In this paper, a theoretical background is presented for the stabilization of non linear systems. A numerical implementation is then proposed. The class of systems concerned with the proposed practical approach is quite large and contains all flat systems as a particular subset. The stabilizing strategy is based on path generation strategy and avoids the integration of the differential system. It was implemented by an extensive use of interpolation functional basis. Two examples of systems known to be hard to stabilize are given to illustrate the proposed algorithm.

1 Introduction

The stabilization of nonlinear systems remains a quite difficult issue as long as no special structure is assumed to hold for the nonlinear system to be stabilized.

One of the most general control strategies that enables - at least conceptually - to stabilize a very wide class of nonlinear systems is the receding horizon control [9, 10, 6, 1]. Indeed, roughly speaking, the only assumptions needed for this strategy to apply is the stabilizability of the system and some classical regularity assumptions. Unfortunately, for general nonlinear systems, the computation of the receding horizon control law implies the minimization at each sampling time of a global non convex open-loop cost function with final state constraint. Only the "first part" of the corresponding open-loop optimal control is applied and the whole procedure is reiterated at the next sampling time to obtain a state feedback.

The use of repeated open-loop computations with eventually a revising feedback [14], in order to formulate a state feedback that underlines the receding horizon strategy, has also been used in the framework of differentially flat systems [3, 2]. Indeed, in this context, the open-loop trajectories of the states are parameterized by the trajectory of what is called the flat output. This enables to generate open-loop trajectories satisfying stable asymptotic behavior while avoiding integration of system equations. The state feedback is then constructed in order to control the dynamic of the tracking error with respect to this a-priori defined open-loop trajectory. Steering a system from an initial state to a final one after some time \( T \) has already been studied in the case of chained form systems [11, 12, 13]. Some methods using sinusoidal, piecewise constant or polynomial inputs can be found in the literature. The use of a certain class of functions to describe the generated trajectory (with smooth properties) has also been used for robot path generation [5].

In this paper, we combine the receding horizon approach with path generation that avoid integration of the system dynamic in order to construct a stabilizing feedback.

It is worth noting that, in classical receding horizon scheme, even when one is concerned only with stabilization - no tracking is needed - the attractibility nature of the origin is insured by the minimization procedure. Indeed, the corresponding Lyapunov function is just the optimal cost, therefore, the repeated optimization insures its decrease. Here, we propose a slightly different strategy in order to avoid unnecessary heavy computations related to optimization when only stabilization is addressed.

The open loop-problem that is systematically solved is that of finding open-loop control that steers the system to the origin. The decreasing property associated with cost minimization in classical receding horizon control is replaced by a particular initialization of the procedure that searches for such admissible open-loop control. This particular initialization uses the result of the last past time search to feed the search procedure with an a-priori "decreasing" candidate open-loop control.

The paper is organized as follows, first of all, the aim of the paper is clearly defined in section 2, then, some
3 Definitions and Notations

Assumption 1

∈ X ⊂ I R

where for all vector function

dk v

dk :=  d k−1 v

d k−2 v

d 0 v
 ∈ I R

Consider the general nonlinear system given by:

f(x, dx/dt, u, du/dt, ..., dn−1u/dtn−1) = 0

where for all vector function v(t) ∈ I R:

dk v

dk :=  d k−1 v

d k−2 v

d 0 v
 ∈ I R

x ∈ X ⊂ I R is the state of the system, u ∈ I R is the control input and X is a region of interest.

f : I R2n+m → I Rm such that f(0) = 0. We shall denote by x(t; t0, x0, u(.)) the solution at instant t of (1) when it exists under the control u(.) with initial conditions x0 at instant t0. Furthermore, we suppose that the dynamic system (1) satisfies the following assumption:

Assumption 1 There exists a finite time t f min such that, for all initial state x0 ∈ X, there exists an open loop control strategy v(.) defined over [0, t f min] that steers the state of the system to 0, namely x(t f min; 0, x0, v(.)) = 0.

Let T > 0 be a sampling time, the aim of this work is to find an implementable state feedback u(t + nT) = K(x(nT), t) that globally stabilizes the dynamic system (1) at 0. In all the paper, we shall take a fixed t f ≥ t f min.

2 Problem Formulation

Consider the general nonlinear system given by:

f(x, dx/dt, u, du/dt, ..., dn−1u/dtn−1) = 0

where for all vector function v(t) ∈ I R:

dk v

dk :=  d k−1 v

d k−2 v

d 0 v
 ∈ I R

x ∈ X ⊂ I R is the state of the system, u ∈ I R is the control input and X is a region of interest.

f : I R2n+m → I Rm such that f(0) = 0. We shall denote by x(t; t0, x0, u(.)) the solution at instant t of (1) when it exists under the control u(.) with initial conditions x0 at instant t0. Furthermore, we suppose that the dynamic system (1) satisfies the following assumption:

Assumption 1 There exists a finite time t f min such that, for all initial state x0 ∈ X, there exists an open loop control strategy v(.) defined over [0, t f min] that steers the state of the system to 0, namely x(t f min; 0, x0, v(.)) = 0.

Let T > 0 be a sampling time, the aim of this work is to find an implementable state feedback u(t + nT) = K(x(nT), t) that globally stabilizes the dynamic system (1) at 0. In all the paper, we shall take a fixed t f ≥ t f min.

3 Definitions and Notations

• For the ease of the reader, we shall denote (B)A := {f(.) : A → B}

• Assumption 1 enables us to define for all state x0 and all t f ≥ t f min, A(x0, t f) as the set of all control strategies that steers the state to 0 at t f:

A(x0, t f) := \{u(.) ∈ (I Rm)0,t f s.t. x(t f; 0, x0, u) = 0 \}

In the foregoing, a fixed value t f will be considered, hence A(x0, t f) will simply be denoted by A(x0).

• Let us define a time translation function S : (I Rq)0,t f × [0, t f] → (I Rq)0,t f by:

S(u, D)(t) = \{ u(t + D) for t ∈ [0, t f − D[ 0 for t ∈ [t f − D, t f]

(3)

• For all time function z(t), we shall denote by:

F(\tilde{z}(t)) = 0

(4)

a general relation between z(t) and a finite number of its derivatives at instant t. For example, according to this notation, equation (1) can be formally written as follows:

\tilde{f}(\tilde{z}(t), \tilde{u}(t)) = 0
• For all $\eta = (\eta_1, \eta_2, \ldots, \eta_q) \in A_1 \times A_2 \times \ldots \times A_q$, we shall note $\eta_i = \pi_i(\eta)$. Furthermore, for $J = \{i_1, \ldots, i_k\}$ where $i_1, \ldots, i_k \in [1, q]$, we denote:

$$
\pi_J(\eta) = \begin{pmatrix}
\eta_{i_1} \\
\vdots \\
\eta_{i_k}
\end{pmatrix}
$$

(5)

4 Theoretical Background

In this section, the theoretical background that underlies the proposed approach is presented. First, a basic version is studied in section 4.1 while in section 4.2, a slightly improved version (w.r.t unmeasured perturbations) is suggested. Strictly speaking, the proposed approach leads to a state feedback that renders the origin attractive which does not guarantee a Lyapunov-like boundness of the transient behavior. Section 4.3 briefly discusses this feature.

4.1 Basic version

Suppose in a quite abstract manner that there is a set $P$ of parameters that enables us to describe the elements of $A := \bigcup_{x_0 \in X} A(x_0)$ (set of admissible open loop controls when $x_0$ spans $X$) such that the following mappings exist:

$$
U : X \times P \to X \times (\mathbb{R}^m_{[0, t_f]})
$$

(6)

$$
R : X \times A \to X \times P
$$

(7)

and are such that:

$$
U \circ R = Id_{X \times A}
$$

(8)

Namely, $U$ computes the control strategy over $[0, t_f]$ from the knowledge of an initial state and a control parameterization, while $R$ computes a parameterization of a given control strategy that steers the state from a given initial state to 0. Note also that $U$ and $R$ keep unchanged the initial condition ($\in X$) passed as first argument.

Suppose in addition that one has a systematic procedure that computes the parameterization of an admissible control strategy in $A(x_0)$, namely:

$$
Q : X \times P \to X \times P \text{ such that } U(Q(x, p)) \in \{x\} \times A(x) \quad \forall (x, p) \in X \times P
$$

(9)

In other words, $Q$ is a systematic procedure that finds an open-loop control strategy that steers the state from $x$ to 0 during the time $t_f$ (be definition of $A(x)$). The variable $p$ is an initial guess for the systematic procedure $Q$ which may be an iterative routine. Note that $Q$ also keeps unchanged the first argument ($\in X$).

We shall suppose that $Q$ satisfies the following assumption:

**Assumption 2** For all $x \in X$ and all $p \in P$:

$$
\left\{ U(x, p) \in \{x\} \times A(x) \right\} \Rightarrow \left\{ Q(x, p) = (x, p) \right\}
$$

(10)

In other word, if one initializes the procedure $Q$ with a "good" guess, $Q$ terminates and gives this initial guess as a solution.

This property of function $Q$ is the key feature that ensures the attractiveness of the origin in the proposed scheme.

We have the following theorem:

**Theorem 4.1** Let $T > 0$ be a sampling time. In the absence of disturbances and under assumptions 1-2, the feedback law:

$$
u(nT + t) = K_u(t) \quad 0 \leq t < T
$$

(11)
where:
\[ K_n := \pi_2 \left( U \circ Q(x(nT), p_n) \right) \in \mathbb{R}^m \] (12)
\[ p_n := \pi_2 \left( R(x(nT), S(K_{n-1}, T)) \right) \in \mathcal{P} \quad n \geq 1 \] (13)
\[ p_0 \] is arbitrarily chosen in \( \mathcal{P} \) (14)

exists and steers the system to the origin after a finite time \( t_f \).

The crucial rule of assumption 2 should be emphasized since it ensures the attractiveness of the origin in the proposed scheme. Thanks to it, if one initializes the procedure \( Q \) with a good parameterization \( p \) (in the sense that the state will vanish after the time \( t_f \)), this strategy will be kept and hence the state of the system will reach the origin after time \( t_f \). This would not be true anymore if one changes strategies at each sampling time.

**Proof**

The existence of the feedback law is obvious to the extent that, from assumption 1, \( \mathcal{A}(x) \neq \emptyset \). Given assumption 2, (9) and (12) imply:
\[ K_n \in \mathcal{A}(x(nT)) \quad \forall n \geq 0 \] (15)

Furthermore, it is clear from definition of \( S \), that \( f(0) = 0 \) and the assumption of a free perturbation evolution gives:
\[ \left\{ K \in \mathcal{A}(\xi) \right\} \Rightarrow \left\{ S(K, T) \in \mathcal{A}(x(T; 0; \xi; K)) \right\} \] (16)

Hence:
\[ (15) \Rightarrow \left\{ K_{n-1} \in \mathcal{A}(x((n-1)T)) \right\} \xrightarrow{(10)} \left\{ S(K_{n-1}, T) \in \mathcal{A}(x(nT)) \right\} \quad \forall n \geq 1 \] (17)

(13) yields:
\[ \left( x(nT), p_n \right) = R\left( x(nT), S(K_{n-1}, T) \right) \] (18)

applying \( U \) to both sides of (18) and using (8) and (17) gives:
\[ U\left( x(nT), p_n \right) = \left( x(nT), S(K_{n-1}, T) \right) \in \{ x(nT) \} \times \mathcal{A}(x(nT)) \] (19)

This implies according to (10):
\[ Q(x(nT), p_n) = (x(nT), p_n) \] (20)

(19), (20) and (12) implies:
\[ K_n = S(K_{n-1}, T) \] (21)

We have by definition of the feedback law:
\[ u(t + nT) = K_n(t) \]
\[ = K_{n-1}(t) \quad \text{according to (21)} \]
\[ = K_{n-2}(t + 2T) = \ldots = K_0(t + nT) \] (22)

and therefore \( u(t) = K_0(t) \). Furthermore, \( K_0 \) is an admissible open-loop control that steers the state to the origin during the time \( t_f \) since it belongs to \( \mathcal{A}(x(0)) \). (22) proves that the actual input resulting from the feedback is exactly \( K_0 \) which ends the proof.

□
4.2 Disturbances handling

The feedback law (11)-(14) may behave dangerously in the presence of disturbances. Indeed, according to this feedback, during the time interval \([nT, (n+1)T]\) we apply an open-loop control depending on the state \(x(nT)\). Therefore, if a ”bad” perturbation arises at a time \(t^*\) in the interval \([nT, (n+1)T]\) such that the corresponding autonomous evolution has an escape time lower than \((n+1)T - t^*\) the state may diverge to \(\infty\) even before the application of control at the next sampling period.

Note that the above problem is a common problem that arises each time a continuous feedback law is implemented numerically. Indeed, a numerically applied feedback is always piecewise constant, therefore, during sampling time intervals, one always has an open-loop control.

The reason why we shall handle this problem explicitly while it is commonly neglected lies in the fact that the computation of the feedback law (11)-(14) may need a time which cannot by indefinitely reduced.

We shall define a ”bad perturbation” indicator as follows. Let \(p_0 := \pi_2(Q(x_0, p^*))\) be an admissible control parameterization that corresponds to a perturbation free closed-loop evolution according to theorem 4.1. Define:

\[
M(x_0, p_0) := \sup_{0 \leq t \leq T} \| x(t; 0; x_0, U(x_0, p_0)) \| \tag{23}
\]

\(M(x_0, p_0)\) is clearly the minimum radius of a ball in \(\mathbb{R}^n\) that contains the perturbation free closed-loop trajectory starting from \(x_0\) and applying the control corresponding to the initial admissible parameterization \(p_0\).

The definition of \(M(x_0, p_0)\) enables us to define a ”bad perturbation” indicator by:

\[
e(nT + t) := \| x(nT + t) \| - \lambda M(x(nT), p_n) \quad 0 \leq t \leq T \tag{24}
\]

\(\lambda > 1\) is a given fixed security margin.

Note that, according to the definition of \(M(x(nT), p_n)\), during a perturbation free evolution we have:

\[
e(nT + t) < (1 - \lambda)M(x(nT), p_n) < (1 - \lambda) \| x(nT) \| \tag{25}
\]

A ”bad perturbation” is therefore defined with respect to a fixed security margin \(\lambda > 1\). It causes the state trajectory to leave the ball \(B(0, \lambda M(x(nT), p_n)) \subset \mathbb{R}^n\) at an instant \(t \in [nT, (n+1)T]\), at the same instant, \(e\) changes from negative to positive.

When this happens, the ”escape”-assumption is considered to be sufficiently plausible to reexamine the strategy defined at instant \(nT\) to be applied over the interval \([t, (n+1)T]\). Therefore, instant \(t \in [nT, (n+1)T]\) becomes a new ”decision instant”. By decision instant \(t\), we mean an instant where a new admissible open-loop strategy is recomputed starting from \(x(t)\) to be applied during the interval \([t, (n+1)T]\).

A consequence of this strategy is that decision instants are no more necessarily of the form \(nT\). That is why we define the following time variable ”last decision instant” \(D\) as follows:

\[
D := \max \left\{ E \left( \frac{t}{T} \right), T, \bar{t} \right\} \tag{26}
\]

where \(E(t\over T)\) is the integer part of \(\frac{t}{T}\) and \(\bar{t}\) is the last instant the indicator \(e(t)\) changed from negative to positive with \(e(t)\) redefined as follows:

\[
e(D + t) := \| x(D + t) \| - \lambda M(x(D), p(D)) \quad 0 \leq t < T \tag{27}
\]

where \(p(D)\) is the last computed control parameterization. \(\bar{t}\) being initialized to 0. Note that the initial value of \(D\) is clearly equal to 0 and equations (26)-(27) enable to correctly define \(D\). Note also that we always have \(t - D \leq T\).

**Remark 4.1** (23) can be defined separately for each component of the state \(x\), which enables to detect ”bad” perturbation more accurately.

Now, we have all needed to formulate the stabilizing feedback law:
Theorem 4.2 Let $D$ be defined by (26)-(27) and $D^*$ denote the value of $D$ before its last change (remember that $D$ changes in a discontinuous manner).

The feedback law defined by:

$$u(D + t) = K_D(t) ; \quad 0 \leq t < T$$

where:

$$K_D := \pi_2 \left( U \circ Q(x(D), p(D)) \right)$$

$$p(D) := \begin{cases} \pi_2 \left( R(x(D), S(K_D^*, D - D^*)) \right) & \text{if } D > 0 \\ \text{Arbitrarily chosen } \in \mathcal{P} & \text{if } D = 0 \end{cases}$$

globally asymptotically stabilizes (1).

Proof: Note first that if there are no perturbations, the feedback law (28)-(30) is exactly the same than that of theorem 4.1 since under this assumption one has $D = uT$.

Therefore, the proof is straightforward according to the above discussion and the result of theorem 4.1. Indeed, by introducing the indicator $e$ and the non uniform decision instants through $D$, we avoid the perturbation-caused "explosion" of the state. The "steering" property of the feedback law (theorem 4.1) enables to conclude.

It is worth noting that at $t = 0$ one has $D = D^* = 0$ and the initial parameterization for the procedure is arbitrarily chosen in $\mathcal{P}$ so that (29)-(30) properly define $K_D$.

4.3 About Lyapunov stability

The feedback strategy proposed in this section can be summarized as follows: in the absence of disturbances, the closed-loop control is chosen so that the closed-loop path is identical to the open-loop path. This simple choice is sufficient to insure the attractiveness of the origin. With the present assumptions, nothing prevents the open-loop controls from steering the state to the origin with large excursions.

In [7], the so called "admissible open-loop path generators" are used to design an asymptotically stabilizing state feedback (in the Lyapunov sense). Beside some regularity assumptions, the above path generators are assumed to satisfy a somehow transitivity assumption (see definition 2.1. of [7]) that can be closely related to assumption 2 mentioned above. Under these conditions, the state trajectory over an infinite time interval can be used to construct a Lyapunov function. Furthermore, in the absence of disturbances, open-loop and closed-loop trajectories coincide again.

5 Numerical Implementation

In the preceding section, we have presented an abstract theoretical framework that leads to a stabilizing feedback for general nonlinear systems. The aim of the present section is to propose a corresponding practical numerical implementation that holds for a particular class -although wide- of nonlinear systems including systems that are not state feedback linearizable (see examples below). For these systems, the proposed method leads to a systematic and parameterized path planning algorithm.

The class of systems showing of particular structural properties and concerned by the implementation is presented in the first subsection. Examples of such systems are then given, emphasizing the wideness of the proposed class. The principle of the method is then described.

5.1 Definition of systems under consideration

Let us consider the system defined by (1). Note that it can be written in an abstract manner $f(\tilde{z}(t)) = 0$.

The following property characterizes the class of systems concerned by the implementation proposed in the remainder of the paper:

Definition 5.1 The set of equations:

$$f(\tilde{z}(t)) = 0$$

is said to be in normal form if there exist:
1. a subdivision of $z = \left( \begin{array}{c} z_f \\ z_D \end{array} \right)$ with, $z_f \in \mathbb{R}^{n_f}, \ z_D \in \mathbb{R}^{n_D}$ and $n_f \neq 0$ such that the constraint (31) is compatible with a completely free a-priori choice of $z_f(\cdot)$ as a time function (as far as the initial constraints are respected).

2. $n_D$ relations $[F_i]_{1 \leq i \leq n_D}$ such that the set of constraints (31) is equivalent to

$$F_i(\hat{z}_f(t), \hat{z}_{D_1}(t), \hat{z}_{D_2}(t), \ldots, \hat{z}_{D_i}(t)) = 0 \quad 1 \leq i \leq n_D \tag{32}$$

Such a normal form shows of particular triangular properties of the system and consequently allows an to compute successively $z_{D_1}, z_{D_2}, \ldots, z_{D_{n_D}}$. Furthermore, as the foregoing examples will underline, particular forms of (32) can simplify the calculation (e.g. when (32) is linear in $\hat{z}_{D_i}$, rendering the search of an open-loop trajectory easier).

### 5.2 Examples

In this section, four illustrative examples are exposed. The first two examples concern classes of systems that are well known in the nonlinear control literature, namely the flat systems and the chained form systems. The last two examples are typically non flat systems (see [3, 2]). In [8], the above strategy has been applied to the rigid spacecraft in failure mode. The robustness w.r.t parameters uncertainties has been rather successfully tested although the problem is known to be hard to solve even in an uncertainty free context.

#### 5.2.1 Differentially flat systems

All differentially flat systems [3, 2] can be transformed into the normal form in the sense of definition 5.1. Indeed, when using our notations, the system $f(\bar{x}(t), \bar{u}(t))$ is differentially flat if one can find $y = h(\bar{x}, \bar{u})$ such that there exist two functions $G$ and $H$ satisfying:

$$x = G(\bar{y}) \quad u = H(\bar{y})$$

Now, if we denote:

$$\tilde{z} := \left( \begin{array}{c} y \\ x \\ u \end{array} \right) \quad \tilde{f}(\tilde{z}) = \left( \begin{array}{c} y - h(\bar{x}, \bar{u}) \\ x - G(\bar{y}) \\ u - H(\bar{y}) \end{array} \right)$$

then, it is straightforward that $\tilde{f}(\tilde{z}) = 0$ is in normal form with the following choices:

$$z_f := y \quad z_D := \left( \begin{array}{c} x \\ u \end{array} \right)$$

Furthermore, the relations $F_i = 0$ become trivial algebraic equalities of the form (33) rather than time-varying differential equations.

$$F_i(\tilde{z}_f, \tilde{z}_{D_i}) = G_i(\tilde{z}_f) - z_{D_i} = 0 \tag{33}$$

#### 5.2.2 Chained form systems

Given a general chained form system (see [13]) with $m + 1$ inputs, $m(m + 1)$ chains and $m + 1$ generators:

$$\begin{align*}
\dot{x}_j^0 &= v_j & 0 \leq j \leq m \\
\dot{x}_{ji}^1 &= x_j^0 v_i & j > i & \text{and} & x_{ij}^1 &= x_i^0 x_j^0 - x_{ji}^1 \\
\dot{x}_{ji}^k &= x_{ji}^{k-1} v_i & 1 \leq k \leq n_j & \text{and} & 0 \leq j, i \leq m & j \neq i
\end{align*} \tag{34}$$

by choosing $z_f := (v_0, \ldots, v_m)$, the system (34) is already in the normal form in the sense of definition 5.1, and we have:

$$z_D = \left( x_0^0, \ldots, x_m^0, x_0^1, \ldots, x_m^1, \ldots, x_0^{n_j}, \ldots, x_m^{n_j} \right)$$

It is then obvious that the relations $F_i$ can be deduced from (34).
5.2.3 Inverted pendulum with horizontal action

Consider the equations of motion of the inverted pendulum:

\[ \begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \frac{1}{L(1 - a \cos^2 x_1)} \left( -aL \sin x_1 \cos x_1 x_2^2 + g \sin x_1 - \frac{a}{m} \cos x_1 u \right) \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= \frac{1}{L(1 - a \cos^2 x_1)}(aLx_2^2 \sin x_1 - ag \sin x_1 \cos x_1 + \frac{a}{m} u)
\end{align*} \]

That we shall write for convenience as follows:

\[ \begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \Phi_2(x_1, x_2) + \Psi_2(x_1)u \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= \Phi_4(x_1, x_2) + \Psi_4(x_1)u
\end{align*} \] (35)

In these equations \( x := [\theta, \dot{\theta}, r, \dot{r}]^T \) and \( u \) is the horizontal force applied on the car. It is well known that the inverted pendulum with the only horizontal force as control action is not a flat system. System (35) is already in the normal form with for example:

\[ z_f := x_1 \quad ; \quad z_D := (x_2 \quad u \quad x_4 \quad x_3)^T \]

and

\[ \begin{align*}
F_1(\tilde{z}_f, \tilde{z}_{D_1}) &:= \tilde{z}_f - z_{D_1} \\
F_2(\tilde{z}_f, \tilde{z}_{D_1}, \tilde{z}_{D_2}) &:= \tilde{z}_{D_1} - \Phi_2(z_f, z_{D_1}) - \Psi_2(z_f)z_{D_2} \\
F_3(\tilde{z}_f, \tilde{z}_{D_1}, \tilde{z}_{D_2}, \tilde{z}_{D_3}) &:= \tilde{z}_{D_2} - \Phi_4(z_f, z_{D_1}) - \Psi_4(z_f)z_{D_2} \\
F_4(\tilde{z}_f, \tilde{z}_{D_1}, \tilde{z}_{D_2}, \tilde{z}_{D_3}, \tilde{z}_{D_4}) &:= \tilde{z}_{D_3} - z_{D_3}
\end{align*} \] (36-39)

5.2.4 The ball and beam

This system studied in [3, 4] is also known to be a non flat system. After a trivial static feedback, the system can be given by:

\[ \begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -Bg \sin(x_3) + Bx_4^2 x_1 \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= w
\end{align*} \] (40)

where \( x := [r, \dot{r}, \theta, \dot{\theta}]^T \), \( r \) and \( \theta \) being respectively the position of the ball on the beam and the inclination of the beam from horizontal. Equations (40) are already in normal form with:

\[ z_f := w \quad ; \quad z_D := (x_4 \quad x_3 \quad x_1 \quad x_2)^T \]

and:

\[ \begin{align*}
F_1(\tilde{z}_f, \tilde{z}_{D_1}) &:= \tilde{z}_{D_1} - z_f \\
F_2(\tilde{z}_f, \tilde{z}_{D_1}, \tilde{z}_{D_2}) &:= \tilde{z}_{D_2} - z_{D_1} \\
F_3(\tilde{z}_f, \tilde{z}_{D_1}, \tilde{z}_{D_2}, \tilde{z}_{D_3}) &:= \tilde{z}_{D_3}^{(2)} + Bg \sin(z_{D_2}) - Bz_{D_2}^2 z_{D_3} \\
F_4(\tilde{z}_f, \tilde{z}_{D_1}, \tilde{z}_{D_2}, \tilde{z}_{D_3}, \tilde{z}_{D_4}) &:= \tilde{z}_{D_4} - z_{D_4}
\end{align*} \] (41-44)

5.3 Principle of the implementation

The numerical implementation is based on the extensive use of a functional basis defined on the interval \([0, t_f]\). In this paper, a Chebyshev polynomial basis is considered:

\[ T_0(t) := 1 \] (45)
\[T_1(t) := \frac{2t}{t_f} - 1\] (46)
\[T_n(t) := 2 \left( \frac{2t}{t_f} - 1 \right) T_{n-1}(t) - T_{n-2}(t)\] (47)

We shall also denote by \(T(t, q, N)\) the \(\mathbb{R}^{n \times Nq}\) matrix defined by:
\[T(t, 1, N) := [T_1(t), \ldots, T_N(t)]\] (48)
\[T(t, q, N) := \left( \begin{array}{ccc}
T(t, 1, N) & 0_{1 \times N} & \cdots & 0_{1 \times N} \\
0_{1 \times N} & T(t, 1, N) & 0_{1 \times N} & \cdots \\
\vdots & \vdots & \ddots & \vdots \\
0_{1 \times N} & \cdots & 0_{1 \times N} & T(t, 1, N)
\end{array} \right) \text{ for } q \geq 2\] (49)

\(T(t, q, N)\) enables clearly to write the approximation of a time function \(v(t)\) on \(\mathbb{R}^q\) in a compact manner as:
\[v(t) \approx [T(t, q, N)] a \quad a \in \mathbb{R}^{Nq}\]

Let us finally define for all \(i \in \mathbb{N}\) and all \(m \in \mathbb{N}, T_i^{(m)}(t)\) to be the \(m\)th derivative of \(T_i(t)\) and \(T_i^{(-m)}(t)\) to be the \(m\)th primitive of \(T_i(t)\) that vanishes at 0. We then naturally define the corresponding \((q \times Nq)\) matrices \(T^{(m)}(t, q, N)\) and \(T^{(-m)}(t, q, N)\) for all \(q \geq 1\) according to (49).

Recall that, to the extent that system (1) can be set in the previous defined normal form, we have the triangular system of differential equations:
\[F_i(\tilde{z}_f(t), \tilde{z}_{D_1}(t), \ldots, \tilde{z}_{D_{l-1}}(t), \tilde{z}_{D_l}(t)) = 0 \quad 1 \leq i \leq n_D\]
with \(\tilde{z}_{D_i} = (z_{D_1}, \frac{dz_{D_1}}{dt}, \ldots, \frac{d^{n_{D_i}}z_{D_1}}{dt^n_{D_i}})\) (51)

Let \(a\) be the vector of parameters that defines the evolution of \(z_f\), i.e.: \(z_f(t) := T(t, n_f, N_f)a \quad a \in \mathbb{R}^{n_f}\) (52)

The key idea is to transform the \(n_D\) relations (50) into an algebraic system of equations in the unknown \(a\) (see equation (54) below), expressing the fact that \(z_{D_i}(t_f) = 0\) for \(i \in [1, n_D]\).

For all \(i \in [1, n_D]\), let \(\alpha_i\) be the coordinates of the best least squares approximation of the solution of differential equation (50), namely:
\[z_{D_i}(t) \approx T(t, 1, N_f)a_i(a)\]
\(\alpha_i\) being the solution of the following least squares problem:
\[\alpha_i(a) = \text{Arg} \left\{ \min_{\alpha_i \in \mathbb{R}^{n_i}} \sum_{j=1}^{N_0} \left\| F_i(\tilde{z}_f(t_j), \ldots, \tilde{T}(t_j, 1, N_f)a_i) \right\|^2 \right\} \] (53)

Furthermore, depending on the nature of \(z_{D_i}\), \(\alpha_i\) may have to respect initial conditions on \(z_{D_i}\).

Writing the final constraint \(z_{D_i}(t_f) = 0\) for \(i \in [1, n_D]\) gives the following triangular system of static nonlinear equations:
\[NLE_i(t_f, x_0, a) := T(t_f, 1, N_D)a_i(a) = 0 \quad i \in [l_D + 1, n_D]\] (54)

which is the nonlinear system of equations in the unknown \(a\) we were looking for. \(l_D\) defined as in the following remark:

**Remark 5.1** The search space can be reduced:

1. By using the initial and final constraints on \(z_f, n_{L_p}\) linear equations can then be obtained.
2. When a set of \(l_D\) equations (50) yields to equations (53) that are linear in \(a\), the initial and final constraints on \(z_{D_i}\) give \(n_{L_D}\) linear constraints on \(a\). The corresponding equation \(NLE_i(t_f, x_0, a) = 0\) is linear and allows to reduce the search space dimension.

Let us define \(n_L := n_{L_f} + n_{L_D}\). The search space dimension can then be reduced from \((n_f, N_f)\) to \((n_f, N_f - n_L)\) by introducing a new unknown vector \(p\) of independent variables. The vector \(a\) will then depend on \(p\). See Appendix A for further details.
5.4 Concrete definition of the abstract subroutines $U$, $R$ and $Q$

In this section, we use the results of the numerical implementation equations in order to define precisely the abstract notions defined in section 5.1, namely, the parameter space $\mathcal{P}$, and the way that one computes for all $x_0 \in \mathcal{X}$, $p_0 \in \mathcal{P}$ and $u_0^f \in (\mathbb{R}^m)_{[0,1]}$ the quantities $U(x_0, p_0)$, $R(x_0, u_0^f)$ and $Q(x_0, p_0)$. This completes the definition of the feedback law proposed in theorem 4.2. We assume that the system under consideration satisfies the assumptions of definition 5.1, with the $\mathcal{F}_i$'s defined by (32).

It becomes clear that the parameters set $\mathcal{P}$ is that to which belongs the vector of unknowns $p$ and therefore $\mathcal{P} := \mathbb{R}^{N_f \times n_f - n_{L_f} - n_{D_f}}$.

Recall that when $z_F(t) = T(t, n_f, N_f) a(p)$, the evolution of $z_D$ can be obtained by solving (54). The corresponding control $u(p_0, t)$ can then be deduced.

The functions $U$ and $R$ are simply given by:

\[
U(x_0, p_0) = (x_0, u(p_0, t)) \quad (55)
\]

\[
R \left( x_0, u_0^f(\cdot) \right) = (x_0, p) \quad (56)
\]

with $p$ being the coordinates of the projection of $z_F(\cdot)$ on the functional basis, when the control $u_0^f(\cdot)$ is applied on the system.

The path search procedure $Q$ is defined as follows:

\[
Q(x_0, p_0) = \left( x_0, \text{solution over } p \text{ of } (NLE_i(t_f, x_0, p) = 0)_{i \leq t \leq n_D} \right) \quad \text{with initial guess } p_0
\]

6 Examples

In this section will be presented the numerical results, and some elements of the implementation. For figure 1. to 3., the legend "Perturbation at $T_{pert} = \tau_{pert}$ $s$ of $X_{pert} = \delta X_{pert}$" means that an additive perturbation of value $\delta X_{pert}, \delta(t - \tau_{pert})$ will append on the states of the system. The "perturbation checking period" is the period used to check if any perturbation occurs on the system between each computation of the command.

6.1 Inverted pendulum

For this example, the initial and final constraints on $z_f$ give $n_{L_f} = 2$. It is clear from (36) that $z_D$ linearly depends on $a$, and therefore the initial and final constraints on $z_D$ give $n_{L_D} = 2$. Furthermore, $z_D(t_f) = 0$ can be taken out from the set of nonlinear equations to be solved. The least squares problem (53) is linear because all the equations (37) to (39) are linear differential equations with respect to $z_D$.

The path planning problem can finally be resumed in solving only three non linear equations in the unknown $p$ with by construction of the method, an initialization very close to a solution.

The figure 1. shows the ability of the method to solve severely nonlinear problems (starting angle of 50°). The simulations were done taking $L = 0.6 m$, $m = 0.125 kg$ and $a = 0.2$.

6.2 Ball and beam

In this example, $z_f$ is the input of the system and therefore only the final constraint can be used to reduce the search space dimension. We have $n_{L_f} = 1$. It is clear from (41) and (42) that $z_D$ and $z_{D_2}$ linearly depend on $a$, and therefore the initial and final constraints on $z_D$ and $z_{D_2}$ give $n_{L_D} = 2$. Furthermore, $z_{D_2}(t_f) = 0$ and $z_{D_2}(t_f) = 0$ can also be taken out from the set of nonlinear equations to be solved. As for the inverted pendulum problem, the least squares problem (53) is linear for all $\alpha_i$.

The path planning problem can be in this case resumed in solving only two non linear equations in the unknown $p$ with an initialization very close to a solution.

See figure 2. for the result with $B = 0.5$.

6.3 Example showing the effect of the perturbation detection

This system was chosen to underline the importance of the perturbation detection as explained in section 4.

The equations of the system are:

\[
\dot{x} = x^2 + u
\]
Let’s choose: \( z_f = x \) and \( z_D = u \)
This system has a finite escape time. Therefore, if a bad perturbation occurs while \( T > \frac{1}{x(kT)} \) (\( k \in \mathbb{N} \)), the system may diverge (See figure 3.).

7 Conclusion

In this paper, a theoretical framework for the stabilization of general nonlinear systems is proposed. Then, a numerical implementation that approximates the theoretical framework is given. Mainly two elements are making the method work powerfully. The first one is structurally held in the method and is the way the procedure \( Q \) is initialised with the previous solution (after a time translation) as defined in section 4. The second lies to the triangular property of the systems, defined in section 5, to which such numerical implementation is possible to apply. This numerical implementation is based on an extensive use of polynomial interpolation. Finally several examples have been given in order to prove the efficiency of the proposed method.

The robustness of the method as well as the problem of real time implementation (although some points were treated) remain to be addressed. However, taking into account that only approximations of the solutions of the differential equations are used (that can be seen as system uncertainties), the stability of the studied systems let us catch sight of good robustness properties.

References


A Appendix

The aim of this section is to explain how the search space dimension can be reduced. An example of such situation can be found in the inverted pendulum example. The initial and final constraints on $z_f$ yield to the first two lines of equation (58). Expressing the initial and final constraints on $z_{D_1} = T^{(1)}(t, n_f, N_f) a$ (which depends linearly on $a$) we obtain the two last lines of equation (58). Furthermore, $z_{D_1}$ can be taken out from the non linear equation to be solved because for any choice of $a$ the final constraint $z_{D_1}(t_f) = 0$ will be satisfied.

$$
\begin{pmatrix}
  T(0, n_f, N_f) \\
  T(t_f, n_f, N_f) \\
  T^{(1)}(0, n_f, N_f) \\
  T^{(1)}(t_f, n_f, N_f)
\end{pmatrix} a =
\begin{pmatrix}
  z_f(0) \\
  0 \\
  z_{D_1}(0) \\
  0
\end{pmatrix}
$$

(58)

More generally, the initial constraints on $z_f$ and eventually special on $z_{D_i}$ when the $F_i$’s are such that $z_{D_i}$ depends linearly on $a$, yield to the $n_L$ following linear equations:

$$Ma = b(x_0) \quad M \in \mathbb{R}^{n_L \times N_f \cdot n_f}$$

(59)

Equation (59) can be used to reduce the search space dimension. Indeed by eliminating dependent variables of $a$, one can define the new unknown vector of independent variables $p$ by:

$$a = F p + \gamma(x_0) \quad p \in \mathbb{R}^{n_p} \quad \text{with} \quad n_p = N_f \cdot n_f - n_L$$

with:

$$F := P \begin{pmatrix}
  -Q^{-1} \cdot R \\
  -I_{n_p \times n_p}
\end{pmatrix}$$

$$\gamma(x_0) := P \begin{pmatrix}
  Q^{-1} \cdot b(x_0) \\
  0_{n_p \times 1}
\end{pmatrix}$$

where $P \in \mathbb{R}^{n_f \cdot N_f \times n_f \cdot N_f}$ is a reordering orthonormal transformation matrix such that:

$$M P = \begin{bmatrix}
  Q_{n_L \times n_L} & R_{n_L \times n_p}
\end{bmatrix} \quad \text{with} \quad Q \text{ full rank.}$$

The problem can now be solved in $p$ instead of in $a$. 

12
The inverted pendulum problem

Initial state $X_0 = [0.873 0 0 0]$

$T_f = 1$ s  $T = 0.1$ s  $N_a = 15$

$\Lambda = 2.000$

Perturbation checking period = 0.01 s

Figure 1: Inverted pendulum with horizontal action
The ball and beam problem

Initial state $X_0 = [0 -0.1 0 0.1]$

$T_f = 3$ s             $T = 1$ s             $N_a = 10$

$\Lambda = 1.100$

Perturbation checking period = 0.1 s

Perturbation at $T_{pert} = 1.25$ s of $X_{pert} = [0 0.1 0 0]$
Finite time divergence

Initial state $X_0 = [0.5]$

$\tau_f = 2$ s              $T = 1$ s              $N_a = 20$

$\Lambda = 1.100$

Perturbation checking period = 0.05 s

Perturbation at $T_{\text{pert}} = 1.25$ s of $X_{\text{pert}} = [1.5]$

Figure 3: System with a finite escape time