Differential Games for Compositional Handling of Competing Control Tasks

Joshua Shay Kricheli [1] Shai Arogeti [2] Gera Weiss [1]

[1]Department of Computer Science, BGU [2]Department of Mechanical Engineering, BGU

Abstract

We present a novel Divide and Conquer (D&C) design methodology for separating the handling of competing objective in controllers. By guarantying a Nash Equilibrium in a virtual game, we synthesize a controller that establishes a balance between the objectives and allows the control engineer a disciplined method for tuning the parameters along the design cycle.

Continuous Linear-Time-Invariant (LTI) System

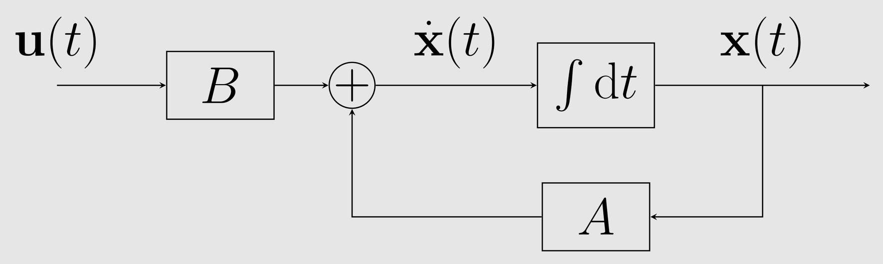
A continuous dynamical system $\mathcal{S}$, with state and input vectors $\mathbf{x}(t) \in \mathcal{X} \subseteq \mathbb{R}^n$ and $\mathbf{u}(t) \in \mathcal{U} \subseteq \mathbb{R}^m$, defined for all $t \in \mathcal{I} = [t_0, T_f] \subseteq \mathbb{R}$, is **LTI** if it is governed by the following first order differential equation, which is called its **state space representation**:

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t)$$

with a given initial condition $\mathbf{x}(t_0) \coloneqq \mathbf{x_0} \in \mathcal{X}$ and possibly a desired state (in case of signal tracking) $\mathbf{x}(T_f) \coloneqq \mathbf{x_T} \in \mathcal{X}$ where:

- $A \in \mathbb{R}^{n \times n}$ is a time-invariant matrix representing the **unforced dynamics** of $\mathcal{S}$;
- $B \in \mathbb{R}^{n \times m}$ is the time-invariant **input coefficients** matrix of $\mathcal{S}$.

$\mathcal{S}$ can be illustrated by the following block diagram:



Continuous Virtually Decomposed LTI System

Given a continuous LTI system $\mathcal{S}$, and $N \in \mathbb{N}$, an equivalent **virtually decomposed LTI system**, denoted $\mathcal{S}_N$ is defined as an LTI system that adheres the following model:

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + \sum_{i=1}^{N} B_i \mathbf{v_i}(t)$$

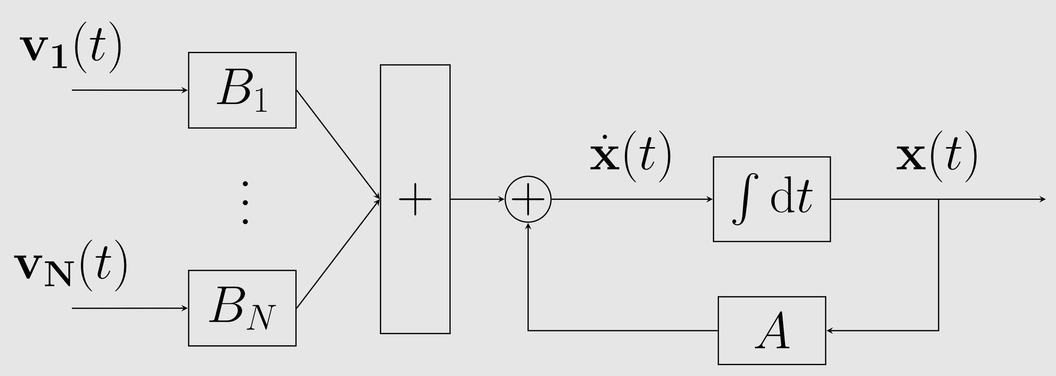
while satisfying:

$$B\mathbf{u}(t) = \sum_{i=1}^{N} B_i \mathbf{v_i}(t)$$

and where:

- $(\mathbf{v_i}(t))_{i=1}^N$ are **virtual input vectors**, with each being a vector of some length $m_i \in \mathbb{N}$ such that $\mathbf{v_i}(t) \in \mathcal{V}_i \subseteq \mathbb{R}^{m_i}$. We denote $\mathcal{V} \coloneqq \bigtimes_{i=1}^N \mathcal{V}_i \subseteq \mathbb{R}^{\sum_{i=1}^N m_i}$;
- $(B_i)_{i=1}^N$ are **virtual input coefficients** matrices where each $B_i \in \mathbb{R}^{n \times m_i}$.

$\mathcal{S}_N$ can be illustrated by the following block diagram:



Open-Loop Nash Equilibrium

Given a decomposed system $\mathcal{S}_N$, its **virtual quadratic cost functions** are a series of functions $(J_i)_{i=1}^N$, where each $J_i \colon \mathcal{X} \times \mathcal{V} \times \mathcal{I} \to \mathbb{R}^+$ assumes the following form:

$$J_i\Big(\mathbf{x}(t), (\mathbf{v_j}(t))_{j=1}^N, t\Big) \coloneqq \int_t^{T_f} \Big[\mathbf{x}^T(\tau) Q_i \mathbf{x}(\tau) + \sum_{j=1}^{N} \mathbf{v_j}^T(\tau) R_{ij} \mathbf{v_j}(\tau)\Big] \mathrm{d}\tau + \mathbf{x}^T(T_f) Q_{f_i} \mathbf{x}(T_f),$$

where:

- $(Q_i)_{i=1}^N$ are **virtual state weights**, each $Q_i \in \mathbb{R}^{n \times n}$ is a positive semi-definite matrix;
- $\Big((R_{ij})_{j=1}^N\Big)_{i=1}^N$ are **virtual input weights**, each $R_{ij} \in \mathbb{R}^{m_j \times m_j}$ is a positive definite matrix.

With that, a series of virtual policies $(\mathbf{v_j^*}(t))_{j=1}^N$ is said to constitute an **Open-Loop Nash Equilibrium** if for all $1 \leq i \leq N$:

$$J_i\Big(\mathbf{x}(t), (\mathbf{v_j^*}(t))_{j=1}^N, t\Big) \leq J_i\Big(\mathbf{x}(t), (\mathbf{v_j^*}(t))_{\substack{j=1 \\ j \neq i}}^N, \mathbf{v_i}(t), t\Big)$$

for any other admissable virtual input $\mathbf{v_i}(t) \in \mathcal{V}_i$ and for all $t \in \mathcal{I}$. In words, a series of virtual inputs $(\mathbf{v_j^*}(t))_{j=1}^N$ constitutes an Open-Loop Nash equilibrium if it is not possible to decrease any cost function $J_i(t)$ only by changing its corresponding virtual input $\mathbf{v_i^*}(t)$ to some other input $\mathbf{v_i}(t)$.

Nash Equilibrium Solution

The Open-Loop Nash Equilibrium problem is solved by closed-loop feedback policies $(\mathbf{v_i^*}(t))_{i=1}^N$ of the following form:

$$\mathbf{v_i^*}(t) \coloneqq -R_{ii}^{-1} B_i^T P_i^*(t) \mathbf{x}(t),$$

where the matrices $(P_i^*(t))_{j=1}^N$ are a solution to the following set of coupled matrix differential Riccati equations:

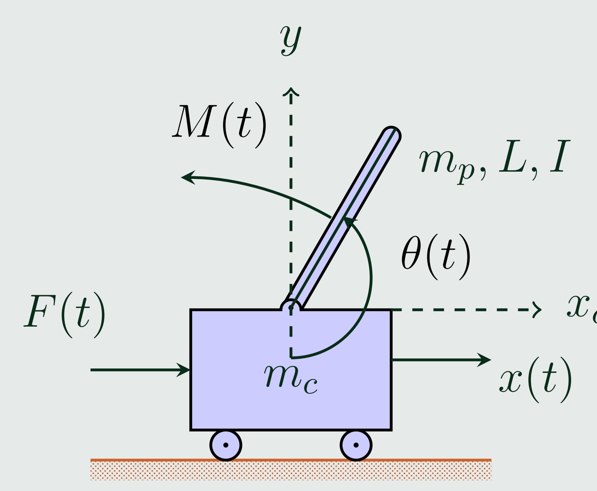
$$\dot{P}_i(t) + A_{cl}^T(t) P_i(t) + P_i(t) A_{cl}(t) + Q_i + \sum_{j=1}^{N} P_j(t) B_j R_{jj}^{-1} R_{ij} R_{jj}^{-1} B_j^T P_j(t) = 0,$$

with $A_{cl}(t) \coloneqq A - \sum_{j=1}^N B_j R_{jj}^{-1} B_j^T P_j(t)$, along with appropriate terminal conditions of the form:

$$(P_i(T_f))_{i=1}^N \equiv (Q_{f_i})_{i=1}^N$$

Inverted Pendulum

Consider the following system termed $\mathcal{S}_{IP}$:



with the following linearized mode:

$$\underbrace{\begin{bmatrix} \dot{x}(t) \\ \dot{\theta}(t) \\ \ddot{x}(t) \\ \ddot{\theta}(t) \end{bmatrix}}_{\dot{\mathbf{x}}(t)} = \underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{m_p^2 g L^2}{m_c m_p L^2 + 4I(m_c+m_p)} & 0 & 0 \\ 0 & \frac{4 m_p g l (m_c+m_p)}{m_c m_p L^2 + 4I(m_c+m_p)} & 0 & 0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x(t) \\ \theta(t) \\ \dot{x}(t) \\ \dot{\theta}(t) \end{bmatrix}}_{\mathbf{x}(t)} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{m_p L^2 + 4I}{m_c m_p L^2 + 4I(m_c+m_p)} & \frac{2 m_p L}{m_c m_p L^2 + 4I(m_c+m_p)} \\ \frac{2 m_p L}{m_c m_p L^2 + 4I(m_c+m_p)} & \frac{4(m_c+m_p)}{m_c m_p L^2 + 4I(m_c+m_p)} \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} F(t) \\ M(t) \end{bmatrix}}_{\mathbf{u}(t)}.$$

We decompose the system using the following virtual inputs:

$$v_x(t) \coloneqq \frac{m_p L^2 + 4I}{m_c m_p L^2 + 4I(m_c+m_p)} F(t) + \frac{2 m_p L}{m_c m_p L^2 + 4I(m_c+m_p)} M(t)$$

$$v_\theta(t) \coloneqq \frac{2 m_p L}{m_c m_p L^2 + 4I(m_c+m_p)} F(t) + \frac{4(m_c+m_p)}{m_c m_p L^2 + 4I(m_c+m_p)} M(t).$$

into the following decomposed system $\mathcal{S}_{IP_{x,\theta}}$:

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}}_{B_x} v_x(t) + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{B_\theta} v_\theta(t)$$

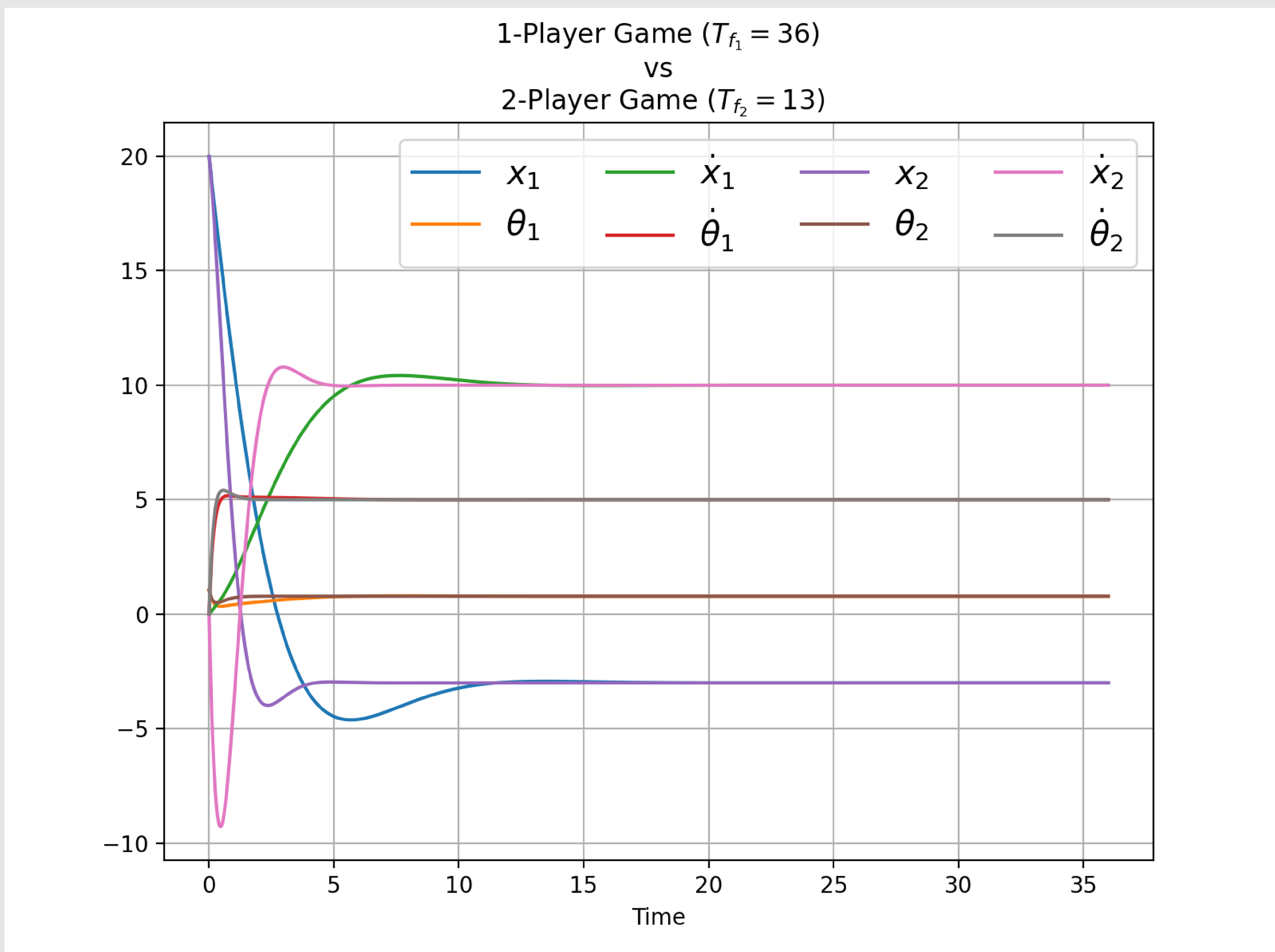
Signal tracking simulation results:

$$\mathbf{x_0} \coloneqq \begin{bmatrix} x_0 \\ \theta_0 \\ \dot{x}_0 \\ \dot{\theta}_0 \end{bmatrix} = \begin{bmatrix} 20\ [m] \\ \pi/3\ [rad] \\ 0 \\ 0 \end{bmatrix} \quad ; \quad \mathbf{x_T} \coloneqq \begin{bmatrix} -3\ [m] \\ \pi/4\ [rad] \\ 10\ [m/s] \\ 5\ [rad/s] \end{bmatrix} \tag{1}$$

Convergence times T_c (time to achieve the condition: $\|\mathbf{x}(t) - \mathbf{x_T}\| < 10^{-5}$) of $\mathcal{S}_{IP}$ and $\mathcal{S}_{IP_{x,\theta}}$ for different m_c, m_p, L values:

T_c $[sec]$ / m_c $[kg]$, m_p $[kg]$, L $[m]$	$\mathcal{S}_{IP}$	$\mathcal{S}_{IP_{x,\theta}}$
20, 5, 2	14	13
50, 8, 3	20	13
100, 10, 4	27	13
200, 15, 1	36	13
1000, 40, 8	74	13
2000, 60, 20	111	13

Comparing State Space Variables with $(m_c, m_p, L) \equiv (200, 15, 1)$



A Proof-of-Concept Python Library

A tool that supports the proposed methodology is openly accessible at `https://github.com/krichelj/PyDiffGame`. The tool uses a novel approach for solving Algebraic Riccati Equations for computing equilibria of differential games.

 krichelj@post.bgu.ac.il