Flatness-based Nonlinear Model Predictive Control strategy

Conference Paper · January 2005

3 authors:

Nejib ben nasr
Prince Sattam bin Abdulaziz University

M'Sahli Faouzi
University of Monastir

Ben Abdennour
National Engineering School of Gabes

SSD'05

Flatness-based Nonlinear Model Predictive Control strategy

BEN NASR N. M'SAHLI F. & BEN ABDENNOUR R.

Research Unit "CONPRI", National school of engineers of Gabes, Route de mednine, 6029, Gabes, TUNISIA

Abstract

The aim of this paper is to present a method that mixing Non-linear predictive control (NLPC) and flatness properties, in order to obtain high level performances in term of tracking. Recently, connection between the trajectory planning problem, flatness and Brunovsky controllability canonical form have been outlined , showing that flatness may be useful to reference trajectory design and predictive control . The advantage of mixing among flatness property and predictive control scheme is to obtain a better trajectory tracking with an error tending asymptotically to zeros. The notion of flatness has in common with the predictive control to put the accent on the predicted trajectories(feed forward), that means on the direct construction of reference trajectories of which initial and final condition are fixed, and that are submitted to supplementary constraints possibly. Simulation results have been carried out to improve the theory exposed in this paper.

Keyword:
predictive control, flatness, nonlinear system, trajectory generation.

1. Introduction

Several researchers have made recent contributions toward obtaining computationally viable methods for computing local solutions to nonlinear control problems [6,7, 11]. However, the obtained results are jugged sub-optimal. A critical need is the ability to compute optimal trajectories *very* quickly, so that they can be used in a real-time setting. For general problems this can be very difficult, but there are classes of systems for which simplifications can be made that vastly simplify the computational requirements for generating trajectories. In this paper we describe one such class of systems, so-called differentially flat systems. As garneted with flat systems, it suffices to control the flat output, to control the whole system [1].

Some recent work, devote to study flatness property for control problem of linear system, but the flat output characterization is so hard. A few authors gave methods that illustrate the flat output parameters. So to show the behavior of the flat output, it's necessary to make an observer or use one of technique that permits a direct calculation of the flat output. For more information see [2]. For nonlinear system the characterization of the flat output parameters isn't in the same way of linear system. Every problem has a specific method to find out his flat output.
In literature, many input-output nonlinear models have been proposed for controlling nonlinear systems that follow the MBPC strategies. Such as, those described by NARMAX, Hammerstein, Wiener and Volterra models [7,10].

2. Nonlinear predictive control

Despite of the wide exposure of and the intensive research efforts attracted over the past few decades on Nonlinear model predictive control (NLMPC), this control strategy is still being perceived as an academic concept rather than a practicable control technique. However, nonlinear model predictive control is gaining popularity in the industrial community. The formulations for these controllers vary widely, and almost the only common principle is to retain nonlinearities in the process model [3].
In nonlinear control, a *receding horizon* approach is typically used, which can be summarized in the following steps:

(1) At time k, solve, on-line, an open-loop optimal control problem over some future interval, taking into account the *current and future* constraints.
(2) Apply the first step in the optimal control sequence.

For all corresponding: Nejib.nasr@isetso.rnu.tn

(3) Receding strategy so that at each instant the horizon is displaced towards the future, which involves the application of the first control signal of the sequence calculated at each step.
The method presented in this paper can be summarized as follows:

2.1 Given a model for a nonlinear system:

The process to control is assumed to be represented by a mono-variable second order parametric Volterra model:

$$A(q^{-1})y(k) = c_0 + B_1(q^{-1})u(k) + B_2(q_1^{-1},q_2^{-1})u^2(k) + \frac{\varepsilon(k)}{\Delta(q^{-1})} \qquad (1)$$

Where y*(k)*, *u(k)* and ε*(k)* are respectively the output, the control input, and the system modeling error. $A(q^{-1})\, and\, B(q^{-1})$ are two polynomials of the backward shifting operator q^{-1} given by:

$$A(q^{-1}) = 1 + a_1 q^{-1} + \cdots\cdots + a_{na} q^{-na} \text{ ;}$$
$$B_1(q^{-1}) = 1 + b_{11} q^{-1} + \cdots\cdots + b_{1nb} q^{-nb} \text{ ;}$$

$B_2(q_1^{-1},q_2^{-1})$ represents the quadratic term of the Volterra model, this quantity is defined by:

$$B_2(q_1^{-1},q_2^{-1})u^2(k) = \sum_{n=0}^{nb}\sum_{m=n}^{nb} b_{2nm}u(k-n)u(k-m)$$

or under a sub-obtimal state space model $x_{k+1} = f(x_k, u_k)$, presented in [10] is given by

$$\begin{bmatrix} x^{(i)}(k+1) \\ x^{(o)}(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 & \cdots & & \\ 1 & 0 & \cdots & 0 & 0 & \cdots & & \\ & & & \vdots & & & & \\ & & & 1 & 0 & & & \\ & & & & \vdots & 0 & \cdots & 0 \\ & & & & \vdots & 1 & \cdots & 0 \\ & & & & \vdots & \vdots & & \\ & & & & \vdots & \cdots & 1 & 0 \end{bmatrix} \begin{bmatrix} x^{(i)}(k) \\ x^{(o)}(k) \end{bmatrix} + \begin{bmatrix} u(k) \\ 0 \\ \vdots \\ 0 \\ \varphi(x(k),u(k)) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad (2)$$

With

$$x^{(i)}(k) = [u(k-1), u(k-2), \cdots, u(k-n_b)]^T \;\; and$$
$$x^{(o)}(k) = [y(k), y(k-1), y(k-2), \cdots, y(k-n_b)]^T \qquad (3)$$

2.2 Determinate the incremental predictive equation:

The incremental predictive form of the parametric Volterra model can be expressed as a function of the current and future control increments [4]:

$$\hat{y}(k+j) = v_0^{(j)} + v_1^{(j)}(q^{-1})\Delta u(k+j) + v_2^{(j)}(q_1^{-1},q_2^{-1})\Delta u^2(k+j) \qquad (4)$$

with

$$v_0^{(j)} = c_{op} + G_j y(k) + \sum_{i=j+1}^{nb+j-1}[\delta_{1i} + \sum_{m=i}^{nb+j-1}\delta_{2im}\Delta u^*(k+j-m)]\Delta u^*(k+j-i) \qquad (5)$$

$$v_{1i}^{j} = \delta_{1i} + \sum_{m=j+1}^{nb1+j-1}\delta_{2im}\Delta u^*(k+j-m) \quad i=1,2,...j \qquad (6)$$

$v_{2im}^{j} = \delta_{2im}$ *i=1, 2, ….j* and *m=1,2…j* *(7)*

The effect of selecting the parameters and the coefficient of the predictive control are not investigated here, for more details see e.g. [4, 6].

2.3 Minimizing a cost function

Nonlinear model predictive control (NMPC) is a control strategy where application of nonlinear optimization methods is essential. A general mathematical formulation of this problem can be stated as follows [6, 7].

$$J = \sum_{j=1}^{hp}[y_c(k+j) - \hat{y}(k+j/k)]^2 + \sum_{j=1}^{hu}\lambda_{uj}\Delta u^2(k+j-1) \qquad (8)$$

h_P, h_u and λ_{uj} are respectively the prediction horizon, the control horizon and the control-weighting factor of the control increments .
Replacing the incremental output by his expression, the cost function (8) can be written as follows:

$$J = (v_0^* + v_1\tilde{u} + v_2\tilde{u}^2)^T(v_0^* + v_1\tilde{u} + v_2\tilde{u}^2) + \lambda_u \tilde{u}^T\tilde{u}$$

with $v_0^* = v_0 - y_c$ (9)

With constraints, the cost function can be minimized numerically by a one-dimensional search algorithm (dynamic algorithm programming). Without constraints the solution leads to a third-degree one-dimensional equation [4,6,7].

3. Differential Flatness

The main feature of differential flatness is the presence of a fictitious output *z* (called a flat or linearizing output) such that [5, 8]:

- Every system variable may be expressed as a function of the components of *z* and of a finite number of its time derivatives.
- *z* may be expressed as a function of the system variables and of a finite number of their time derivatives;
- The number of independent components of *z* is equal to the value of independent input elements.

Roughly speaking, a system is said to be differentially flat if all of the feasible trajectories for the system can

be written as functions of a flat output $z(\cdot)$ and its derivatives.
More precisely as given, a nonlinear system:

$$x_{k+1} = f(x_k, u_k) \text{ such as } x_k \in \Re^n, u_k \in \Re^m \quad (10)$$

We say a system is *differentially flat* if there exist an application:

$$h : \Re^n \times (\Re^m)^{r+1} \mapsto \Re^m \ / \ z_k = h(x_k, u_k, u_{k+1}, \ldots u_{k+r})$$

Such that all feasible solutions of the underdetermined differential equation (10) can be written as:

$$\Phi : (\Re^m)^{\ell} \mapsto \Re^n \ / \ x_k = \Phi(z_k, z_{k+1}, \ldots u_{k+\ell})$$

$$\Psi : (\Re^m)^{\ell+1} \mapsto \Re^m \ / \ u_k = \Psi(z_k, z_{k+1}, \ldots u_{k+\ell+1}) \quad (11)$$

Here it's clear that the system trajectory is related through the flat output trajectory, so as usual with flat systems, it suffices to control the flat output, to control the whole system.
After a modelization phases, we need to linearized the system. The main idea is to express the future flat output z_{k+i} *with* $i \in [0, r]$ according to the system states and until has made appear the first combining of control. We notes n=*i*, the number of discrete derivation. $v_k = z_{k+n}$.
Thus as, a new control of an innovative linear system: $z_{k+1} = Az_k + Bv_k$, which is considered a controllable form.
Another consequence of (11) is that, if one imposes $v_k = z_{k+n}$, this last system is equivalent to the system (10) to the following sense: all trajectory of this last system, controlled by *v*, is the image of a trajectory of (11), controlled by *u*, and vice versa, all trajectory of (11) is the image of a trajectory of the linear system controllable under the shape canonical $v_k = z_{k+n}$.

4. Open loop trajectory generation

The goal of the flatness property uses into a predictive control algorithm is to bring the system from a stationary set point $z_k^a = h(x_k^a, u_k^a, u_{k+1}^a, \ldots u_{k+r}^a)$ to another stationary set point $z_k^b = h(x_k^b, u_k^b, u_{k+1}^b, \ldots u_{k+r}^b)$ under a time interval [0; T].
Indeed, the reference trajectory $y^d(t)$ is chosen to be a polynomial between the two set points (a) and (b), according to a desired dynamics. This reference trajectory to follow by the process output is deduced from the desired flat output z_k^d, were z_k^d are the samples of $z^d(t)$ which is polynomial of class C^4 given as follow [5]:

$$\sigma(t) = \begin{cases} 0 & if \quad t < 0 \\ 35t^4 - 84t^5 + 70t^6 - 20t^7 & if \ 0 \le t \le 1 \\ 1 & if \quad t > 1 \end{cases} \quad (12)$$

z_k^d is defined as constant between the moment $(0, t_0)$ and (t_{f1}, t_{f2}) and for $t > t_{f3}$. Otherwise, $z^d(t)$ is a polynomial which is given by the following expression for i=1,3:

$$z^d(t) = z^d(t_{f_{i-1}}) + (z^d(t_{f_i}) - z(t_{f_{i-1}}))\sigma\left(\frac{t - t_{f_{i-1}}}{t_{f_i} - t_{f_{i-1}}}\right) \quad (13)$$

Finally, from point of view of flatness on the predictive control in terms of tracking, we consider that the reference trajectory can be replaced by the flat states y_k^d, defined as fellow:

$$y_k^d = \Phi(z_k^d, z_{k+1}^d, \ldots u_{k+\ell}^d). \quad (14)$$

5. Simulation results

The chosen example used in aim to valid the theory exposed above is given [9]. A continuous state space representation of this example is as follow:

$$\begin{cases} \dot{x}_1 = x_3 - x_2 u \\ \dot{x}_2 = -x_2 + u \\ \dot{x}_3 = x_2 - x_1 + 2x_2(u - x_2) \end{cases}$$

The flat output is given as fellow: $z = x_1 + \dfrac{x_2^2}{2}$
The control u appears in the 3rd derivative, we note then that the new control noted $v = z^{(3)}$.
It comes then:

$$\begin{pmatrix} \dot{z} \\ \ddot{z} \\ \dddot{z} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} z \\ \dot{z} \\ \ddot{z} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} v$$

The output is defined as : $y = [0 \ 0 \ 1]x$
The different solutions of the presented system, according to the flat output z are:

$$\begin{cases} x_1 = -(1 + \ddot{z}) + \sqrt{1 + 2(z + \ddot{z})} \\ x_2 = \ddot{z} + x_1 \\ x_3 = \dot{z} + \ddot{z}^2 + 2x_1\ddot{z} + x_1^2 \end{cases}$$

The system model is implanted in the Matlab-simulink environment of which the goal is to get the input/output vector for the identification phase.

Matlab® discrete these equations by the 4th order Runge-Kutta method.
The vector characterizing the model that linking the output x_3 with the input u is given by:

$$A = \begin{bmatrix} 1 & -1.9897 & 0.9997 \end{bmatrix}^T;$$

$$B_1 = \begin{bmatrix} -0.0318 & -0.0096 \end{bmatrix};$$

$$B_2 = \begin{bmatrix} 0.0396 & 0.0656 & 0 \\ 0 & 0.0388 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

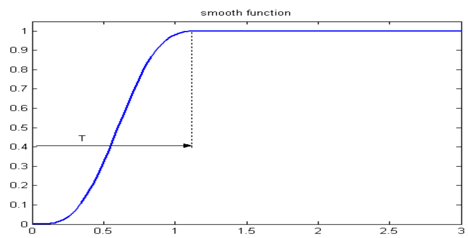


Fig.1: Smooth function $\sigma(t)$ generated trajectory in open loop

The implantation of the constrained model predictive control algorithm NLMPC, shows good performance in term of tracking and regulation. But a meaningful overshoot appears (15%), generated by the presence of hard constraints on the system input (fig.2). This problem can be anticipated by increasing of the prediction horizon. However this solution has the disadvantage to increase the time of calculation. We can overcome this last problem by combining differential flatness and predictive control strategy.

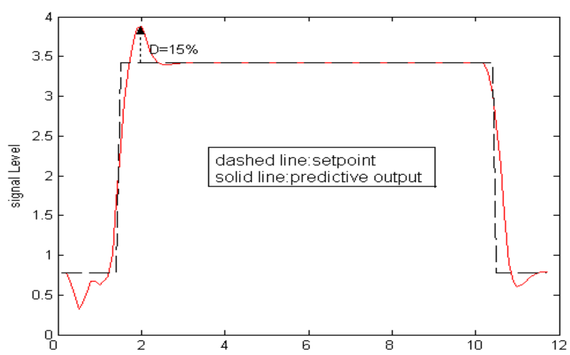


Fig2.a

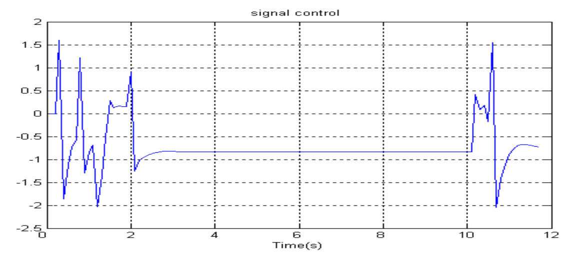
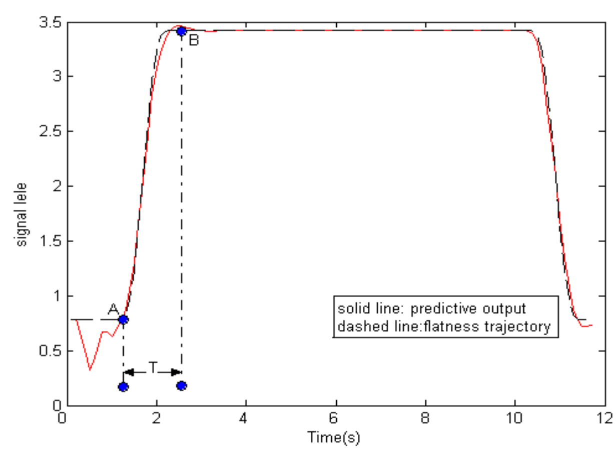


fig2.2

Fig.2: Predictive control responses and control signal for $hp = 3, hc = 1$ and $-0.2 \le \Delta u \le 0.2$

The conception of a reference trajectory (fig.3) permits to avoid an exciting undesirable mode, such an oscillating one. Since the system is anti-causal the trajectory can be obtained by a pure derivation. Besides, the disruptions managed by the closed loop are lower, that permits adjusting of the regulation parameters easier and less sensitive, without damaging the robustness.
Our goal is to displace the system from a starting point A verse one point B (fig.3) during a transition time T, while keeping the regulator parameters. Then we are going to replace the reference step function by a sufficiently regular trajectory, deducted out of the property of flatness (in open loop). However, an exponential convergence is gotten without overtaking on an adjustable period T. the consequently, we obtain an economic substantial energy and a less fatigue of system.



$hp = 3, hc = 1$ and $T = 1.2\text{s}$

Fig3.a

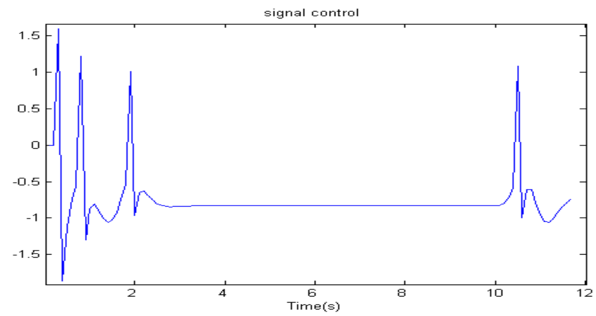


Fig3.2

Fig.3: Closed loop responses with flatness trajectory generated

6. Conclusion

The main contribution of this paper is to combine non linear model predictive control strategy with Flatness theory to control nonlinear process. It should also underline, that the computed control signal can very well be certified by numerical experiments.

The choice of the reference trajectory is directly deduced from the desired flat output. By this choice, the problem due to the presence of non-stable zeros can be prevailing over.

References:

[1] M. J. van Nieuwstadt and R. M. Murray. "Real time trajectory generation for differentially flat systems". *International Journal of Robust and Nonlinear Control*, 8(11):995–1020, 1998.

[2] F.Rotella, F.J.Carrillo and M.Ayadi, "Digital flatness- based robust controller applied to a thermal process". *IEEE international conference on control applications*, pp. 936-941, 2001.

[3] Matthew J. Tenny_and James B. Rawlings TWMCC (Texas-Wisconsin Modeling and Control Consortium) Technical report number 2002-04

[4] N.,Ben Nasr, F.,M'sahli, et R., Ben Abdennour, "NonLinear Model Predictive Control of an Estrification Semi Batch Reactor Using Second order Volterra Models" , *WSEAS TRANSACTIONS on SYSTEMS, Issue 1, Volume 3* January 2004, pp. 265-272, Malta.

[5] M. Fliess, J. L´evine, P. Martin and P. Rouchon, "Flatness and defect in nonlinear systems: Introducing theory and applications", *International Journal of Control*, Vol.61, pp.1327-1361, 1995.

[6] R.Haber, R.Bars, L.Engel, " Three Extended Horizon Adaptive Nonlinear Predictive Control Schemes on the Parametric Volterra Model", *ECC*.Germany, 1999

[7] F.J.Doyle, B.A.Ogunnike and R.K.Pearson, "Non-linear model-based control using seconds order Volterra models". *Automatica*.1995,Vol5, N31.pp.697-714.

[8] J.Levine. "Systèmes plats: mode d'emploi". *CIFA* , Tunisia, 2004

[9] B.Laroche, M. Philippe and P. Rochon. "Motion planning for the heat equation", *international journal of robotic and nonlinear control*. pp. 629-643. 2000.

[10] F. M'sahli, R.B.abdennour, and M.Ksouri, "Identification and Predictive Control of Nonlinear Process Using a Parametric Volterra Model", *International Journal of Computational Engineering Science*, Vol 2, No 4, 2001, pp. 633-651.

[11] G.R.Sriniwas and Y.Arkun, "A global Solution to the Nonlinear Model Predictive Control Algorithms Using Polynomial ARX Models". *Computers chem. Engng*, 1997, Vol.21, No 4 pp.431-439.