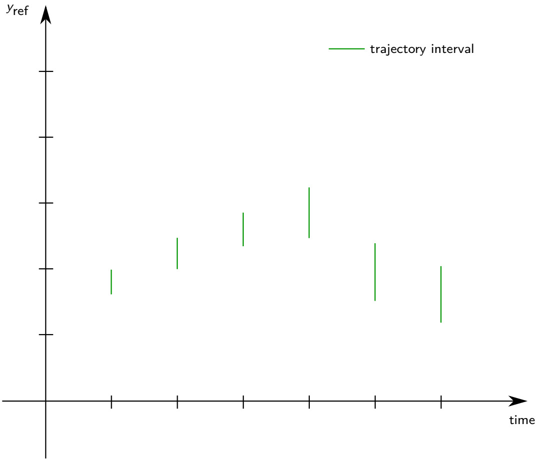
Objective

Determine the input $u(k)$ to track the *reference trajectory* $y_{\text{ref}}(k)$ of the output $y(k)$ over a given prediction horizon.

Uncertain trajectory tracking

We consider uncertain trajectory

$$y_{\text{ref}}(k) \in [y_{\text{ref}}(k)]$$



Issue

Goal

Characterize the set of admissible controls at each time k such that the ouput value remains in reference trajectory intervals.

$$\mathcal{U}^*(k) = \{u(k) | h(f(x(k), u(k))) \in [y_{\text{ref}}(k+1)]\}$$

Computing such set is generally intractable.

$\Rightarrow$ Computation of an inner approximation of the solution set.

Outline

Outline

"Global numerical approach to nonlinear discrete-time control", IEEE Trans. on Automatic Control [Jaulin, Walter (1997)]

The result is a subpaving where all boxes belong to the characterized image set.

$$\mathcal{U}_{1..\ell} = \{(u(1), ..., u(\ell)) | \forall i \in [1, \ell], f(x(i), u(i)) \in [y_{\text{ref}}(i+1)]\}$$

with $y(k) = x(k)$.

Drawbacks

- **Computation time** : time complexity of the branch & Prune algorithm is in $O(e^{m \times \ell})$ with m the dimension of u and ℓ the prediction horizon.
- **Precision** : The precision is a user defined parameter according the stopping criterion:
  - minimum size of the considered intervals;
  - allocated time for computation.
- **Restriction** : the reference trajectory is defined over the entire (whole) state vector $(y(k) = x(k))$.

"Set computation for nonlinear control", Reliable Computing [Jaulin, Ratschan et Hardouin (2004)]

The result is a validated sequence of controls on a prediction horizon ℓ :

$$(\tilde{u}(1), ..., \tilde{u}(\ell)) \in \mathcal{U}_{1..\ell} = \{(u(1), ..., u(\ell)) | \forall i \in [1, \ell], f(x(i), u(i)) \in [y_{\text{ref}}(i+1)]\}$$

with $y(k) = x(k)$.

Drawbacks

- **Computation time** : complexity is reduced since at each time i the control is fixed;
- **Precision** : we only get one sequence of controls;
- **Restriction** : the reference trajectory is defined over the entire (whole) state vector $(y(k) = x(k))$.

Outline

Objectives

1. **Restriction :** the trajectory can be on a part of the components of the state vector;
2. **Computation time and precision :** enhance time computation according to the precision.

Proposed method

Use of the flatness concept for dynamic systems.

Flatness (Fliess et al.)

Definition of x and u in function of a set of fundamental variables of the system: the *flat output*.

$$\begin{cases} x(k+1) = f(x(k), u(k)) \\ y(k) = h(x(k)). \end{cases}$$

This system is flat if there exist for a certain integer M, a function μ of $x(k)$ and $u(k)$ such that

$$F_k = \mu(x(k), u(k), \ldots, u(k+M)).$$

F_k is called the flat output and verify the following relations for all k and for all $J = n - 1$:

$$\begin{aligned} x(k) &= \psi(F_k, F_{k+1}, \ldots, F_{k+J}); \\ y(k) &= h \circ \psi(F_k, F_{k+1}, \ldots, F_{k+J}); \\ u(k) &= \varphi(F_k, F_{k+1}, \ldots, F_{k+J+1}). \end{aligned}$$

Computation of the set of admissible control for a flat system

Flatness gives

$$u(k) = \varphi(F_k, F_{k+1}, \ldots, F_{k+J+1})$$

If we consider that the reference trajectory gives the appropriate values for the flat ouput, The set to characterize at each time k is

$$\mathcal{U}^*(k) = \{\varphi(F_k, F_{k+1}, \ldots, F_{k+J+1}) | F_k \in [y_{\text{ref}}(k)] \, \forall k\}$$

This corresponds to the computation of the set

$$\{\mathcal{F}(x) | x \in [x]\}.$$

Two distinctive cases

Computation of an inner approximation of the set

- $u \in \mathbb{R}$ (SISO) : generalized affine forms ;
- $u \in \mathbb{R}^m$, $m \geqslant 1$ (MIMO) : Branch & Prune algorithm.

SISO case

To define a generalized affine form, one need to:

- compute $[\Delta_i]$, an outer approximation of the Jacobian of f over a box;
- compute $f^{\varepsilon}(t_1, \ldots, t_n)$ for a given $(t_1, \ldots, t_n) \Rightarrow$ corresponding to the center by taking $t = 0$.

Generalized affine set[1]

A generalized affine set of a function is a triplet (Z, c, J) with

- $Z \in \mathcal{M}(n + m + 1, p)$ a vector of affine forms;
- $c \in \mathbb{R}^p$ a vector corresponding to the centre;
- $J \in (\mathcal{M}(n, p))^n$ a matrix of affine forms.

Combining affine arithmetics and Kaucher arithmetics. The generalized affine forms allow to produce an interval $[y] \subseteq \{f(x) | x \in [x]\}$.

E. Goubault, O. Mullier, S. Putot, and M. Kieffer.
Inner approximated reachability analysis.
In *Hybrid Systems: Computation and Control*, pages 163–172, New York, NY, USA, 2014. ACM.

MIMO case

Employed method

- Extension of the results from Jaulin-Goldsztejn to the case $f : \mathbb{R}^n \to \mathbb{R}^m$;
- Computation of a subpaving of the set $\{f(x) | x \in [x]\}$;
- Branch & Prune algorithm + sufficient condition for a box $[y]$ to belong to $\{f(x) | x \in [x]\}$.

Outline

SISO Example

$$\begin{cases} x_1(k+1) = x_1(k)x_2(k) \\ x_2(k+1) = x_2(k) + u(k) \end{cases}$$

with $x_0 = (1\ 1)^T$.
We have computed the flat output F_k and the expression of $u(k)$ according to the flat output:

$$u(k) = \frac{F_{k+2}}{F_{k+1}} - \frac{F_{k+1}}{F_k}$$

and we know the reference trajectory

$$\begin{aligned} (y_{\text{ref}}(k))_{k=1..3} &= (F_1, F_2, F_3) \\ &= (1, [1,2], [5,7]). \end{aligned}$$

We apply the both presented methods to compute $u(0)$ and $u(1)$.

- Method 1 : generalized affine forms.
- Method 2 : intervals.

Method 1 (Generalized affine forms)

$$(F_1, F_2, F_3) = (1, [1, 2], [5, 7])$$

$$u(k) = \frac{F_{k+2}}{F_{k+1}} - \frac{F_{k+1}}{F_k}$$

We compute the affine forms corresponding to F_i :

$$\begin{aligned}
\hat{F}_0 &= 1 \\
\hat{F}_1 &= 1 \\
\hat{F}_2 &= \frac{3}{2} + \frac{1}{2}\varepsilon_2 \\
\hat{F}_3 &= 6 + \varepsilon_3
\end{aligned}$$

Method 1 (Generalized affine forms)

The computation gives:

- for $u(0)$

$$\check{u}_0 = \begin{pmatrix} \hat{u}_0 & = \frac{1}{2} + \frac{1}{2}\varepsilon_2 \\ c & = \frac{1}{2} \\ [J] & = \begin{pmatrix} 0 & \frac{1}{2} & 0 \end{pmatrix} \\ \hat{J} & = \begin{pmatrix} 0 & \frac{1}{2} & 0 \end{pmatrix} \end{pmatrix}.$$

- for $u(1)$

$$\check{u}_1 = \begin{pmatrix} \hat{u}_1 & = 3 - \frac{5}{4}\varepsilon_2 + \frac{3}{4}\varepsilon_3 + \eta \\ c & = \frac{5}{2} \\ [J] & = \begin{pmatrix} 0 & \left[-4, -\frac{9}{8}\right] & \left[\frac{1}{2}, 1\right] \end{pmatrix} \\ \hat{J} & = \left((0) \; \left(-\frac{23}{8} + 0.28125\varepsilon_2 - 0.395833\varepsilon_3 + 1.61458\eta\right) \right. \\ & \left. \quad (0.75 - 0.125\varepsilon_2 + 0.125\eta) \right) \end{pmatrix}.$$

Method 2 (bisection and intervals)

The bisection method requires to know the expression of the function and its derivatives:

$$u(k) = \frac{F_{k+2}}{F_{k+1}} - \frac{F_{k+1}}{F_k}$$

$$J_u = \begin{pmatrix} \frac{F_{k+1}}{F_k^2} & -\frac{F_{k+2}}{F_{k+1}^2} - \frac{1}{F_k} & \frac{1}{F_{k+1}} \end{pmatrix}$$

Results : discussion

Stopping criterion for method 2 and (Jaulin et al.) : minimum size for an interval to be treated is 0.1.

Method	Method 1	Method 2	(Jaulin et al.)
Results Computation time	$[0, 1]$ 1.37672×10^{-6}s	$[0, 1]$ 0.0002s	$[0, 0.9765625]$ 0.37935s[2]

Table: Results for $u(0)$.

Method	Method 1	Method 2	(Jaulin et al.)
Results Computation time	$[0.875, 4.125]$ 2.86311×10^{-6}s	$[0.59375, 5.6875]$ 0.013656s	$[0.8789, 5.2734]$ 0.37935s[1]

Table: Results for $u(1)$.

[2] time to compute $u(1)$ and $u(2)$ together

Outline

Contribution

New method to characterize the set of admissible controls

- enhances the result according to the existing approach in term of precision and computation time;
- combination of set-membership methods and flatness of dynamic systems;
- handle a larger class of problems ($y(k) \neq x(k)$).

Contribution

New method to characterize the set of admissible controls

- enhances the result according to the existing approach in term of precision and computation time;
- combination of set-membership methods and flatness of dynamic systems;
- handle a larger class of problems ($y(k) \neq x(k)$).

Perspectives

Extend the results to the class of continuous-time systems:

- control u is a function of the flat output and its successive time derivatives;
- use of validated integration methods.