Algorithms for Symbolic/Numeric Control of Affine Dynamical Systems *

Aude Rondepierre

Université Joseph Fourier,
Laboratoire de Modélisation et Calcul,
50 av. des Mathématiques.
B.P. 53 X, 38041 Grenoble, France.

Aude.Rondepierre@imag.fr

Jean-Guillaume Dumas

Université Joseph Fourier,
Laboratoire de Modélisation et Calcul,
50 av. des Mathématiques.
B.P. 53 X, 38041 Grenoble, France.

Jean-Guillaume.Dumas@imag.fr

ABSTRACT

We consider a general linear dynamical system and want to control its behavior. The goal is to reach a given target by minimizing a cost function. We provide a new generic algorithm with together exact, symbolic and numerical modules. In particular new efficient methods computing a block Kalman canonical exact decomposition and the optimal solutions are presented. We also propose a new numerical algorithm under-approximating the controllable domain in view of its analytical resolution in the context of singular sub-arcs.

Categories and Subject Descriptors:
I.1.2 [Symbolic and algebraic manipulation]: Algorithms, J.1.7 [Computer Applications]: Command and control.

General Terms: Algorithms.

Keywords: Affine Optimal Control Problems, Canonical Transformation, Controllability.

1. INTRODUCTION

Aerospace engineering, automatics and other industries provide a lot of optimization problems, which can be described by optimal control formulations: change of satellites orbits, flight planning, motion coordination [7] ([16] for more applications in aerospace industry). Optimal control has so become a more and more challenging domain and its theory has been extensively developed for many years. Nevertheless, the problem of synthesis of optimal feedback is not solved, even for linear systems. In some specific cases like time-optimal control problems, adequate solutions have been found [18, §3],[2, 17, 16]. Also, control theory lacks generic algorithms, specially when singular subarcs appear[14, 19, 2].

Furthermore, in "real-life", optimal control problems are fully nonlinear. Therefore most of the algorithms presented here have been developed towards their application to the hybrid control of nonlinear dynamical systems: in [20], we propose a piecewise affine approximation by way of a hybrid automaton. In each cell, the local optimal control problem is affine and techniques developed here will be applied.

In this paper we consider a linear dynamical system:

$$\left\{ \begin{array}{lcl} \dot{X}(t) & = & AX(t) + Bu(t) \\ X(0) & = & X_0 \end{array} \right. \quad (1)$$

where $\forall t \geq 0, X(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{U}_m = \{s_1, \dots, s_p\} \subset \mathbb{R}^m$ is the control. We want to control the system (1) from an initial state X_0 to a target $X_f = 0$ at an unspecified time t_f, in such a way that the functional: $J(X_0, u(.)) = \int_0^{+\infty} l(X(t), u(t))dt$ is minimized.

Here, we provide a full implementation analyzing linear optimal control problems as general as possible. Our algorithm is divided in four steps:

(1) Canonical transformation (see §2).
(2) Approximation of the controllable set (see §3.2).
(3) Computation of optimal solutions (see §4).
(4) Inverse transformation (see §2.2).

Each step can be done in several different ways and some salient features of our presentation are:

- a new and more efficient implementation by block matrices of the exact computation of Kalman decomposition.
- symbolic computation of the boundaries of domains, where the optimal control is constant (see e.g. algorithms 5 and 6).
- a new numerical method to compute an under-approximation of the controllable domain.
- an efficient implementation of the optimal solution computation, for a very large class of cost functions using subroutines previously defined (see algorithm 7).

Our approach enables the high dimensions treatment, even when compared to numerical softwares. Indeed, numerical methods developed from the Hamilton-Jacobi-Bellman

*Work partially supported by the Region Rhône-Alpes (Calcel project).

ISSAC'05, July 24–27, 2005, Beijing, China.

(HJB) theory are known to suffer from the dimension: they generally require to generate a grid over a bounded region of the state space. If the state dimension is n and the number of discretization points per dimension is 50 (which is the minimum acceptable: 100 could still be a bit sparse), one has to consider 50^n grid points. Despite the development of efficient techniques for the choice of the discretization points like adaptative mesh, computations grow exponentially in the state dimension. Consequently dimension 4 or 5 cannot be exceeded, while e.g. aerospace [16] requires treatments of dimensions 6 or 7. By the use of Hybrid Computation [4] combining numerical analysis and computer algebra, we are now able to deal with high dimensions (see [20, Part II]): first the mesh is made on the fly to reduce the complexity. Then, at the vertices of the mesh, only a linear interpolation of our complex system is performed. In each cell, the system is linear and one need to develop methods as symbolic as possible: basically, an analytical approach must allow to improve the hybrid approximation.

The paper is organized as follows: in section 2, we will develop explicit algorithms to compute exactly the canonical transformation of any linear optimal control problem and then the exact inverse transformation. In section 3, we provide a numerical controllability analysis and then in section 4, the analytical computation of optimal solutions of the canonical problem.

2. CANONICAL TRANSFORMATION

Linear control systems have been widely analyzed. In [13, 12], Kalman considers constant linear optimal control problems without constraints on the control. In this context, we have two well-known results: the first one is a controllability criterion (see [13, 3] for more details), and the second is the following decomposition theorem:

THEOREM 1 ([12] KALMAN CANONICAL STRUCTURE). *Let A and B be real matrices having respective sizes $n \times n$ and $n \times m$. There exists an invertible $n \times n$ matrix T such that:*

$$T^{-1}AT = \begin{bmatrix} A_1 & A_2 \\ 0 & A_3 \end{bmatrix} \qquad T^{-1}B = \begin{bmatrix} B_1 \\ 0 \end{bmatrix}$$

where $r = rk([B\ AB \ldots A^{n-1}B]) = rk([B_1\ A_1B_1 \ldots A_1^{n-1}B_1])$ A_1 is a r dimensional square matrix and B_1 a $r \times m$ matrix.

There exist many numerical algorithms computing the Kalman canonical form of full rank linear dynamical systems. Next, we consider rank deficient systems, for which exact computation of the rank is needed. Therefore we propose a new explicit and exact algorithm for the Kalman decomposition. Our approach is to use block versions of the linear algebra algorithms as in [5] in order to improve the locality of the computations and treat larger problems faster. Indeed, we are then able to compute exactly the rank of the system and use the LQUP decompostion of [6] (nowadays quite as fast as numerical routines) to perform the decomposition.

2.1 Block Canonical decomposition

We consider the general linear system (1). Our decomposition is divided into two steps: we first reduce the system to one with a full rank mapping of the control and second apply a LQUP decomposition to the Kalman matrix.

2.1.1 Simplification to $rk(B) = m$

LEMMA 1. *Let us consider the linear system (1). There exists a full rank matrix $\tilde{B} \in \mathbb{R}^{n \times rk(B)}$ and a linear mapping $\Phi \in \mathbb{R}^{rk(B) \times m}$ such that: $\dot{X}(t) = AX(t) + \tilde{B}\Phi u(t)$.*

PROOF. $b = rk(B)$. If $b < m$, then there exists a column permutation $P \in \mathbb{R}^{m \times m}$ s.t.: $BP = [\tilde{B}|B_0]$ where $\tilde{B} \in \mathbb{R}^{n \times b}$ and $rk(\tilde{B}) = b$. Moreover, the column vectors of B_0 are linearly dependent of those of $\tilde{B}$, i.e.: $\exists \Lambda \in \mathbb{R}^{b \times m-b}, B_0 = \tilde{B}\Lambda$. Hence: $B = \tilde{B}\ [I_b|\Lambda]P^{-1}$ and $\Phi = [I_b|\Lambda]P^{-1}$. □

In the following, we will denote by $FullRank(B)$ the algorithm computing $(b, \tilde{B}, \Phi)$ from a matrix B as in the lemma.

2.1.2 Block Kalman Canonical Form

Now we want to decompose the state space of our linear system into a controllable part and an uncontrollable one. The classical method is to introduce the linear subspace $W(A, B) = span(B, AB, \ldots, A^{n-1}B)$ and then prove that W is the first subspace of $\mathbb{R}^n$ satisfying both: (i) $Im(B) \subset W$ (ii) W is A-invariant. The method is then to decompose the state space $\mathbb{R}^n$ into $W \oplus \bar{W}$: one has to compute a basis of the subspace $W(A, B)$ and to complete it for the whole state space. The matrix T of theorem 1 would be the change matrix from the canonical basis to the computed basis.
In this paper we propose a new approach via block matrix computation developed in collaboration with C. Pernet: we use the so-called $LQUP$ decomposition of a $x \times y$ matrix of rank r, where $U = \left[\begin{array}{c|c} U_1 & U_2 \\ \hline 0 & 0 \end{array}\right]$ is $x \times y$, U_1 is an upper triangular $r \times r$ invertible matrix, L is $x \times x$, lower block-triangular, and P and Q are permutation matrices [10].

Algorithm 1 BlockKalmanForm

Require: A $n \times n$ matrix, B $n \times m$ matrix.
Ensure: r, T, A_1, A_2, A_3, B_1 as in theorem 1.

$K = [B|AB|\ldots|A^{n-1}B]$;
$(L, Q, U_1, U_2, P, r) = LQUP(K^T)$;
if $r = n$ **then**
  Return $(n, I_n, A, \emptyset, \emptyset, B)$.
end if
Form $\delta = [I_r|0]Q^T LQ \left[\begin{array}{c} I_r \\ \hline 0 \end{array}\right]$, lower triangular.
Form $d = [I_{r+1..nm}|0]Q^T LQ \left[\begin{array}{c} I_r \\ \hline 0 \end{array}\right]$.
$G = [I_r|0]Q^T K^T$.
$C_1 = G(A^T P^T \left[\begin{array}{c} I_r \\ \hline 0 \end{array}\right] U_1^{-1}\delta^{-1})$
$C_2 = [0|I_{n-r}]P(A^T P^T \left[\begin{array}{c} I_r \\ \hline 0 \end{array}\right] U_1^{-1}\delta^{-1})$
$C_3 = [0|I_{n-r}]PA^T P^T \left[\begin{array}{c} -U_1^{-1}U_2 \\ \hline I_{n-r} \end{array}\right]$
$Q_1 = [I_m|0]Q \left[\begin{array}{c} I_r \\ \hline d\delta^{-1} \end{array}\right]$ $\{Q_1$ is $m \times r\}$
Return $(r, \left[\begin{array}{c} G \\ \hline [0|I_{n-r}]P \end{array}\right]^T, C_1^T, C_2^T, C_3^T, Q_1^T)$.

THEOREM 2. *Algorithm 1 is correct and its arithmetic complexity is $O(n^\omega m)$*[1]*.*

[1] where ω is the exponent of matrix multiplication (3 for the classical algorithm and 2.3755 for Coppersmith-Winograd's)

PROOF. The full proof is given in appendix A. It has three parts and is actually another, constructive, proof of Kalman's theorem:
1. First, use the generalization of the companion matrix decomposition to prove that $GA^T = C_1 G$.
2. Second, use the latter to show that $T^{-1}AT$ is block triangular.
3. Show that $T^{-1}B$ has generic rank profile.
4. Now for the complexity: building the Kalman matrix is n matrix multiplications $n \times n$ by $n \times m$, each requiring $O(n^{\omega-1}m)$ operations. Following [6, Lemma 4.1], the LQUP decomposition requires $O\left(n^{\omega-1}(mn+n)\right)$ operations. Those two costs dominate the remaining operations: two triangular inversions $O(r^\omega)$, some permutations and column selections, and small matrix multiplications (GA^T is $O(rn^{\omega-1})$ and $d\delta^{-1}$ is $O(nmr^{\omega-1})$ where $r \leq n$.). $\square$

Our implementation and constructive proof of the Kalman decomposition are based on LQUP factorization and block matrix computation. The better locality induced by this block version enables the use of very fast Basic Linear Algebra Subroutines, even with symbolic computations [6]. Therefore the computation time is improved. Moreover if we first apply the algorithm *FullRank* of paragraph 2.1.1, the system (1) can be replaced by another linear one:

$$\dot{Y}(t) = \begin{bmatrix} A_1 & A_2 \\ 0 & A_3 \end{bmatrix} Y(t) + \begin{bmatrix} B_1 \\ 0 \end{bmatrix} \tilde{u}(t) \qquad (2)$$

via possibly two variable changes: $\begin{cases} Y(t) &= T^{-1}X(t) \\ \tilde{u}(t) &= \Phi u(t) \end{cases}$
Next, we use these decomposition in order to define a canonical optimal control problem, simpler to solve.

2.2 Inverse transformation

In this section, the focus is on the explicit construction of a new linear optimal control problem under the dynamic (2). A new cost function and new state and control spaces have to be constructed and initial solutions have to be recovered.

2.2.1 Control Space

In this paragraph we focus on the construction of a new control space for the linear system (2). By assumptions (see section 1), the control $u(.)$ satisfies: $\forall t \geq 0,\ u(t) \in \mathbb{U}_m = Conv(s_1, \ldots, s_p)$. Moreover the image of a polyhedron in finite dimension by a linear mapping is polyhedral. So the new control polyhedron is: $\Phi\mathbb{U}_m = Conv(\Phi s_1, \ldots, \Phi s_p)$. Note that, if $rk(B) = m$, then $\Phi = I_m$, so that no control change is needed. When $rk(B) < m$, the main difficulty to build our optimal control problem is that there is not any invertible relation between u and $\tilde{u}$; consequently to switch from one control problem to the other, we will first need to define the pseudo-inverse of the control change matrix: $\tilde{s}_1, \ldots, \tilde{s}_{p'}$ are the vertices of ΦU_m. We introduce the Moore-Penrose pseudo-inverse $\Psi \in \mathbb{R}^{m\times b}$) [21] of the matrix $\Phi = [I_b|\Lambda]P^{-1}$: $\Psi = P\left[\frac{I_b}{0}\right]$ defined by: $\forall i \in \{1, \ldots, p'\}, \Psi\tilde{s}_i = s_k$, where $k = min\{j \in \{1, \ldots, p\} \ / \ \Phi s_j = \tilde{s}_i\}$. By linearity, Ψ is also well defined on the whole polyhedron $\Phi\mathbb{U}_m$, indeed: $\forall \tilde{u} \in \Phi U_m, \exists(\alpha_i)_{i=1\ldots p'} \in [0,1]^{p'}, \sum_{i=1}^{p'} \alpha_i = 1, \tilde{u} = \sum_{i=1}^{p'} \alpha_i \tilde{s}_i$. Hence $\Psi\tilde{u} = \sum_{i=1}^{p'} \alpha_i \Psi\tilde{s}_i$ and the proposition 1 is proven:

PROPOSITION 1. **(i)** $\Phi\Psi = I_b$, **(ii)** $\forall u \in \mathbb{U}_m, Bu = \tilde{B}\Phi u$, **(iii)** $\forall \tilde{u} \in \Phi\mathbb{U}_m, \tilde{B}\tilde{u} = B\Psi\tilde{u}$.

2.2.2 State Space

By construction, the change matrix T is non singular. Therefore, a trajectory $Y(.)$ from an initial point Y_0 corresponds to a trajectory $X(.) = TY(.)$ from the initial point $X_0 = TY_0$. Every trajectory is necessarily related to a control, the table 1 displays the correspondence between each trajectory.

Initial Problem (1)		Canonical Problem (3)
$(X(.), u(.))$	$\rightarrow$	$(T^{-1}X(.), \Phi u)$
$(TY(.), \Psi\tilde{u})$	$\leftarrow$	$(Y(.), \tilde{u}(.))$

Table 1: Matching trajectories

PROOF. The key point here is that a trajectory $(X(.), u(.))$ in the X-space is a solution of the system (1):

$$\begin{array}{rcl} X(t) &=& e^{At}X_0 + e^{At}\int_0^t e^{-Aw}Bu(w)dw \\ T^{-1}X(t) &=& e^{(T^{-1}AT)t}T^{-1}X_0 \\ && +\ e^{(T^{-1}AT)t}\int_0^t e^{-(T^{-1}AT)w}T^{-1}Bu(w)dw] \\ T^{-1}X(t) &=& e^{(T^{-1}AT)t}T^{-1}X_0 \\ && +\ e^{(T^{-1}AT)t}\int_0^t e^{-(T^{-1}AT)w}T^{-1}\tilde{B}\Phi u(w)dw] \end{array}$$

Then $(T^{-1}X(.), \Phi u(.))$ is a solution of (2), i.e. a trajectory in the Y-space. $\square$

2.2.3 Cost Function

Let X_0 be a controllable point. The value function related to the initial control problem (1) is defined by: $V(X_0) = \inf_{u(.)}\int_0^{+\infty} l(X(t), u(t))dt$. We want to define a new value function $\tilde{V}(Y_0) = \inf_{\tilde{u}(.)}\int_0^{+\infty}\tilde{l}(Y(t), \tilde{u}(t))dt$ such that the two related optimal control problems are equivalent.

First, the idea is to define a new cost function $\tilde{l}$, such that the value function is invariant by canonical transformation (i.e.: $V(X_0) = \tilde{V}(T^{-1}X_0)$). In this case, $\tilde{l}(Y, \tilde{u}) \mapsto l(TY, \Psi\tilde{u})$ and the new optimal control problem becomes:

"Minimize $\tilde{J}(Y_0, \tilde{u}(.)) = \int_0^{+\infty}\tilde{l}(Y(t), \tilde{u}(t))dt$ with respect to the control $\tilde{u}(.)$ under the dynamic (2) and the constraints: $\forall t \geq 0, \tilde{u}(t) \in Conv\{\tilde{s}_1, \ldots, \tilde{s}_{p'}\}$". (3)

We then have to verify that optimal solutions of this new problem correspond to optimal solutions of (1):

PROPOSITION 2. *Let* $(Y^*(.), \tilde{u}^*(.))$ *be an optimal solution of (3). Then* $(TY^*(.), \Psi\tilde{u}^*(.))$ *is an optimal solution of the initial problem (1) and* $V(TY_0) = \tilde{V}(Y_0)$.

The proof is by inspection of $J(X_0, \Psi\tilde{u}^*)$ and is given in appendix B.1.

2.2.4 Algorithms

To conclude the section, we describe two algorithms: *SimplifySystem* and *InverseTransformation*. From one given optimal control problem, *SimplifySystem* allows to define the canonical optimal control problem (see §2.1) ; once this problem is solved, *InverseTransformation* exactly computes the related optimal solutions of the initial problem (1) by the use of proposition 2. In the following algorithms, the pseudo-inverse Ψ of Φ is given e.g. by [21].

In this section we achieved the transformation of any linear optimal control problem into a canonical one. Moreover we have proved that optimal solutions of the canonical problem give optimal solutions of our initial problem. We have

Algorithm 2 SimplifySystem

Require: A, B, $U_m = [s_1, \ldots, s_p]$, l.

Ensure: $r, T, \Phi, \Psi, A_1, A_2, A_3, \tilde{U}, \tilde{l}$. (Data for the new optimal control problem: r, the state change matrix, the control change, the dynamic, the control space and the cost function).

{Definition of the new control space:}

$(b, \tilde{B}, M)$:=FullRank(B);

$\Psi := PseudoInverse(\Phi)$;

$\tilde{U} :=$ ConvexHull($\Phi s_1, \ldots, \Phi s_p$);

{Definition of the new optimal control problem:}

$(r, T, A_1, A_2, A_3, B_1) := BlockKalmanForm(A, \tilde{B})$;

{Definition of the new cost function:}

$\tilde{l} := (Y, \tilde{u}) \mapsto l(TY, \Psi\tilde{u})$

Return $(r, T, \Phi, \Psi, A_1, A_2, A_3, \tilde{U}, \tilde{l})$.

Algorithm 3 InverseTransformation

Require: $T, \Psi, Y^*, \tilde{u}^*$.

Return $(TY^*, \Psi\tilde{u}^*, \mathbb{U}_m, \Phi\mathbb{U}_m)$.

also proposed exact computation algorithms for switching to one problem to the other. Now, we can work on the canonical problem.

3. CONTROLLABLE DOMAIN

In this section, we consider the canonical optimal control problem previously defined and raise the question of its controllability: how to compute the set of initial points Y_0 for which the control problem (2) with the constraints $Y(0) = Y_0$; $Y(t_f) = 0$ and $\forall t \geq 0,\ u(t) \in \mathbb{U}_m = \{s_1, \ldots, s_p\} \subset \mathbb{R}^m$) admits a solution.

Let us state: $\forall t \geq 0,\ Y(t) = (Y_1(t), Y_2(t))$ where: $Y_1(t) \in \mathbb{R}^r$ and $Y_2(t) \in \mathbb{R}^{n-r}$. Thus the state space splits clearly up into an uncontrollable part ($\dot{Y}_2 = A_3 Y_2$) and a controllable one ($\dot{Y}_1 = A_1 Y_1 + A_2 Y_2 + B_1 u$). We study the controllability question in the two configurations.

3.1 Stabilization of the uncontrollable part

Let us consider the uncontrollable part:

$$\dot{Y}_2(t) = A_3 Y_2(t) \tag{4}$$

Clearly, $0 \in \mathbb{R}^{n-r}$ is an equilibrium point of (4). Thus the target 0 is reachable from everywhere if 0 is a stable focus of (4). In other words, the matrix A_3 has to be stable (all its eigenvalues have negative real parts).

In the following, we prove that the non-stability of A_3 involves constraints on $Y_2(0)$, so that we can easily come down to the case of a stable matrix A_3: we apply the Schur decomposition to A_3 and choose to sort its eigenvalues in such a way that: $\forall i = 1 \ldots k, Re(\alpha_i) < 0$ and $\forall i = k+1 \ldots (n-r), Re(\alpha_i) \geq 0$. Then there exists a unitary $Q \in \mathbb{C}^{n \times n}$ such that: $Q^* A_3 Q = D + N$ where $D = diag(\alpha_1, \ldots, \alpha_{n-r})$ and $N \in \mathbb{C}^{(n-r)\times(n-r)}$ is strictly upper triangular. Moreover (4) is easily solvable: $\forall t \geq 0, Y_2(t) = e^{A_3 t} Y_2(0)$. Hence:

$$\begin{aligned} Q^* Y_2(t) &= e^{Q^* A_3 Q t} Q^* Y_2(0) \\ &= e^{(D+N)t} Q^* Y_2(0) = e^{Dt} e^{Nt} Q^* Y_2(0) \\ &= \begin{bmatrix} e^{\alpha_1 t} & \star & \star \\ & & \star \\ & & e^{\alpha_{n-r} t} \end{bmatrix} Q^* Y_2(0) \end{aligned}$$

Nevertheless we **do not need to compute** e^{Nt}. Indeed, we can recursively show (by starting from $n-r$ to $k+1$) that: $([0|I_{n-r-k}]Q^*)Y_2(0) = 0$

Hence: $\forall t \geq 0,\ ([0|I_{n-r-k}]\ Q^*)Y_2(t) = 0$.

So under the variable change: $\tilde{Y}_2 = (Q^* Y_2)_{1..k}$, the system (2) then becomes:

$$\left\{ \begin{array}{rcll} \dot{Y}_1(t) & = & A_1 Y_1(t) & + \quad \tilde{A}_2 \tilde{Y}_2(t) + B_1 u(t) \\ \dot{\tilde{Y}}_2(t) & = & & \tilde{A}_3 \tilde{Y}_2(t) \end{array} \right.$$

where: $\tilde{A}_2 = A_2 Q \left[\frac{I_k}{0}\right]$ and $\tilde{A}_3 = (D+N)\left[\frac{I_k}{0}\right]$ is stable.

We have shown that the analysis of the uncontrollable part of the system (2) leads to define a subspace of the state space, namely $\{(Y_1, \tilde{Y}_2, 0) \in \mathbb{R}^r \times \mathbb{R}^k \times \mathbb{R}^{n-r-k}\}$. In this subspace, $\tilde{Y}_2(.)$ trajectories converge towards 0. From now on, we therefore restrict our analysis to a system (2) where the matrix A_3 is stable.

3.2 Under-Approximation of the Controllable Domain

Now, we assume w.l.o.g that the points s_i defining the control boundaries are such that: $s_i \notin Conv_{j \neq i}(s_j)$. Therefore, each point s_i is a vertex of the polytope $\mathbb{U}_m$ and we have (see §2): $rk(B) = m$, $rk([B|AB|\ldots|A^{n-1}B]) = n$.
We want to find the set of controllable points of our system. By time reversal, we come down to the computation of the attainable set from the target point 0. In [1], for safety verification, the idea is to compute a conservative over-approximation of the attainable set. They can thus certify that the system can not escape from an admissible set of states. On the contrary, we need a guaranty that Y_0 is controllable. Therefore we instead compute an under-approximation of this set.

Let us start by defining the controllable set C in our context:
$C = \{Y \in \mathbb{R}^n / \exists T \geq 0, \exists u : [0,T] \to U_m, Y = \int_0^T e^{-A\tau} Bu(\tau)d\tau\}$.
Indeed, any solution of a linear system $\dot{Y}(t) = AY(t) + Bu(t)$ has the form: $Y(t) = e^{At}Y(0) + \int_0^t e^{A(t-s)}Bu(s)ds$.

PROPOSITION 3. *The controllable domain C is a convex subset of the state space.*

The proof is given in appendix B.2. It defines (by convexity and at maximal time) a new control from that of some controllable points within C.
Now we can introduce our under-approximation of the domain by time-reversal of the control polytope:

COROLLARY 1. *Let $Y_i(.)$ be the trajectory from 0 by time reversal according to $u = s_i$. If $C(t) = Conv_{1..k}(Y_i(t))$, then*

$$C(t) \subset C \text{ and } \forall Y \in C(t), \exists \text{ a control } u, Y = \int_0^t e^{-A\tau} Bu(\tau)d\tau$$

Any point in $C(t)$ is said controllable at least in time t *and $C(t)$ is an* under-approximation *of the controllable set in time t.*

This gives us an algorithm to build our under-approximation in time T. Nevertheless for a given time T, the quality of the approximation could be very poor (see example 1, figure 1-(a)). To refine it, we choose to discretize the time interval $[0, T]$ in N subintervals. The under-approximation in time T is the convex hull of under-approximations in time $j * h$ for

$j = 1..N-1$ (where $h = T/N$) and the quality is significantly improved (see example 1, figure 1-(b)). We have thus defined the following algorithm, *UnderApproximation*, computing a set of controllable points.

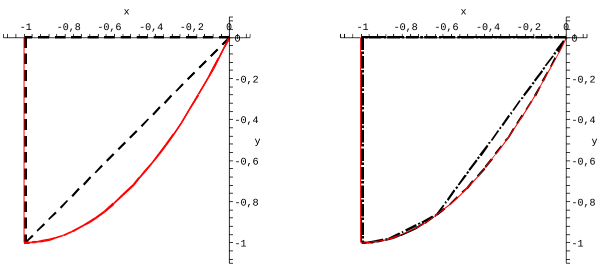
Algorithm 4 UnderApproximation

Require: A, B, U, T, h (where $U = Conv\{s_1, \ldots, s_p\}$).
Ensure: an under-approximation with a step $h = T/N$ of the controllable domain in time T.

```
ApproxVertices:=[0];
for all time step j (from 1 to N) do
    for all vertex s_i do
        Y_i(.) = trajectory from 0 with u = s_i;
        ApproxVertices:=ApproxVertices ∪{Y_i(jh)};
    end for
end for
Return ConvexHull(ApproxVertices);
```

Example 1 (2D Under-Approximations). *Let us consider the system:* $\dot{Y} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} Y + \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix} u$ *with* $u \in Conv([0, 0]^T, [1, 0]^T, [0, 1]^T)$.

The following figures show in dashes under-approximations of the controllable set (represented in plain line) for three refinements.



(a) (b)

Figure 1: Under-approximations in time $T = 5$ (a) without refinement ($N = 1$) (b) by discretizing ($N = 5$ in dash-dots - $N = 30$ in dashes, nearly superposed)

4. OPTIMAL SOLUTIONS

In this section, we present some theoretical results and algorithms for solving linear optimal control problems. The algorithm is as general and symbolic as possible to design optimal controllers. Recall that we want to control a linear system: $\dot{Y}(t) = AY(t) + Bu(t)$, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ from a controllable initial state Y_0 to a final state $Y_f = 0$ at an unspecified time t_f using the admissible control functions $u \in \mathbb{U}_m = Conv_{1..p}(s_p) \subset \mathbb{R}^m$ in such a way that: $J(Y_0, u(.)) = \int_0^{+\infty} l(Y(t), u(t))dt$ is minimized. According to the decomposition algorithm developed in section 2, we also assume: $rk(B) = m$ and $rk([B|AB|\ldots|A^{n-1}B]) = n$. To solve this canonical system, we now introduce the Hamiltonian function: $H(Y, u, \lambda) = l(Y, u) + \lambda^T AY + \lambda^T Bu$.
The pseudo-Hamiltonian formulation of the optimal control problem and the Pontryagin Minimum principle provide us the following optimization problem [18, §1], [2, §2],[17, §4]:

$\mathcal{P}$: "Minimize H with respect to the control variable $u \in \mathbb{U}_m$ under the constraints:

$$\dot{Y}(t) = \frac{\partial H}{\partial \lambda}(Y(t), u(t), \lambda(t)) \quad (5)$$

$$\dot{\lambda}(t)^T = -\frac{\partial H}{\partial Y}(Y(t), u(t), \lambda(t)) \quad (6)$$

and $H(Y^\star(t), \lambda^\star(t), u^\star(t)) = 0$ along the optimal trajectory."

Our algorithm is divided in two main steps: first, the controllable set is partitioned (see 4.1) in domains, inside which the optimal control is constant. In practice, we propose symbolic algorithms computing the boundaries of these cells (see §4.2). The second step requires to compute an optimal trajectory from an entry point to the target within each cell. In this section, the cost function l is assumed linear in the control: $l(Y, u) = l_0(Y) + l_1(Y)u$. The case were this function is nonlinear is actually simple: indeed, the Hamiltonian optimization problem could be solved by classical tools,since $H_u \overset{not}{=} \frac{\partial H}{\partial u} = 0$ is then solvable in the control variable u.

4.1 Singular control

Let us consider the optimization problem $\mathcal{P}$. By definition, $\mathcal{P}$ is a linear program. It thus admits solutions which may occur on the boundary of the polyhedral set $\mathbb{U}_m$. Now, any solution (Y, u, λ) of the Hamiltonian system (5) is said to be *extremal* and distinguish regular and singular solutions:

Definition 1. *An extremal* $(Y(t), u(t), \lambda(t))$ *is called* regular *on an interval* $[t_0, t_1]$*, if there exists* k *s.t., for almost all* $t \in [t_0, t_1]$*,*

$$[l_1(Y(t)) + \lambda^T(t)B]s_k < \min\{[l_1(Y(t)) + \lambda^T(t)B]s_i; i \neq k\}$$

Therefore, for any regular extremal $(Y(t), u(t), \lambda(t))$, the optimal control is given by the relation:

$$u(t) = s_i$$
$$\text{if } [l_1(Y(t)) + \lambda^T(t)B]s_i < \min_{j\neq i}\{[l_1(Y(t)) + \lambda^T(t)B]s_j\}$$

Consequently one can define a partition of the controllable set (see definition 2) as follows:

Definition 2. *An optimal trajectory* $Y(.)$ *belongs to the* domain Γ_i *on a time interval* $[t_0, t_1]$ *if the condition:* $\forall t \in [t_0, t_1]$, $\forall j \in \{1, ..., m\} - \{i\}$, $[l_1(Y(y)) + \lambda^T(t)B](s_i - s_j) < 0$ *holds. Thus at any point of the domain* Γ_i*, the optimal control is* $u(.) = s_i$ *and the related field vector is* $AY + Bs_i$*.*

Now, we introduce the switching function $S_{i,j}$, that describes transitions between the domains Γ_i and Γ_j:

Definition 3 (Switching function).

$$S_{i,j}(t) = H_u(Y(t), u(t), \lambda(t))(s_i - s_j)$$
$$= [l_1(Y(t)) + \lambda^T(t)B](s_i - s_j)$$

Then, the single zeros of $S_{i,j}$ give us the switching time between the domains Γ_i and Γ_j. However it may also be possible to find time intervals where the switching function is identically equal to zero. This typically corresponds to the appearance of singular arcs in each face of the polyhedral control set. Thus singular trajectories are:

Definition 4. *[22] A trajectory* $Y(.)$ *is called* ij-singular *on a time interval* $[t_0, t_1]$ *if the condition "*$S_{i,j}(t) = 0$ *and* $\forall k \neq i, j$, $S_{j,k}(t) < 0$*" holds for almost all* $t \in [t_0, t_1]$*.*

Just note that definition could be naturally extended to I-singular trajectories ($I \subset \{1, \dots, p\}$). According to definitions 2 and 4, we show that ij-singular trajectories geometrically correspond to the boundary between Γ_i and Γ_j. On this singular boundary, the optimal control is said to be singular and satisfies:

PROPOSITION 4. *Let us consider an ij-singular trajectory $Y(.)$ on a time interval $[t_0, t_1]$. Then:*

$$\forall t \in [t_0, t_1],\ u(t) \in [s_i, s_j].$$

Likewise, on an I-singular trajectory, $u(t) \in Conv_{k \in I}(s_k)$.

4.2 Boundaries computation

At this point of our analysis, we have partitioned our state space in domains delimited by:

- singular boundaries (see e.g. [17, fully optimal problem]).
- mixed and non singular boundaries (see [11, ex. 1]).
- non singular boundaries (see [18, time-optimal problems]).

In our linear control problem, the Hamiltonian has the form: $H_0(Y, \lambda) + H_1(Y, \lambda)u$. Now, let us consider the boundary between domains Γ_i and Γ_j. Then, in the whole paragraph, we use H with the form: $H(Y, v, \lambda) = H_0(Y, \lambda) + H_1(Y, \lambda)(s_j + (s_i - s_j)v)$ where $v \in [0, 1]$ (since $u \in [s_i, s_j]$ with proposition 4) to show how to symbolically compute the considered boundary, when it exists.

4.2.1 Switch rules

In this paragraph we briefly describe a method to compute the allowable "switching directions" [11] in the state space.

¿From examination of the sign of $\frac{d}{dt} H_v(Y(t), v, \lambda(t))$ at switching points (i.e. $H_v(Y(t), \lambda(t)) = 0$ and $H(Y(t), v, \lambda(t)) = 0$), it is possible to determine whether switchings from $u = s_i$ to $u = s_j$ are allowed in a given region of the state space.

4.2.2 Singular boundaries

In this paragraph we present a symbolic algorithm computing singular boundaries when they exist. This algorithm is essentially based on the Pontryagin maximum principle [18] and classical results in the theory of singular extrema (see [14, 19, 2] for more details).

We show on table 2 some performances of this algorithm in high dimension where U_m is a random simplex in $\mathbb{R}^m$ and $n = m$. Note that we still have to check that the

n	2	3	4	5	6	7	8
cpu (s)	0.16	0.22	0.35	0.56	0.91	1.51	2.43
n	9	10	11	12	13	14	15
cpu (s)	4.21	7.03	10.53	19.06	31.38	53.85	94.18

Table 2: Symbolic singular boundaries timings

so-computed boundary really exist in the controllable domain and that the switching conditions are satisfied: $\forall k \notin \{i, j\}, S_{i,k} < 0$. However, we show next that these conditions are not always sufficient to determine if a computed singular boundary is valid or not. Such cases appear when the computed singular control explicitly depends of the state Y.

Algorithm 5 ij-singular boundary

Require: i and j, indices of the considered Γ domains.
Require: $H(Y, v, \lambda)$.
Ensure: φ, where $\varphi(X) = 0$ defines the ij-boundary
Ensure: $u^\star$ the ij-singular optimal control.
Ensure: $\lambda^\star$ the optimal Pontryagin parameter.
1: Compute the smallest integer K such that: $\frac{\partial}{\partial v}(\frac{d^{2K}}{dt^{2K}} H_v) \neq 0$ (where $H_v = \frac{\partial H}{\partial v}$)
2: **if** the Legendre-Clebsh (LC) condition [14, 19]: $(-1)^K \frac{\partial}{\partial v}(\frac{d^{2K}}{dt^{2K}} H_v) \geq 0$ is not satisfied **then**
3: Return "no singular solution".
4: **end if**
5: Solve $(\mathcal{S})$ $\{H = 0, H_v = 0, \left(\frac{d^i}{dt^i} H_v = 0\right)_{i=1..2K}\}$ {$(\mathcal{S})$ is linear in v and λ, hence we obtain the exact singular values of v and λ in relation with Y. The remaining relation gives the equation ($\varphi(Y) = 0$) of the boundary.}
6: Return $(\varphi(Y), s_j + (s_i - s_j)v(Y), \lambda(Y))$.

While the related boundary is bounded, the whole boundary between Γ_i and Γ_j is necessarily also made of a regular part. The next paragraph is devoted to its computation.

4.2.3 Mixed boundaries

In this paragraph we assume that we have already computed the singular boundary between two domains Γ_i and Γ_j and check the existence condition of these boundary. So we have its equation: $\varphi(Y) = 0$ under the constraint $0 \leq v(Y) \leq 1$, the singular control u^* and the related λ^*. We now want to compute the related regular boundary (see algorithm 6).

Algorithm 6 MixedBoundary

Require: i and j, indices of the considered Γ domains.
Require: φ equation of the ij-boundary.
Require: $\lambda^\star$ optimal Pontryagin parameter on the ij-boundary.
Ensure: a parameterization of the non singular boundary between Γ_i and Γ_j.
1: Parameterize the singular boundary (by the implicit functions theorem) :$\psi(\xi)$ (i.e. such that $\varphi(\psi(\xi)) = 0$).
2: **for** $s \in \{s_i, s_j\}$ **do**
3: Compute the trajectory $Y[\psi(\xi), s]$ from $\psi(\xi)$ by time reversal with $u = s$:
$$Y[\psi(\xi), s](t) = e^{At}\psi(\xi) + e^{At}\int_0^t e^{-Aw} B s dw$$
4: Solve the Euler-Lagrange equations (6) with the initial condition $\lambda[\psi(\xi), s](0) = \lambda^*(\psi(\xi))$.
5: Compute the first time $t(\xi) < 0$ for which the switching condition between s_i and s_j holds, i.e.: $S_{i,j}(t) = 0$ (see definition 3). *No solution $t(\xi)$ invalidates the singular boundary so that the boundary between Γ_i and Γ_j is necessarily regular.*
6: **end for**
7: Return the switching curve equations(if they exist):
$$Y[\psi(\xi), s](t(\xi)) = 0,\ s \in \{s_i, s_j\}$$

Consider the system [11, Example 1]:

$$\begin{cases} \dot{X}(t) &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} X(t) + \begin{bmatrix} 1 \\ -1 \end{bmatrix} u(t) \\ X(0) &= X_0 \end{cases} \tag{7}$$

where $X \in \mathbb{R}^2$, $|u| < 1$ and the performance index to be minimized is: $J(X, u(.)) = \int_0^T \frac{1}{2}x_1(t)^2 dt$. (Note that (7) already is under its canonical form since $rk([B|AB]) = 2$).

[11] provides the full analysis of the singular controls and the graph of the allowable switching regions [11, fig. 5]. There, the singular boundary is defined by $\phi(x_1, x_2) = x_1 + 2x_2 = 0$ and $-1 \leq x_2 \leq 1$ and the switching function is $S(t) = 2(\lambda_1(t) - \lambda_2(t))$. Due to the constraint $-1 \leq x_2 \leq 1$, the singular boundary does not allow us to draw a partition of the state space. However, we are able to complete [11]'s results by the computation of the whole boundary between the controls $u = 1$ and $u = -1$: as we have a valid singular boundary, we can now apply our $MixedBoundary$ algorithm as shown on figure 2:
1. Parametrize the singular boundary: $\psi(\xi) = (-2\xi, \xi)$.
2/3. For u=1, compute the trajectory from $\psi(\xi)$ according to $u = 1$ by time reversal: $Y[\psi(\xi), 1](t) = [-\frac{1}{2}t^2 + \xi t + t - 2\xi, -t + \xi]$.
4. Solve the Euler-Lagrange equations $S(0) = 0$
5. Compute the first time $t(\xi)$ such that $S(t(\xi)) = 0$ holds:

$$t(\xi) = 2\xi - 2\sqrt{\xi^2 - 3\xi + 3}$$

6. Repeat steps 3., 4. and 5. with $u = -1$.
7. Return the parametrized switching curve equation:

$$\begin{cases} (2\xi\sqrt{\xi^2 - 3\xi + 3} - 2\xi^2 + 6\xi - 6 - 2\sqrt{\xi^2 - 3\xi + 3}, \\ \qquad -\xi + 2\sqrt{\xi^2 - 3\xi + 3},\ \xi \in [-1, 1]) \\ (2\xi\sqrt{\xi^2 + 3\xi + 3} + 2\xi^2 + 6\xi + 6 + 2\sqrt{\xi^2 + 3\xi + 3}, \\ \qquad -\xi - 2\sqrt{\xi^2 + 3\xi + 3},\ \xi \in [-1, 1]) \end{cases}$$

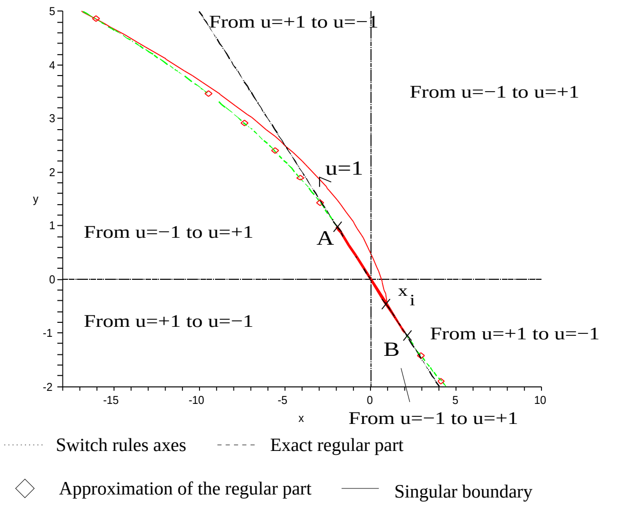


Figure 2: MixedBoundary in Gibson's problem

REMARK 1. *In some cases the equation $S_{i,j}(t) = 0$ is not solved by Maple, so that we cannot express the time $t(\xi)$ in step 5. Nevertheless, by discretizing the given singular boundary, we can compute a discretization of the regular part of the searched boundary. Then, by interpolation, we find again the whole boundary.*

4.2.4 Non Singular Boundaries

In this paragraph, we consider the case where there is no singular or mixed boundary between the two domains Γ_i and Γ_j. The optimal control is then called **bang-bang** (i.e. piecewise constant with values in $\{s_i, s_j\}$). Let us distinguish two possible configurations:
- For all $(i, j) \in \{1, \ldots, m\}^2$, $i \neq j$, the boundary between Γ_i and Γ_j is non singular.
In this case, one method is to compute all the switching functions $S_{i,j}$ (see definition 3). After that we can study the zeros of $S_{i,j}$ and deduce the transition time τ between two of them. By time reversal, we start from the origin and build the switching curve. This method is well described in [17].
- There exists $k \in \{1, \ldots, m\} - \{i, j\}$, such that the boundary between Γ_j and Γ_k exists and is singular. In this case we come down to the same technique than for mixed boundaries: the idea is to take a parameterization of the singular boundary between Γ_j and Γ_k. We consider by time reversal the trajectory from a point of this boundary according to $u = s \in \{s_j, s_k\}$ and compute the first negative time for which the switching condition: $S_{j,k}(t) = 0$ holds.
This latter algorithm, $NonSingBoundary$ is based on the following proposition:

PROPOSITION 5. *Let $Y(.)$ be an optimal trajectory from an initial point Y_0 and $u(.)$ the associated optimal control. We assume that there exists a time $T > 0$, such that: $\exists \epsilon > 0$, $Y(.)$ regular over $[T - \epsilon, T]$ and $Y(T + .)$ is ij-singular. Then: $\forall t \in [T - \epsilon, T]$, $u(t) \in \{s_i, s_j\}$.*

In conclusion, we have proposed three algorithms to compute boundaries between the domains Γ_i. We can now define a general one $Boundary$ that compute the boundary between two given domains Γ_i and Γ_j: $(\varphi, \omega, u) := Boundary(i, j)$ where φ is the equation of the boundary, ω is defined by: $\omega(Y) = \begin{cases} 1 \text{ if } \varphi(Y) = 0 \text{ is singular} \\ 0 \text{ otherwise} \end{cases}$ and u^* is the related optimal control. We therefore have all the necessary subroutines to solve our problem.

4.3 Linear Optimal Control

In this section, we detail the general algorithm for solving any linear optimal control problem. The principle is as follows: after a virtual partition of the state space in Γ_i domains, one follows the trajectory within each cell of the partition. Indeed, in each cell and in every boundaries, the control is known thanks to the algorithms of section 4.2. When the trajectory reaches a boundary, there is a switching of cell and a control change. This goes on till the target is reached.
Once the canonical problem is solved with algorithm 7, we just have to apply the inverse transformation 3 to come down to optimal solutions of our initial control problem.

5. CONCLUSION

We have presented an algorithm for solving general linear optimal control problems. First we propose an explicit method to transform any problem into a canonical one by the way of a block Kalman decomposition. We have also developed generic algorithms solving the canonical problem even when complex boundaries occur. Yet, two important new features of our algorithms are that we give a full generic implementation and that they are mainly symbolic. Also,all the algorithms presented here have been implemented in Maple and work in high dimensions[2].

[2]The maplets are available online at: `www-lmc.imag.fr/lmc-mosaic/Jean-Guillaume.Dumas/SHOC`

Algorithm 7 LinearOptimalControl

Require: A, B, Y_0, $l(Y,u) = l_0(Y) + l_1(Y)u$ and $\{s_1,\dots,s_m\}$.
Ensure: Optimal trajectory, control and value function.
$V := 0$; {Initialize switching functions}
$S_{i,j} = \frac{\partial}{\partial u} H(Y, s_j + (s_i - s_j)v, \lambda) = l_1(Y) + \lambda^T B(s_i - s_j)$. {Virtual partition of the state space}
$I = \{i \in [|1,m|]/\{\lambda/\forall j \neq i, S_{i,j} < 0\} \neq \emptyset\}$.
for all $i \in I$ and $j \in I$ such that $i < j$ **do**
    $(\varphi_{i,j}, \omega_{i,j}, u_{i,j}) := Boundary(i,j)$.
    $(\tilde{\Gamma}_j)_{j\in I}$ is the induced partition of the controllable set.
end for
{Identification of the domains where $u = s_i$}
for all $j \in I$ **do**
    if $\partial \tilde{D}_j == \bigcup_k \{Y$ s.t. $\varphi_{i,k}(Y) = 0\}$ **then**
        $\Gamma_i := \tilde{\Gamma}_j$.
    end if
end for
$k := 0$; $T_0 := 0$;
{Within each cell, reach the boundary}
while $Y_k \neq 0$ **do**
    Find i s.t. $Y_k \in \overline{\Gamma_i}$.
    if $Y_k \in \partial\Gamma_i$ **then**
        Find j s.t. $\varphi_{i,j}(Y_k) = 0$.
        $u := u_{i,j}(Y_k)$;
        if $\omega_{i,j}(Y_k) == 1$ {0 is reached on this ij-singular boundary.} **then**
            $T_{k+1} :=$ Solution of $Y[Y_k, u](t) = 0$;
            break while loop;
        end if
    else
        $u := s_i$;
    end if
    {Piecewise solution}
    Compute $T_{k+1} = \inf\{t > 0; Y[Y_k,u](t) \in \partial\Gamma_i\}$
    $Y_{k+1} := Y[Y_k,u](T_{k+1})$.
    $u^\star := u$ for $t \in [T_k, T_{k+1}]$;
    $Y := Y[Y_k,u]$ for $t \in [T_k, T_{k+1}]$;
    $V := V + \int_0^{T_{k+1}-T_k} l(Y[Y_k,u](t)dt$
end while
Return $(Y, u^\star, V)$

Further developments already are in progress:

• Complete the whole algorithm for a cost function nonlinear in the control. In this case, the Hamiltonian optimization problem could be solved by classical tools. Indeed, $\frac{\partial H}{\partial u} = 0$ can now be solved in the control variable u.

• The UnderApproximation and solving algorithms have been performed on linear dynamical systems under the canonical form where $A_2 = 0$ (see §3.2 and section 4). These two algorithms have to be extended for any canonical form (see (2)). In practice, this corresponds to the appearance of a perturbation time function $t \to A_2 e^{A_3 t} Y_2(0)$ in the dynamical system. The technique does not change, but practical implementations are slightly more complex.

• The UnderApproximation could be refined and a study of the approximation error has still to be made. The idea is to consider cases where the dynamical system for $u = s_i$ admits one (or an infinite number of) equilibrium point P_i (note that 0 is an equilibrium point when $u = 0$). The under-approximation can e.g. be completed by the convex hull of trajectories from P_j that go through (or tend towards) P_j by time reversal according to $u = s_i$. Also, a rigorous proof of the convergence of our under-approximation towards the real controllable set has still to be completed.

6. ACKNOWLEDGMENTS

We would like to thank Kevin Hamon for his collaboration and for the work done towards generic algorithms (see [9]).

7. REFERENCES

[1] E. Asarin, T. Dang, and A. Girard. Reachability of non-linear systems using conservative approximations. In *Proceedings of the 2003 Hybrid Systems: Computation and Control*, pages 20–35, Apr. 2003.

[2] A. Bryson and Y. Ho. *Applied Optimal Control.* Hemisphere, 1975.

[3] D. Delchamps. *State Space and Input-Output Linear Systems.* Springer-Verlag, 1988.

[4] J. Della Dora, A. Maignan, M. Mirica-Ruse, and S. Yovine. Hybrid computation. In *ISSAC'2001, London, Ontario*, July 2001.

[5] J.-G. Dumas, T. Gautier, and C. Pernet. Finite field linear algebra subroutines. In *ISSAC'2002, Lille, France*, pages 63–74, July 2002.

[6] J.-G. Dumas, P. Giorgi, and C. Pernet. FFPACK: Finite Field Linear Algebra Package. In *ISSAC'2004, Santander, Spain*, July 2004.

[7] R. Fierro, A. K. Das, V.Kumar, and J. P. Ostrowski. Hybrid control of formations of robots. 2001.

[8] G. H. Golub and C. F. Van Loan. *Matrix computations third ed.*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[9] K. Hamon. Contrôle optimal et algorithme de calcul des trajectoires. Technical report, LMC-IMAG, 2004.

[10] O. Ibarra, S. Moran, and R. Hui. A generalization of the fast lup matrix decomposition algorithm and applications. *Journal of Algorithms*, 3:45–56, 1982.

[11] C. Johnson and J. Gibson. Singular solutions in problems of optimal control. *IEEE Transactions on Automatic Control*, 8:4–15, 1963.

[12] R. Kalman. Canonical structure of linear dynamical systems. In *Proceedings of the National Academy of Sciences*, pages 596–600, 1961.

[13] R. Kalman. Mathematical description of linear dynamical systems. *Siam Journal on Control*, 1:152–292, 1963.

[14] H. Kelley, R. Kopp, and H. G. Moyer. *Singular Extremals*, pages 63–101. Academic Press, 1967.

[15] C. Pernet. Calcul du polynôme caractéristique sur des corps finis. Master's thesis, Université Joseph Fourier, jun 2003.

[16] H. Pesch. A practical guide to the solutions of real-life optimal control problems. *Parametric Optimization. Control Cybernet*, 23:7–60, 1994.

[17] E. Pinch. *Optimal Control and the Calculus of Variations.* Oxford University Press, 1993.

[18] L. Pontryagin, V. Boltiansky, R. Gamkrelidze, and E. Michtchenko. *Théorie mathématique des processus optimaux.* Editions de Moscou, 1974.

[19] H. Robbins. A generalized Legendre-Clebsch condition for the singular cases of optimal control. *IBM Journal of Research and Development*, 11(4):361–372, 1967.

[20] A. Rondepierre and J.-G. Dumas. Hybrid optimal control of dynamical systems. Technical report IMAG-ccsd -00004191, arXiv math.OC/0502172. Fvrier 2005.

[21] B. D. Saunders. Black box methods for least squares problems. In *ISSAC'2001, London, Ontario*, 2001.

[22] M. Zelikin and V. Borisov. Optimal chattering feedback control. *Journal of Mathematical Sciences*, 114(3):1227–1344, 2003.

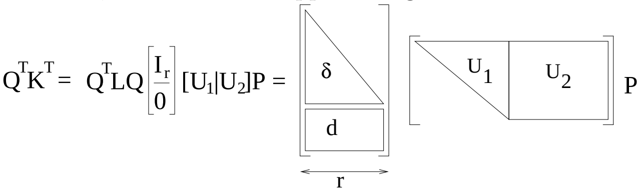
APPENDIX

A. PROOF OF THEOREM 2

0. Facts: the LQUP factorization [10] of K^T gives:

$$K^T = LQ\left[\begin{array}{c|c} U_1 & U_2 \\ \hline 0 & 0 \end{array}\right]P = (LQ\left[\begin{array}{c} I_r \\ \hline 0 \end{array}\right])[U_1|U_2]P \qquad (8)$$

Then, from [6, section 4.3], Q^TK^T is the product of a permutation, a lower and an upper triangular matrix:



This shows that $G = \delta[U_1|U_2]P$. Now, from its definition, we see that the matrix K satisfies that any block column vector of AK is a linear combination of the block column vectors of K (since A^n is a linear combination of the A^i, namely that of its characteristic polynomial). Well this is also true of the permuted independent columns of K. In other words, there exist a linear combination Γ such that $GA^T = \Gamma G$ (see e.g. [8, §7.4] or [15, §3] for more details).

1. We now have to prove that $GA^T = C_1G$. Unfortunately, C_1 might be different form Γ since G is not square. But the fact that there exist a solution is sufficient as we will see next. First, $C_1GP^T = C_1\delta[U_1|U_2]$. Then, by the definition of C_1, we obtain $C_1GP^T = GA^TP^T\left[\begin{array}{c} I_r \\ \hline 0 \end{array}\right][I_r|U_1^{-1}U_2]$. We replace now GA^T by ΓG to prove that the latter permutes. Then $C_1GP^T = \Gamma\delta[U_1|U_2]PP^T\left[\begin{array}{c} I_r \\ \hline 0 \end{array}\right][I_r|U_1^{-1}U_2]$, so that $C_1GP^T = \Gamma\delta[U_1|U_2] = \Gamma GP^T$. We thus get $C_1G = \Gamma G = GA^T$ since the permutation matrix P is invertible.

2. Then note that the inverse of $\left[\begin{array}{c} G \\ \hline [0|I_{n-r}]P \end{array}\right] = T^T$ is

$$T^{-T} = P^T\left[\begin{array}{cc} U_1^{-1}\delta^{-1} & -U_1^{-1}U_2 \\ \hline 0 & I \end{array}\right]. \qquad (9)$$

This enables to compute $T^{-1}AT$ or, equivalently, $T^TA^TT^{-T}$ as follows:

$$\begin{aligned} T^TA^TT^{-T} &= \left[\begin{array}{c} G \\ \hline [0|I_{n-r}]P \end{array}\right]A^TT^{-T} = \left[\begin{array}{c} GA^T \\ \hline [0|I_{n-r}]PA^T \end{array}\right]T^{-T} \\ &= \left[\begin{array}{c} C_1G \\ \hline [0|I_{n-r}]PA^T \end{array}\right]T^{-T} \\ &= \left[\begin{array}{cc} C_1GP^T\left[\begin{array}{c} U_1^{-1}\delta^{-1} \\ \hline 0 \end{array}\right] & C_1G\left[\begin{array}{c} -U_1^{-1}U_2 \\ \hline I \end{array}\right] \\ C_2 & C_3 \end{array}\right] \\ &= \left[\begin{array}{cc} C_1 & 0 \\ C_2 & C_3 \end{array}\right] \text{ as } GP^T = [\delta U_1|\delta U_2]. \end{aligned}$$

3. There only remains to prove that $(T^{-1}B)^T = [B_1|0]$: just remark that $B^TT^{-T} = [I_m|0]K^TT^{-T}$ which, from (8) and (9), gives

$$\begin{aligned} B^TT^{-T} &= [I_m|0](QQ^T)LQ\left[\begin{array}{c} I_r \\ \hline 0 \end{array}\right][U_1|U_2]PT^{-T} \\ &= [I_m|0]Q\left[\begin{array}{c} \delta \\ \hline d \end{array}\right][\delta^{-1}|0] = [I_m|0]Q\left[\begin{array}{c|c} I_r & 0 \\ \hline d\delta^{-1} & 0 \end{array}\right] \end{aligned}$$

Moreover if B is assumed full rank (equal to $m \le r$), the permutation Q has no influence on the m first rows of K^T. In this case, B_1 is just the identity.

B. PROOFS OF THE PROPOSITIONS

B.1 Proof of proposition 2

Let us state: $X = TY$. $(Y^*, \tilde{u}^*)$ is an optimal solution of problem 3, so that: $\tilde{V}(Y_0) = \tilde{J}(Y_0, \tilde{u}^*)$. Let us prove that the control $\Psi\tilde{u}^*$ is optimal in the problem (1):

$$\begin{aligned} J(X_0, \Psi\tilde{u}^*) &= \int_0^{+\infty} l(X(t), \Psi\tilde{u}^*(t))dt \\ &= \int_0^{+\infty} \tilde{l}(Y(t), \tilde{u}^*(t))dt \\ &= \tilde{J}(Y_0, \tilde{u}^*) = \tilde{V}(Y_0) \end{aligned}$$

Moreover, $\Psi\tilde{u}^*$ is an admissible control for (1), so, by definition of the value function: $J(X_0, \Psi\tilde{u}^*) \ge V(X_0)$. But, if (1) possesses some solutions from X_0, then there exists a control u^* such that: $V(X_0) = J(X_0, u^*)$. So, according to table 1:

$$\begin{aligned} V(X_0) &= \int_0^{+\infty} l(X(t), u^*(t))dt = \int_0^{+\infty} \tilde{l}(Y(t), \Phi u^*(t))dt \\ &= \tilde{J}(Y_0, \Phi u^*) \ge \tilde{V}(Y_0) \end{aligned}$$

The only possibility is then $J(X_0, \Psi\tilde{u}^*) = \tilde{V}(Y_0) = V(X_0)$

B.2 Proof of proposition 3

We take some controllable points $(Y_i)_{1\le i\le k} \in C^k$ and another point $Y_0 \in Conv(Y_i; 1 \le i \le k)$; by convexity: $\exists(\alpha_i)_{1\le i\le k} \in [0,1]^k, \sum_{i=1}^k \alpha_i = 1$ and $Y_0 = \sum_{i=1}^k \alpha_i Y_i$. We want to prove that Y_0 is controllable. By definition, each Y_i satisfies:
$\exists T_i \ge 0, \exists u_i : [0,T] \to U_m, Y_i = \int_0^{T_i} e^{-A\tau}Bu_i(\tau)d\tau$. Consequently, $Y_0 = \sum_{i=1}^k \alpha_i \int_0^{T_i} e^{-A\tau}Bu_i(\tau)d\tau$. We then fix T to the maximal time T_i and define a new control as follows:
$\hat{u}_i(\tau) = \left\{\begin{array}{lll} u_i(\tau) & if & \tau \le T_i \\ 0 & if & T_i < \tau \le T \end{array}\right.$ and $u(\tau) = \sum_{i=1}^k \alpha_i\hat{u}_i(\tau)$.
Clearly $u(.)$ is an admissible control function (i.e. $\forall\tau \ge 0, u(\tau) \in \mathbb{U}_m$) and we conclude: $Y_0 = \int_0^T e^{-A\tau}Bu(\tau)d\tau$, hence: $Y_0 \in C$.

B.3 Proof of proposition 4

We consider an ij-singular trajectory at a time $t \in [t_0, t_1]$. Let us show that: $u(t) \in [s_i, s_j]$. As we know, the control is defined on a polyhedral convex set, so that: $\exists(\alpha_k(t))_{k=1\ldots m} \in [0,1]^m, \sum_{k=1}^m \alpha_k(t) = 1$ and $u(t) = \sum_{k=1}^m \alpha_k(t)s_k$. So we come down equivalently to minimize the Hamiltonian H with respect to $\alpha = (\alpha_k(t))_{k=1\ldots m, k\ne i}$. H is redefined by: $H = l(X) + \lambda^Tf(X) + \lambda^Tg(X)s_i + \sum_{k=1,k\ne i}^m \alpha_k(t)\lambda^Tg(X)(s_k - s_i)$
Moreover according to the definition 4 of an ij-singular trajectory, we have: $\forall k \ne i,j,\ \lambda^Tg(X)(s_k - s_i) > 0$. So necessarily, we must have: $\forall k \ne i,j,\ \alpha_k(t) = 0$ i.e.: $u(t) \in [s_i, s_j]$. The control of an I-singular trajectory is shown likewise to belong to $Conv(s_k)$.

B.4 Proof of proposition 5

By assumptions, the trajectory $Y(.)$ is regular over $[T-\epsilon, T]$. So we have: $\exists k \in \{1, \ldots, m\}, \forall t \in [T-\epsilon, T],\ u(t) = s_k$. Moreover $u(.)$ is the optimal control, so that (cf definition 1): $\forall t \in [T-\epsilon, T],\ S_{k,j}(t) < 0$ and consequently: $\forall\epsilon' \in]0, \epsilon[,\ \forall t \in [T-\epsilon', T],\ S_{k,j}(t) < 0$.
Moreover $\lambda(.)$ is a continuous time function, so does $S_{k,j}$; therefore we have:
$S_{k,j}(T) \le 0$ when $\epsilon' \to +\infty$. Moreover the singular trajectory's definition 4 involves: $S_{i,j}(T) = 0$ and $S_{j,k}(T) < 0$. Hence we necessarily have: $k \in \{i, j\}$.