Interpolating Control Toolbox (ICT)

Sheila Scialanga and Konstantinos Ampountolas

***Abstract*— Interpolating control toolbox (ICT) is a free and open-source MATLAB toolbox that implements interpolation-based control (IC) for time-invariant and uncertain time-varying linear discrete-time systems with local state and control constraints. The toolbox combines geometrical features to compute robust invariant sets offline and solves a linear programming problem to compute the required IC online. This paper provides an overview on interpolating control and shows how to use ICT to robustly control input centralised and input decentralised interconnected systems. ICT includes some demo files to compare the performance of centralised versus decentralised IC.**

I. INTRODUCTION

ICT is a free and open-source MATLAB toolbox for computing interpolation-based control of linear time-invariant (LTI) and time-varying (LTV) systems subject to local state and control constraints. The technique behind it is based on the interpolating control (IC) scheme introduced by [1], [2] and later extended by [3], [4] for decentralised IC. The main idea of IC is to blend a local high-gain (inner) controller, which satisfies some user-desired performance specifications, with a global low-gain (outer) controller via interpolation. The two controllers are defined in some invariant sets that are determined offline; the applied control is determined on-line with an inexpensive interpolation between the two controllers. For the interpolation, a low-dimensional linear programming (LP) problem is solved at each time instant.

The present version of the ICT implements the improved centralised interpolating control (cIC) method developed in [5] and the more efficient decentralised formulation for interconnected systems as in [3], [4]. The decentralised interpolating control (dIC) approach solves state and control constrained problems for weakly interconnected systems via distributed interpolation in low-dimensional spaces. Furthermore, dIC is well-suited for large-scale applications, e.g., transport systems, irrigation systems, heat conduction, etc.

ICT combines geometrical features to determine (robust) invariant sets and perform the required IC. A number of toolboxes include geometrical tools to compute and implement control routines, e.g., Hybrid toolbox [6], MPT3 [7], PnPMPC toolbox [8]. Computational geometry in ICT relies on the Invariant Set toolbox [9], which is extended by the authors to account for LTV systems.

IC computation requires *offline* and *online* processing and is effectuated in the ICT as follows:

- Define the system matrices and (local) control/state constraints (offline);
- Compute the maximal robust positively invariant sets for a high-gain feedback inner controller (offline);
- Compute the maximal robust controllable invariant set or the maximal M-step robust controllable set – Outer controller (offline);
- Compute the centralised control with `cIC` or the decentralised control with `dIC` by linear interpolation between the inner and outer controllers (online);

Finally, ICT includes demo files containing examples for LTI or LTV system of different sizes. These files also include routines to compare the performance of cIC and dIC.

S. Scialanga and K. Ampountolas are with the School of Engineering, University of Glasgow, Glasgow G12 8QQ, United Kingdom. E-mail: s.scialanga.1@research.gla.ac.uk; konstantinos.ampountolas@glasgow.ac.uk.

II. PRELIMINARIES

Notation: A set $S \in \mathbb{R}^n$ is *convex* if $\lambda\, x_1 + (1-\lambda)\, x_2 \in S$, for all $x_1, x_2 \in S$ and $\lambda \in [0,1]$. The *convex hull* of a set $S \in \mathbb{R}^n$ is the set of all convex combinations of points in S. It is denoted as *Conv(S)* = $\{\sum_{i=1}^{k} \lambda_i x_i : x_i \in S, \sum_{i=1}^{k} \lambda_i = 1, \lambda_i \geq 0, i = 1, \ldots, k\}$. Convexity is preserved by various operations, among others the intersection. A *Polyhedron* $\mathcal{P}$ with *$\mathcal{H}$-representation* is the set $\mathcal{P} = \{x \in \mathbb{R}^n : F\, x \leq g\}$ where $F \in \mathbb{R}^{n_1 \times n}$ and $b \in \mathbb{R}^{n_1}$. Namely, the system inequalities $f_i x \leq g_i$, $i = 1, \ldots, n_1$, where f_i are the rows of F and g_i are the elements of g. A Polyhedron class can be defined in ICT using the Multi-parametric toolbox [7] as,

```
>> P = Polyhedron(F,g);
```

specifying the half-space representation matrix and vectors F and g, respectively. Halfspaces and hyperplanes are convex sets, the intersection of convex sets is a convex set, and thus all polyhedrons are convex. A bounded polyhedron is a *polytope*. Given the polytope $\mathcal{P} = \{[x_1^{\mathsf{T}}\, x_2^{\mathsf{T}}] \in \mathbb{R}^{n_1+n_2} : F_{x_1}\, x_1 + F_{x_2}\, x_2 \leq g\} \subset \mathbb{R}^{n_1+n_2}$, the *projection* onto the x_1-space is defined as $\mathit{Proj}_{x_1} = \{x_1 \in \mathbb{R}^{n_1} : \exists\, x_2 \in \mathbb{R}^{n_2}$ such that $F_{x_1}\, x_1 + F_{x_2}\, x_2 \leq g\}$.

Problem formulation: Consider the following linear time-varying discrete-time system

$$\mathcal{S} : x(k+1) = A(k)x(k) + B(k)u(k), \tag{1}$$

where $x(\cdot) \in \mathbb{R}^n$ and $u(\cdot) \in \mathbb{R}^m$ are, respectively, the state and the control vector; $A(k) \in \mathbb{R}^{n \times n}$ and $B(k) \in \mathbb{R}^m$ are the state and control matrices. The family of time-varying matrices is characterised by polytopic uncertainty

$$A(k) = \sum_{l=1}^{q} \alpha^{(l)}(k) A^{(l)}, \quad B(k) = \sum_{l=1}^{q} \alpha^{(l)}(k) B^{(l)},$$

with $\sum_{l=1}^{q} \alpha^{(l)}(k) = 1$, where q is the number of realisations, and $\alpha^{(l)}(k)$, $l = 1, \ldots, q$ are unknown and time-varying non-negative constants. The matrices $A^{(l)}$ and $B^{(l)}$, $l = 1, \ldots, q$ are all given. We assume that the state x is measurable and available for feedback, and that a robustly

asymptotically stabilising state-feedback controller $u(k) = -Kx(k)$, exists, where $K \in \mathbb{R}^{m\times n}$ is a gain matrix.

The system (1) can be re-written as an input decentralised and interconnected system consisting of N subsystems,

$$\mathcal{S}_i : \begin{cases} x_i(k+1) = A_i(k)x_i(k) + B_i(k)u_i(k) \\ \qquad\qquad + \sum_{\substack{j=1 \\ j\neq i}}^{N} A_{ij}(k)x_j(k), \quad i \in \mathcal{N}, \end{cases} \tag{2}$$

where $x_i(\cdot) \in \mathbb{R}^{n_i}$ and $u_i(\cdot) \in \mathbb{R}^{m_i}$ are, respectively, the state and control vectors for the subsystem $i \in \mathcal{N} = \{1, 2, \ldots, N\}$; $A_i(k) \in \mathbb{R}^{n_i\times n_i}$ and $B_i(k) \in \mathbb{R}^{n_i\times m_i}$ are the state and control matrices; and, $A_{ij}(k) \in \mathbb{R}^{n_i\times n_j}$ is an *interconnection state matrix* between subsystem i and j. The overall system $\mathcal{S}$ is then $\mathcal{S} = \bigcup_{i\in\mathcal{N}} \mathcal{S}_i$ with state vector $x^\mathsf{T} = \begin{bmatrix} x_1^\mathsf{T} & x_2^\mathsf{T} & \cdots & x_N^\mathsf{T} \end{bmatrix} \in \mathbb{R}^n$ and control vector $u^\mathsf{T} = \begin{bmatrix} u_1^\mathsf{T} & u_2^\mathsf{T} & \cdots & u_N^\mathsf{T} \end{bmatrix} \in \mathbb{R}^m$, where $n = \sum_{i\in\mathcal{N}} n_i$ and $m = \sum_{i\in\mathcal{N}} m_i$. The state matrix $A(k) = A_D + A_C$, where A_D assembles the local dynamics of the subsystems and A_C assembles the couplings; $B(k)$ is the overall control matrix with block diagonal structure. For the input decentralised system, we assume that local state-feedback controllers

$$u_i(k) = -K_i x_i(k), \quad i \in \mathcal{N} \tag{3}$$

that robustly stabilise each subsystem $\mathcal{S}_i$, $i \in \mathcal{N}$, exist; where $K_i \in \mathbb{R}^{m_i\times n_i}$, $i \in \mathcal{N}$, are gain matrices.

The systems (1) and (2) can be coupled with bounded local state and control constraints

$$\begin{cases} x_i(k) \in \mathcal{X}_i,\ \mathcal{X}_i = \{x_i \in \mathbb{R}^{n_i} \mid F_{x_i}x_i \leq g_{x_i}\}, \\ u_i(k) \in \mathcal{U}_i,\ \mathcal{U}_i = \{u_i \in \mathbb{R}^{m_i} \mid F_{u_i}u_i \leq g_{u_i}\}, \end{cases} \tag{4}$$

$\forall\, k \geq 0$, $i \in \mathcal{N}$, where F_{x_i}, F_{u_i} are constant matrices and g_{x_i}, g_{u_i} are constant vectors of appropriate dimension with positive elements, and the origin is contained in the interior of the sets. The inequalities are component-wise. Then, the constraints for the overall system (1) are defined as $\mathcal{X} = \prod_{i=1}^{N} \mathcal{X}_i$ and $\mathcal{U} = \prod_{i=1}^{N} \mathcal{U}_i$.

To account for couplings between subsystems in (2), we convert the original system into a decoupled system [10]. To this end, we consider an interconnected dynamical system with additive norm-bounded disturbances given by $D_i(k)w_i(k) = \sum_{j=1,j\neq i}^{N} A_{ij}(k)x_j(k)$, where $w_i^\mathsf{T} = \begin{bmatrix} x_1^\mathsf{T} & \cdots & x_{i-1}^\mathsf{T} & x_{i+1}^\mathsf{T} & \cdots & x_N^\mathsf{T} \end{bmatrix} \in \mathbb{R}^{n-n_i}$ and $D_i = \begin{bmatrix} A_{i,1} & \cdots & A_{i,i-1} & A_{i,i+1} & \cdots & A_{i,N} \end{bmatrix}$, $i \in \mathcal{N}$, are the vector and matrix of interconnections, respectively. The couplings are brought to the general form of polytopic constraints,

$$w_i(k) \in \mathcal{W}_i,\ \mathcal{W}_i = \{w_i \in \mathbb{R}^{n_i} \mid F_{w_i}w_i \leq g_{w_i}\}, \tag{5}$$

$\forall\, k \geq 0$, $i \in \mathcal{N}$, where F_{w_i} and g_{w_i} are suitable [3], [4]. Finally, the interconnected system (2) can be re-written as:

$$x_i(k+1) = A_i(k)x_i(k) + B_i(k)u_i(k) + D_i(k)w_i(k), \tag{6}$$

for all $i \in \mathcal{N}$, subject to the state, control, and coupling constraints (4)–(5).

Definition of constraints: ICT defines the state, control and coupling constraints of (6) as cell arrays `A`, `B`, `D` from the state and control matrices `Ac`, `Bc` of (1) as follows,

```
>> [A, B, D] = cen2dec(Ac, Bc, n, m);
```

The user needs first to define the centralised matrices `Ac`, `Bc` as cell arrays, where each element of the array is a realisation for the matrix, e.g., `Ac{1}`, `Ac{2}`, `Ac{3}` is a the state matrix cell with 3 realisations $A^{(1)}$, $A^{(2)}$, and $A^{(3)}$; `n` and `m` contains the size of the local state and control vector. The output are the cell matrices `A{i,j}`, `B{i,j}`, `D{i,j}` of the interconnected systems (6), where `i`$\in \mathcal{N}$ and `j` defines the number of realisations. If the matrices of the interconnected systems are given, the routine `dec2cen` creates the matrices for the overall system (1).

Constraints are defined locally and brought to centralised form with `blkdiag` and `cat` MATLAB functions, e.g.,

```
>> % for each subsystem
>> X{i} = [Fx{i} gx{i}];
>> % constraints for overall system
>> Fxc = blkdiag(Fx{:});
>> gxc = cat(1,gx{:});
>> Xc = [Fxc gxc];
```

The overall control constraints `Uc` are obtained with a similar construction. Finally, disturbance constraints (5) for (6) are created with the following command

```
>> W = couplingconstraints(Fx,gx);
```

III. OFFLINE PROCESSING

Set invariance is important for IC to guarantee recursive feasibility and asymptotic stability of the closed-loop system. A useful toolbox (*invsetbox*) for computing invariant sets of LTI systems has been developed by Kerrigan [11], [9]. ICT extends *invsetbox* to account for constrained uncertain time-varying systems where the matrices are subject to polytopic uncertainty. Computations and manipulations over sets in ICT are carried out with the extended version of the invariant set toolbox. For a review and relevant methods in computational geometry, see e.g., [11], [12], [13].

A. Robust Invariant Sets

Given the state, control, and disturbance matrices for (6), we assume that for each subsystem i exists a decentralised state-feedback controller $u_i(k) = -K_i\, x_i(k)$, $i \in \mathcal{N}$, such that $\mathcal{S} = \bigcup_{i\in\mathcal{N}} \mathcal{S}_i$ is stable. The resulting closed-loop state matrix $A_i - B_iK_i, i \in \mathcal{N}$, is Schur. Robust linear state feedback control for LTV systems can be computed e.g. by YALMIP [14], which solves a semidefinite programming problem with linear matrix inequalities (LMI).

The following definitions applies to both the overall system (1) and the interconnected systems (6) with (4), (5).

Definition 3.1 (Robust Positively Invariant Set): Given the local controller (3) for each subsystem $i \in \mathcal{N}$ and $A_i^K = (A_i - B_iK_i)$, the set $\Omega_i \subseteq \mathcal{X}_i$ is a robust positively invariant constraint-admissible set with respect to $x_i(k+1) = A_i^K x_i(k) + D_i w_i(k)$ subject to the local constraints (4), (5), if and only if, $\forall\, x_i(k) \in \Omega_i$ and $\forall\, w_i(k) \in \mathcal{W}_i$, the system evolution satisfies $x_i(k+1) \in \Omega_i$ and $-K_i x_i(k) \in \mathcal{U}_i$, $\forall\, k \geq 0$.

The largest robust positively invariant set that respects constraints is called *Maximal Admissible Set* (MAS) [15]

and can be defined in polyhedral form as, $\Omega_i = \{x_i \in \mathbb{R}^{n_i} : F_i^0 x_i \leq g_i^0\}, \quad i \in \mathcal{N}$. The MAS characterises the high-gain inner controller, which satisfies some user-desired performance and guarantees system's overall stability.

To enlarge the domain of attraction of the controlled system, we define the *Robust Controllable Invariant Set*.

Definition 3.2 (Robust Controllable Invariant Set): Given the interconnected system (6) and the constraints (4), (5), the set $\Psi_i \subseteq \mathcal{X}_i$ is robust controllable invariant, if and only if, for all $x_i(k) \in \Psi_i$, there exists an admissible control $u_i(k) \in \mathcal{U}_i$ such that $x_i(k+1) \in \Psi_i$, $\forall\, i \in \mathcal{N}$, $\forall\, w_i(k) \in \mathcal{W}_i$, $\forall\, k \geq 0$.

The computation of the robust controllable invariant set can be computational prohibitive, in particular for the centralised system $\mathcal{S}$. To overcome this difficulty, ICT includes the option of computing the outer set as the M-step Robust Controllable Set or as MAS of some low-gain controller [5].

Definition 3.3 (M-step Robust Controllable Set): The set $P_i^M \subseteq \mathcal{X}_i$ is the set of all states for which exists an admissible control sequence such that the system (6) reaches the MAS Ω_i in no more than M steps along an admissible trajectory, i.e. one that satisfies (4), (5). The set P_i^M is called *M-step robust controllable set* and can be described by $P_i^M = \{x_i \in \mathbb{R}^{n_i} : F_i^M x_i \leq g_i^M\}, \quad i \in \mathcal{N}$.

B. Computing Invariant Sets

Table I shows the functions included in ICT for computing the invariant sets defined in Section III-A. Note that the procedures to compute the robust invariant sets will succeed if $\mathcal{W}_i$, $i \in \mathcal{N}$, is appropriately bounded [3], [4]. If any state x_j in (2) is free, a generous upper bound should be introduced to guarantee connective stability.

The function arguments are the system matrices, the local constraints, and the gain matrix `K`; `tmax` is the maximum number of steps allowed; `T` in `kinfset` is the target set. We consider the set of state constraints `X` as target set to compute the robust controllable invariant set. The outputs `Omega`, `Psi`, and `PM` are the invariant sets, `tstar` is the number of steps executed, and `fd` is a flag over the determinedness.

The invariant sets for the system (1) are computed with the same functions by omitting the disturbances with `[]`,

```
>> oinfsetcl(Ac,Bc,[],Xc,Uc,[],-Kc,tmax);
>> kinfset(Ac,Bc,[],Xc,Uc,[],Xc,tmax);
>> sinfset(Ac,Bc,[],Xc,Uc,[],Omegac,tmax);
```

Decentralised interpolating control for interconnected subsystems (6) demands for separable invariant sets, that is,

```
>> for i = 1:N
     [Omega{i},tstar,fd] = ...
      oinfsetcl(A{i,:},B{i,:},D{i,:},...
      X{i},U{i},W{i},-K{i},tmax)
   end
```

`kinfset` and `sinfset` are implemented likewise, for each $i \in \mathcal{N}$. Note that the output sets are given in minimal $\mathcal{H}$-representation, i.e., redundant inequalities were removed.

IV. ONLINE PROCESSING

Given an LTI or LTV system with polytopic uncertainty subject to local constraints (4), (5), the developed toolbox computes offline the required invariant sets and associated inner and outer control to enable the interpolation between them. The online processing involves the interpolation between the two controllers which is effectuated by solving iteratively (at each time instant) an LP problem. ICT implements the centralised interpolating control as in [1], [5] as well as some variations and improved formulations [16], [17]. For the decentralised control of interconnected systems ICT implements the dIC proposed by the authors in [3], [4].

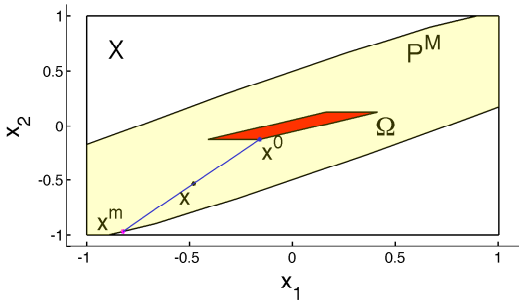

Fig. 1: Any state $x(k)$ can be decomposed as a convex combination of $x^0(k) \in \Omega$ and $x^m(k) \in P^M$.

A. Centralised Interpolating Control

ICT implements the improved interpolating control proposed in [5]. The improved interpolating control determines the global outer controller in an augmented state and control space, and thus no vertex representation [18] of the controllable invariant set is needed as in [2].

Consider the input centralised constrained system (1)–(4). It is assumed that an outer invariant set Φ is available (either a low-gain MAS, the M-step invariant set, or the controllable invariant set). Consider the extended state-space $Q^M = \{x \in \mathbb{R}^n, u^1 \in \mathbb{R}^m, u^2 \in \mathbb{R}^m, \ldots, u^M \in \mathbb{R}^m, \}$ such that x can be steered in M-steps inside the outer set along an admissible trajectory. The $\mathcal{H}$-representation of Q^M is,

$$Q^M = \left\{ x \in \mathbb{R}^n, U^M \in R^{Mm} \colon \bar{F}^M \begin{bmatrix} x \\ U^M \end{bmatrix} \leq \bar{g}^M \right\},$$

where U^M is the vector of admissible control sequences $u^i \in U$, $i = 1, \ldots, M$. Any state $x(k)$ can be decomposed as,

$$x(k) = s(k)\, x^m(k) + (1 - s(k))\, x^0(k), \tag{7}$$

where x^0 is in the inner high-gain MAS Ω, x^m is so that a control sequence exists such that $[x^{m\mathsf{T}}\ U_M^{\mathsf{T}}]^{\mathsf{T}} \in Q^M$, and $s(k) \in [0, 1]$ is the interpolating coefficient. Fig. 1 illustrates the interpolation concept in a two-dimensional state space $\mathcal{X}$, where the set P^M, i.e. the projection of Q^M onto the x-space, is depicted in yellow and the MAS Ω is depicted in red. The control law is decomposed as

$$u(k) = s(k)\, u^1(k) + (1 - s(k))\, u^0(k), \tag{8}$$

where $u^0(k) = -K\, x^0(k)$ and $u^1(k)$ is the first control of the sequence U_M. The interpolating controller is then computed by solving the following LP problem at each time step k

TABLE I: Functions included in ICT to compute invariant sets for LTI and LTV systems.

Function	Description
`[Omega,tstar,fd] = oinfsetcl(A,B,D,X,U,W,-K,tmax)`	Calculates the maximal robust positively invariant set
`[Psi,tstar,fd] = kinfset(A,B,D,X,U,W,T,tmax)`	Calculates the robust controllable invariant set
`[PM,tstar,fd] = sinfset(A,B,D,X,U,W,Omega,tmax)`	Calculates the M-step robust controllable invariant set

(index k is omitted for clarity)

$$s^*(x) = \min_{s, r^m, V^M} s$$
$$\text{subject to: } \begin{cases} s(1-g^0) - F^0 r^m \leq -F^0 x \\ \bar{F}^M \begin{bmatrix} r^m & V^M \end{bmatrix}^{\mathsf{T}} \leq s\bar{g}^M \\ 0 \leq s \leq 1 \\ v^1, \ldots, v^M \in s\,U \end{cases}, \quad (9)$$

where $r^m = s\,x^m$, $V^M = s\,U^M$, and $v^i = s\,u^i$ for $i = 1, \ldots, M$, is the change of variable implemented in order to obtain a linear formulation of the optimisation problem [5]. Note that the process takes into account the constraints that are verified at each time step. We can then compute the centralised interpolating control of (1) with the ICT by,

```
>> [x, u, s] = cIC(A, B, -K, X, U,...
   Omega, Phi, M, Nsteps, x0, alphas);
```

The outputs are the `Nsteps` trajectories of the state, the cIC `u`, and the interpolating coefficient `s` for the initial condition `x0`; and the $(q \times$ `Nsteps`$)$-`alphas` realisations previously defined by the user. `x` and `u` are matrices of dimension $n \times$ `Nsteps` and $m \times$ `Nsteps`, respectively; `Omega` and `Phi` are matrices and vectors that define the half-space representations of the inner and outer sets. The control law (7), (8), (9) guarantees recursive feasibility and asymptotic stability for all $x \in Conv\{\Omega, P^M\}$ [5]. ICT allows to check whether the initial state `x0` is in a feasible set `P` using MPT:

```
>> P.contains(x0);
```

where `P` is the convex hull of the inner and outer invariant sets defined as `P = Polyhedron(F,g)`, where `F` and `g` are the matrix and the vector of the half-space representation.

B. Robust Decentralised Interpolation-based Control

Decentralised interpolating control (dIC) for LTI and LTV uncertain systems has been developed in [3], [4]. dIC computes separable robust controllable invariant sets for local control design, which overcomes the computational burden of large-scale systems and cIC. Provided the availability of separable invariant sets, the interpolation concept presented in the previous section can then be extended to interconnected systems with local constraints (2), (4), (5).

For any subsystem, a robust maximal admissible set Ω_i is computed for a given feedback control high-gain matrix K_i, $\forall\, i \in \mathcal{N}$. This represents the inner MAS, i.e., the set of the states that can be steered to the origin with some user-desired performance specification. The outer set Φ_i for each subsystem is defined as the robust controllable invariant set Ψ_i, $\forall\, i \in \mathcal{N}$, or as the M-step robust controllable set if M is maximal, i.e., if $P_i^{M+1} = P_i^M$, $\forall\, i \in \mathcal{N}$. Similarly to [5], if the maximal robust controllable invariant set or the M-step robust controllable set cannot be determined, a low-gain robust maximal admissible set can be considered as outer invariant set.

Suppose now that any known state $x_i(k) \in \Phi_i$ can be decomposed as follows

$$x_i(k) = s_i(k) x_i^m(k) + (1 - s_i(k)) x_i^0(k), \quad i \in \mathcal{N}, \quad (10)$$

where $x_i^0(k) \in \Omega_i$ and $x_i^m(k)$ is such that there exists a control $u_i^1(k) \in \mathcal{U}_i$ defined in the outer set such that $A_i(k)x_i^m(k) + B_i(k)u_i^1(k) + D_i(k)w_i(k) \in \Phi_i$, $\forall\, w_i \in \mathcal{W}_i$; and $s_i(k) \in [0, 1]$ is the interpolating coefficient. Similarly, the control in each subsystem is decomposed as follows

$$u_i(k) = s_i(k) u_i^1(k) + (1 - s_i(k)) u_i^0(k), \quad i \in \mathcal{N}, \quad (11)$$

where $u_i^0(k) = -K_i^0 x_i^0(k)$ is the inner stabiliser controller (3) of each subsystem $\mathcal{S}_i$, $i \in \mathcal{N}$, and u_i^1 is the outer control.

The interpolating control is obtained by solving the following LP problem for each subsystem $i \in \mathcal{N}$ at each discrete time k (index k is omitted for clarity) [4]:

$$s_i^*(x_i) = \min_{s_i, r_i^m, v_i^1} s_i, \quad (12)$$
$$\begin{cases} s_i\, g_i^0 - F_i^0\, r_i^m \leq g_i^0 - F_i^0 x_i \\ F_i^1 \big(A_i^{(l)}\, r_i^m + B_i^{(l)}\, v_i^1\big) \leq s_i \Big(g_i^1 - \max\limits_{w_i^{(l)} \in \mathcal{W}_i} F_i^1 D_i^{(l)} w_i^{(l)}\Big) \\ 0 \leq s_i \leq 1, \quad v_i^1 \in s_i\, \mathcal{U}_i \end{cases}$$

where the second inequality holds for $l = 1, \ldots, q$, $i \in \mathcal{N}$; F_i^1 and g_i^1 define the half-space representation of Φ_i. A change of variables is applied to a bilinear optimisation problem: $r_i^m = s_i\, x_i^m$, and $v_i^1 = s_i\, u_i^1$. Note that $x_i^0 \in \Omega_i$ can be recovered from the equality $s_i\, x_i^0 = x_i - s_i\, x_i^m$. ICT computes the interpolation-based control for the interconnected systems (6) subject to constraints (4), (5) with q-`alphas` realisations over `Nsteps` time steps with input fields that specify the system matrices `A`, `B`, `D`, high-gain matrix `K`, local contraints `X`, `U`, `W`, and invariant sets `Omega` and `Phi`:

```
>> [x, u, s] = dIC(A, B, D, -K, X, U,...
   W, Omega, Phi, Nsteps, x0, alphas);
```

Each row $i \in \mathcal{N}$ of `s` contains the local interpolating coefficients over time. The coefficients s_i are non-increasing Lyapunov functions, and thus guarantee the stability of the system. Stability and recursive feasibility is guaranteed for the initial state $x_i^0 \in \Phi_i$ [4]. Compared to `cIC`, `dIC` solves on-line a low-dimensional LP problem for each subsystem at each time step and guarantees better exploitation of the signal space with a fast convergence to the MAS. Furthermore, the LP problem for decentralised interpolating control is less computationally expensive compared to the overall interpolating scheme and appropriate for hardware-embedded or real-time control of large-scale systems.

Remark 1: IC considers the realisations of the state and control matrices when computing the invariant sets. If the system is LTI, i.e., the matrices are constant and $q = 1$, the computation of the invariant sets do not need to consider the uncertainty in the matrices, so it is less expensive. In this case, the second inequality of the LP (12) holds for $l = 1$.

V. VISUALISATION

ICT includes some plotting routines to display the outputs of the control. Particularly, ICT plots the state and control trajectories, and the interpolating coefficient over time by:

```
>> plotX(x);  plotU(u);  plotS(s);
```

A polyhedron $\mathcal{P} = \{x \colon Fx \leq g\}$ can be defined and plotted using the Multi-Parametric Toolbox [7] as follows:

```
>> P = Polyhedron(F,g);  P.plot;
```

Note that `plot` routine works only with objects of up to $\mathbb{R}^3$-space. Another interesting plotting function is `plotEvo`. It plots the outer and inner invariant sets and draws the state evolution over time as follows:

```
>> [x, u, s, r0, v1] = cIC(A, B, -K, X, U,...
   Omega, Phi, M, Nsteps, x0, alphas);
>> plotEvo(A,B,Omega,Phi,x,r0,v1,s);
```

This function can be used in $\mathbb{R}^2$ where the user can observe the interpolation between the states x^0 and x^m over time.

VI. EXAMPLES

This section provides a general example to implement robust centralised interpolating control [5] and decentralised interpolating control for interconnected systems [4]. Further numerical examples can be found in [1], [5] for centralised IC, and in [3], [4] for decentralised robust control.

A. Pseudocode for Computing dIC

Consider an interconnected system (2) consisting of 2 subsystems with 2 states each and with $q = 2$ realisations; coupled with local state and control constraints (4). State and control matrices are time-varying as follows,

$$\begin{aligned}
\mathcal{S}_1 : x_1(k+1) =& A_1(k)x_1(k) + B_1(k)u_1(k) + A_{1,2}(k)x_2(k)\\
\mathcal{S}_2 : x_2(k+1) =& A_2(k)x_2(k) + B_2(k)u_2(k) + A_{2,1}(k)x_1(k)
\end{aligned}$$

$$\begin{cases}
A_i(k) =& \alpha^{(1)}(k)A_i^{(1)} + \left(1-\alpha^{(1)}(k)\right)A_i^{(2)},\\
B_i(k) =& \alpha^{(1)}(k)B_i^{(1)} + \left(1-\alpha^{(1)}(k)\right)B_i^{(2)},\\
A_{ij}(k) =& \alpha^{(1)}(k)A_{ij}^{(1)} + \left(1-\alpha^{(1)}(k)\right)A_{ij}^{(2)},
\end{cases}$$

with $i, j = 1, 2$ and $j \neq i$; $A_i^{(l)}$, $B_i^{(l)}$, and $A_{i,j}^{(l)}$ for $i, j = 1, 2$, $j \neq i$, are given for each $l = 1, 2$ realisation. Note that for $q = 2$, $\alpha^{(2)}(k) = 1 - \alpha^{(1)}(k)$. Matrices and vectors of interconnections of the subsystems $\mathcal{S}_1$ and $\mathcal{S}_2$ are defined as $D_1(k) = A_{1,2}(k)$ and $w_1 = x_2$, and $D_2(k) = A_{2,1}(k)$ and $w_2 = x_1$, respectively. Then, the input centralised system (1) has state and control matrices $A(k) = \alpha^{(1)}A^{(1)} + \left(1-\alpha^{(1)}\right)A^{(2)}$ and $B(k) = \alpha^{(1)}B^{(1)} + \left(1-\alpha^{(1)}\right)B^{(2)}$ with

$$A^{(l)} = \begin{bmatrix} A_1^{(l)} & A_{1,2}^{(l)} \\ A_{2,1}^{(l)} & A_2^{(l)} \end{bmatrix} \quad B^{(l)} = \begin{bmatrix} B_1^{(l)} & 0 \\ 0 & B_2^{(l)} \end{bmatrix} \; l = 1, 2.$$

State and control constraints are defined as $x_i \in \mathcal{X}_i$, $u_i \in \mathcal{U}_i$, for $i = 1, 2$. Then, the coupling constraints are $w_1 \in \mathcal{X}_2$ and $w_2 \in \mathcal{X}_1$, that is, $\mathcal{W}_1 = \mathcal{X}_2$ and $\mathcal{W}_2 = \mathcal{X}_1$. An initial state x_0 which belongs to the outer sets is defined.

Algorithm 1 outlines the main steps for computing the dIC of an interconnected system with N subsystems and q realisations. It computes the required robust controlled invariant sets for each subsystem and solves an LP problem on-line for each subsystem at each time step. dIC is an admissible control action for weakly interconnected systems. A system with strong couplings calls for centralised control. Centralised IC for the overall system can be then computed similarly to Algorithm 1.

Algorithm 1: dIC for interconnected system (2)

input : Matrices `A{i,l}`, `B{i,l}`, `D{i,l}`, `K{i}`, the sets `X{i}`, `U{i}`, for `i = 1,...,N` and `l = 1,...,q`.
output: `x, u, s`.

1 **Compute** the disturbance constraints `W` with `couplingconstraints(Fx,gx)`
2 **for** $i \leftarrow 1$ **to** N **do**
  Compute of the sets `Omega{i}` and `Psi{i}` with `oinfsetcl` and `kinfset`;
  end
3 **Define** the random alpha realisations `alphas` and the number of steps `Nsteps`;
4 **Define** the initial state `x0` that belongs to the outer sets;
5 **Compute** the decentralised interpolating control with `dIC`;
6 **Plot** the state evolution: `plotX(x)`;
7 **Plot** the control evolution: `plotU(u)`;
8 **Plot** the interpolating coefficient: `plotS(s)`;

B. Numerical example

Consider a constrained LTV system with two state variables and one control variable [5]. The state matrices are time-varying and have two realisations ($q = 2$) given by,

$$A(k) = \alpha(k)A^{(1)} + (1-\alpha(k))A^{(2)}$$

$$A^{(1)} = \begin{bmatrix} 1 & 0.1 \\ 0 & 0.99 \end{bmatrix}, \quad A^{(2)} = \begin{bmatrix} 1 & 0.1 \\ 0 & 0 \end{bmatrix},$$

with $\alpha(k) \in [0, 1]$, and the control matrix is constant and equal to $B = [0 \quad 0.0787]^{\mathsf{T}}$. State and control variable are subject to constraints $|x_1| \leq 1, |x_2| \leq 1, |u_1| \leq 2$. In ICT we can define the system matrices and constraints as:

```
>> A{1} = [1 0.1; 0 .99]; A{2} = [1 0.1; 0 0];
>> B{1} = [0 0.0787]';  B{2} = [0 0.0787]';
>> K = [30.3781 9.6139];
>> % State and control constraints
>> Fx = [eye(2); -eye(2)]; gx = ones(4,1);
>> X = [Fx gx];
>> Fu = [1; -1]; gu = [2; 2];  U = [Fu gu];
```

The MAS Ω is then computed with respect to the high-gain matrix $K = [30.3781 \quad 9.6139]$ and the constraints above. The outer set is computed as the maximal M-step controllable set P^M with $M = 66$.

```
>> % Maximal Admissible invariant Set
>> tmax = 20; % max number of steps
>> [Omega,tstar,fd] = oinfsetcl(A,B,[],X,...
    U,[],-K,tmax);
>> % M-step controlled invariant set
>> tmax = 66; % max number of steps
>> [PM,tstar,fd] = sinfset(A,B,[],X,U,[],...
    Omega,tmax);
```

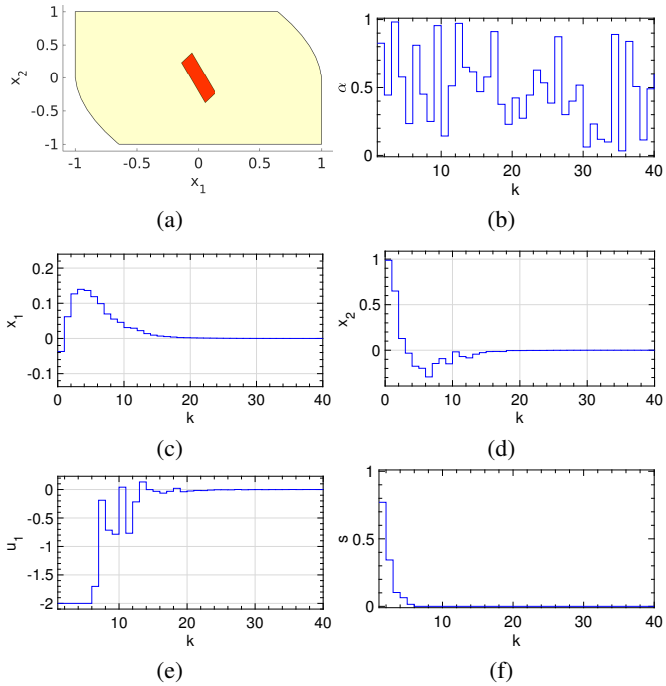

Fig. 2: (a): Invariant sets. The yellow set is the outer M-step contrallable set P^{66}; the red set is the MAS for the feedback controller $u = -Kx$; (b): $\alpha(k)$ realisations; (c), (d), (e): state and control trajectories; (c): Interpolating coefficient for cIC.

Figure 2(a) shows the computed invariant sets that are computed in 0.6 seconds and 43 seconds, respectively. Note that in general the maximal P^M is not equal to the maximal robust controllable invariant set Ψ. $\Psi \setminus P^M$ is the set of the states that will remain inside Ψ but will not converge to the MAS. However, in this particular example $\Psi \equiv P^M$. cIC is computed for $N = 40$ time steps using:

```
>> Nsteps = 40;
>> M = 1;
>> x0 = [-.037; 0.9875];
>> % define a random alpha vector
>> alphas = rand(N,1);
>> [x, u, s] = cIC(A, B, -K, X, U, ...
    Omega, PM, M, Nsteps, x0, alphas);
```

Figs 2(c)–2(e) depict the state and control trajectories computed with cIC for the initial state $x_0 = [-0.0370\ 0.9875]^\mathsf{T}$ and alpha realisations over time as in Fig. 2(b). The M-step extended state space Q^M is computed with $M = 1$. cIC stabilises the LTV system subject to constraints around the origin in less than 1 second. It steers the system into the MAS in 6 steps (see Fig. 2(f)). The interpolating coefficient over time is positive and decreasing, so guarantees stability.

VII. CONCLUSIONS

At the present time, ICT is the first toolbox that implements interpolation-based control for centralised and decentralised control systems. It includes the invariant set toolbox [9] that is extended by the authors to account for linear time-varying systems. ICT is a standalone toolbox and available to download from `https://ictoolbox.github.io`. Its user's manual will be available in the near future on the same website. The use of MPT [7] with ICT is optional and only required for plotting invariant sets.

The ICT has been tested on a Debian-based Linux distribution Mint 17.3 Rosa and on Mac OS X with MATLAB R2016b and R2014b, respectively. To give an idea of the performance of ICT, consider the example in [4]. The overall system is a LTV system with 6 states, 3 inputs and 2 realisations ($q = 2$), and can be decomposed in an interconnected system with 3 subsystems of 2 states and 1 control each. ICT for centralised systems computes the invariants sets in 36.36 CPU-seconds, whereas the three low-dimension invariant sets are obtained in 11.24 CPU-seconds. `cIC` and `dIC` compute the control in 0.63 CPU-seconds and 0.57 CPU-seconds, respectively for `Nsteps` $= 15$ time steps with 3.10GHz Quad-core Intel i7-3770S. The computational improvements of dIC will be likely to be much higher for large-scale systems that can be appropriately decomposed into interconnected systems with distinct controls.

REFERENCES

[1] H.-N. Nguyen, *Constrained Control of Uncertain, Time-Varying, Discrete-Time Systems*. Cham, Switzerland: Springer, 2014.

[2] H.-N. Nguyen, P.-O. Gutman, S. Olaru, and M. Hovd, "Implicit improved vertex control for uncertain, time-varying linear discrete-time systems with state and control constraints," *Automatica*, vol. 49, no. 9, pp. 2754–2759, 2013.

[3] S. Scialanga and K. Ampountolas, "Interpolating constrained control of interconnected systems," *IFAC-PapersOnLine*, vol. 51, no. 9, pp. 7–12, 2018.

[4] ——, "Robust constrained interpolating control of interconnected systems," in *Proc. 57th IEEE Conference on Decision and Control*, 2018, pp. 7016–7021.

[5] H.-N. Nguyen, P. O. Gutman, and R. Bourdais, "More efficient interpolating control," in *Proc. 2014 European Control Conference (ECC)*, 2014, pp. 2158–2163.

[6] A. Bemporad, "Hybrid Toolbox - User's Guide," 2004, http://cse.lab.imtlucca.it/~bemporad/hybrid/toolbox.

[7] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *2013 European Control Conference (ECC)*, 2013, pp. 502–510, http://control.ee.ethz.ch/~mpt.

[8] S. Riverso, M. Farina, and G. Ferrari-Trecate, "Plug-and-play decentralized model predictive control for linear systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 10, pp. 2608–2614, 2013.

[9] E. Kerrigan, "Invariant set toolbox for matlab," 2003. [Online]. Available: http://www-control.eng.cam.ac.uk/eck21/matlab/invsetbox/

[10] D. D. Šiljak, *Decentralised control of complex systems*. New York: Academic Press, Inc., 1991.

[11] E. C. Kerrigan, "Robust constraint satisfaction: Invariant sets and predictive control," Ph.D. thesis, University of Cambridge, UK, 2000.

[12] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, Cambridge, 2017.

[13] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control*. Birkhäuser Basel, 2008.

[14] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proc. of the CACSD Conference*, Taipei, Taiwan, 2004.

[15] E. G. Gilbert and K. T. Tan, "Linear systems with state and control constraints: the theory and application of maximal output admissible sets," *IEEE Transactions on Automatic Control*, vol. 36, no. 9, pp. 1008–1020, 1991.

[16] H.-N. Nguyen, P.-O. Gutman, S. Olaru, and M. Hovd, "Robust optimization-based control of constrained linear discrete time systems with bounded disturbances," *IFAC Proceedings Volumes*, vol. 46, no. 2, pp. 917–922, 2013.

[17] D. Rubin, P. Mercader, H.-N. Nguyen, P.-O. Gutman, and A. Baños, "Improvements on interpolation techniques based on linear programming for constrained control," in *20th IFAC World Congress*, 2017, pp. 1403–1408.

[18] P. O. Gutman and M. Cwikel, "Admissible sets and feedback control for discrete-time linear dynamical systems with bounded controls and states," *IEEE Transactions on Automatic Control*, vol. 31, no. 4, pp. 373–376, 1986.