International Graduate School on Control, Stuttgart, May 2024

Dissipation Inequalities and Quadratic Constraints for Control, Optimization, and Learning

Lesson 5: Polynomial and Time-Varying Dynamics

Murat Arcak[1] and Peter Seiler[2]

[1] University of California, Berkeley

[2] University of Michigan, Ann Arbor

Learning Objectives

In this lesson we will

- Discuss the use of sum-of-squares SOS optimizations for constructing Lyapunov and storage functions for uncertain polynomial systems.
- Describe the generalization of the dissipation inequality / IQC conditions for uncertain systems with time-varying nominal dynamics.
- Present the corresponding dissipation inequality / IQC results for the discrete-time systems.

Outline

1. Sum-of-squares SOS
2. Time-varying results (LTV)
3. Nonlinear reachability analysis
4. Discrete-time results

Region of Attraction (ROA)

Consider the autonomous nonlinear dynamical system

$$\dot{x}(t) = f(x(t))$$

where $x(t) \in \mathbb{R}^n$ is the state at time t and $f: \mathbb{R}^n \to \mathbb{R}^n$. Assume:

- $f(0) = 0$, i.e. $x = 0$ is an equilibrium point.
- $x = 0$ is an asymptotically stable equilibrium point.
- f is a polynomial function of x.

Define the region of attraction (ROA) as:

$$\mathcal{R} := \{\xi \in \mathbb{R}^n \ : \ \lim_{t \to \infty} \phi(\xi, t) = 0\}$$

where $\phi(\xi, t)$ denotes the solution at time t starting from the initial condition $\phi(\xi, 0) = \xi$.

Objective: Compute or estimate the ROA.

We will show how to perform this computation using sum-of-squares (SOS) optimization.

Polynomials

- Given $\alpha \in \mathbb{N}^n$, a monomial in n variables is a function $m_\alpha\colon \mathbb{R}^n \to \mathbb{R}$ defined as $m_\alpha(x) := x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$.
- The degree of a monomial is defined as $\deg m_\alpha := \sum_{i=1}^n \alpha_i$.
- A polynomial in n variables is a function $p\colon \mathbb{R}^n \to \mathbb{R}$ defined as a finite linear combination of monomials:

$$p := \sum_{\alpha \in \mathcal{A}} c_\alpha m_\alpha = \sum_{\alpha \in \mathcal{A}} c_\alpha x^\alpha$$

where $\mathcal{A} \subset \mathbb{N}^n$ is a finite set and $c_\alpha \in \mathbb{R}\ \forall\ \alpha \in \mathcal{A}$.

- The set of polynomials in n variables $\{x_1, \dots, x_n\}$ will be denoted $\mathbb{R}[x_1, \dots, x_n]$ or, more compactly, $\mathbb{R}[x]$.
- The degree of a polynomial f is defined as

$$\deg f := \max_{\alpha \in \mathcal{A}, c_\alpha \neq 0} \deg m_\alpha .$$

Vector Representation

If p is a polynomial of degree $\leq d$ in n variables then there exists a coefficient vector $c \in \mathbb{R}^{l_w}$ such that $p = c^\top w(x)$ where

$$w(x) := \left[1,\ x_1,\ x_2,\ \ldots,\ x_n,\ x_1^2,\ x_1 x_2,\ \ldots,\ x_n^2,\ \ldots,\ x_n^d\right]^\top$$

And l_w denotes the length of w. It is easy to verify $l_w = \binom{n+d}{d}$.

Example: Using SOSTOOLs/Multipoly,

```
pvar x1 x2
p = 2*x1^4 + 2*x1^3*x2 - x1^2*x2^2 + 5*x2^4;
x = [x1;x2];
w = monomials(x,0:4);
c = poly2basis(p,w);
[c w]
```

```
[  0,           1]
[  0,          x1]
[  0,          x2]
[  0,        x1^2]
[  0,       x1*x2]
[  0,        x2^2]
[  0,        x1^3]
[  0,     x1^2*x2]
[  0,     x1*x2^2]
[  0,        x2^3]
[  2,        x1^4]
[  2,     x1^3*x2]
[ -1,  x1^2*x2^2]
[  0,     x1*x2^3]
[  5,        x2^4]
```

Gram Matrix Representation

If p is a polynomial of degree $\leq 2d$ in n variables then there exists a $Q = Q^\top \in \mathbb{R}^{l_z}$ such that $p = z^\top \, Q \, z$ where

$$z := \left[1,\ x_1,\ x_2,\ \ldots,\ x_n,\ x_1^2,\ x_1 x_2,\ \ldots,\ x_n^2,\ \ldots,\ x_n^d\right]^\top$$

The dimension of z is $l_z = \binom{n+d}{d}$. Equating coefficients of p and $z^\top \, Q \, z$ yields linear equality constraints on the entries of Q.

- Define $q := \text{vec}(Q)$ and $l_w = \binom{n+d}{d}$.
- There exists $A \in \mathbb{R}^{l_w \times l_z^2}$ and $c \in \mathbb{R}^{l_w}$ such that $p = z^\top Q z$ is equivalent to $A\, q = c$.
- There are $h := \frac{l_z(l_z+1)}{2} - l_w$ linearly independent homogeneous solutions $\{N_i\}_{i=1}^{h}$ each of which satisfies $z^\top N_i \, z = 0$.

Summary: All solutions to $p = z^\top \, Q \, z$ can be expressed as the sum of a particular solution and a homogeneous solution.

Gram Matrix Example

Polynomial p in two variables:

$$p = 2x_1^4 + 2x_1^3 x_2 - x_1^2 x_2^2 + 5x_2^4$$

Gram matrix data:

$$z = \begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_2^2 \end{bmatrix}, \quad Q = \begin{bmatrix} 2 & 1 & -0.5 \\ 1 & 0 & 0 \\ -0.5 & 0 & 5 \end{bmatrix}, \quad N = \begin{bmatrix} 0 & 0 & -0.5 \\ 0 & 1 & 0 \\ -0.5 & 0 & 0 \end{bmatrix}$$

Note that p $= z^\top Q z$ and $z^\top N z = 0$.

Hence $p = z^\top (Q + \lambda N) z$ for all $\lambda \in \mathbb{R}$.

Positive Semidefinite (PSD) Polynomials

$p \in \mathbb{R}[x]$ is positive semi-definite (PSD) if $p(x) \geq 0 \;\; \forall x$.

- If p is a (homogeneous) quadratic function then the Gram matrix is unique. Moreover, p is PSD iff the Gram matrix is PSD.
- However, testing if p is PSD is NP-hard when the polynomial degree is at least four.
- Our computational procedures will be based on constructing polynomials which are PSD.

Objective: Given $p \in \mathbb{R}[x]$, we would like a polynomial-time sufficient condition for testing if p is PSD.

Sum of Squares (SOS) Polynomials

p is a <u>sum of squares (SOS)</u> if there exist polynomials $\{f_i\}_{i=1}^N$ such that $p = \sum_{i=1}^N f_i^2$.

- The set of SOS polynomials in n variables $\{x_1, \ldots, x_n\}$ will be denoted $\Sigma[x_1, \ldots, x_n]$ or $\Sigma[x]$.
- If p is a SOS then p is PSD.
  - The Motzkin polynomial, $p = x^2y^4 + x^4y^2 + 1 - 3x^2y^2$, is PSD but not SOS.
  - Hilbert (1888) showed that the sets of PSD and SOS polynomials are equal only for: a) $n = 1$, b) $d = 2$, and c) $d = 4, n = 2$.
- p is a SOS iff $\exists Q = Q^\top \geq 0$ such that $p = z^\top Q z$.

SOS Example (Parrilo, PhD, 2000)

All possible Gram matrix representations of

$$p = 2x_1^4 + 2x_1^3x_2 - x_1^2x_2^2 + 5x_2^4$$

are given by $z^\top(Q + \lambda N)z$ where:

$$z = \begin{bmatrix} x_1^2 \\ x_1x_2 \\ x_2^2 \end{bmatrix}, \; Q = \begin{bmatrix} 2 & 1 & -0.5 \\ 1 & 0 & 0 \\ -0.5 & 0 & 5 \end{bmatrix}, \; N = \begin{bmatrix} 0 & 0 & -0.5 \\ 0 & 1 & 0 \\ -0.5 & 0 & 0 \end{bmatrix}$$

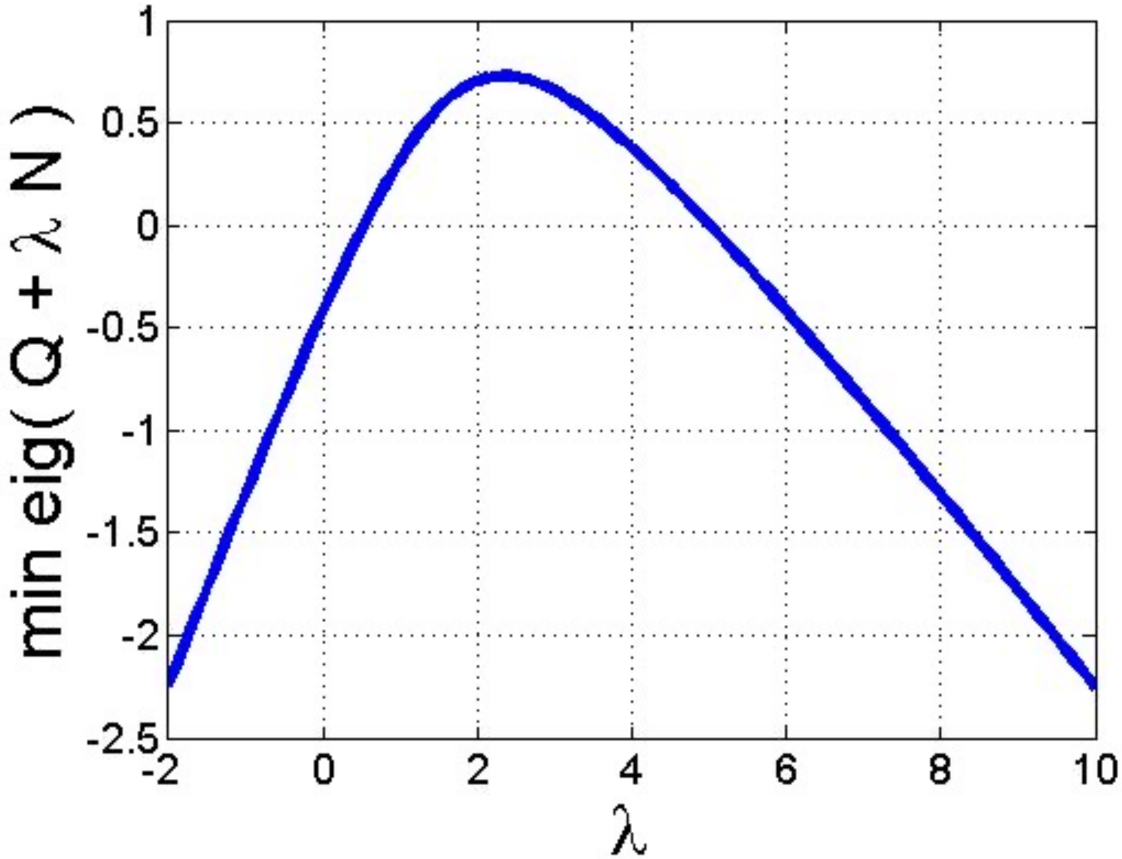
p is SOS iff $Q + \lambda N \geq 0$ for some $\lambda \in \mathbb{R}$.

SOS Example (Parrilo, PhD, 2000)

All possible Gram matrix representations of

$$p = 2x_1^4 + 2x_1^3x_2 - x_1^2x_2^2 + 5x_2^4$$

$Q + \lambda\, N \geq 0$ for some $\lambda = 5$ so p is SOS.



SOS Example (Parrilo, PhD, 2000)

All possible Gram matrix representations of

$$p = 2x_1^4 + 2x_1^3x_2 - x_1^2x_2^2 + 5x_2^4$$

$Q + \lambda N \geq 0$ for some $\lambda = 5$ so p is SOS.

An SOS decomposition can be constructed from a Cholesky factorization $Q + 5N = L^\top L$ where:

$$L = \frac{1}{\sqrt{2}} \begin{bmatrix} 2 & 1 & -3 \\ 0 & 3 & 1 \end{bmatrix}$$

Thus

$$\begin{aligned} p &= 2x_1^4 + 2x_1^3x_2 - x_1^2x_2^2 + 5x_2^4 \\ &= (Lz)^\top (Lz) \\ &= \frac{1}{2}\left(2x_1^2 - 3x_2^2 + x_1x_2\right)^2 + \frac{1}{2}\left(x_2^2 + 3x_1x_2\right)^2 \in \Sigma[x] \end{aligned}$$

Connection to LMIs

Checking if a given polynomial p is a SOS can be done by solving a linear matrix inequality (LMI) feasibility problem.

Primal (Image) Form:

- Find $A \in \mathbb{R}^{l_w \times l_z^2}$ and $c \in \mathbb{R}^{l_w}$ such that $p = z^\top Q z$ is equivalent to $A\, q \;=\; c$ where $q = \mathrm{vec}(Q)$.
- p is a SOS if and only if there exists $Q \geq 0$ such that $A\, q = c$.

Dual (Kernel) Form:

- Let Q_0 be a particular solution of $p = z^\top Q z$ and let $\{N_i\}_{i=1}^{h}$ be a basis for the homogeneous solutions.
- p is a SOS if and only if there exists $\lambda \in \mathbb{R}^h$ such that

$$Q_0 + \sum_{i=1}^{h} \lambda_i N_i \geq 0.$$

SOS Feasibility

SOS Feasibility: Given polynomials $\{f_k\}_{k=0}^{m}$, does there exist $\alpha \in \mathbb{R}^m$ such that $f_0 + \sum_{k=1}^{m} \alpha_k f_k$ is a SOS?

The SOS feasibility problem can also be posed as an LMI feasibility problem since α enters linearly.

Primal (Image) Form:

- Find $A \in \mathbb{R}^{l_w \times l_z^2}$ and $c_k \in \mathbb{R}^{l_w}$ such that $f_k = z^\top Q z$ is equivalent to $A\,q = c_k$ where $q = \text{vec}(Q)$.
- Define $C := -[\,c_1, c_2, \cdots c_n\,] \in \mathbb{R}^{l_w \times n}$.
- There is an $\alpha \in \mathbb{R}^m$ such that $f_0 + \sum_{k=1}^{m} \alpha_k f_k$ is a SOS iff there exists $\alpha \in \mathbb{R}^m$ and $Q \geq 0$ such that $A\,q + C\,\alpha = c_0$.

Dual (Kernel) Form:

- Let Q_k be particular solutions of $f_k = z^\top Q\, z$ and let $\{N_i\}_{i=1}^{h}$ be a basis for the homogeneous solutions.
- There is an $\alpha \in \mathbb{R}^m$ such that $f_0 + \sum_{k=1}^{m} \alpha_k f_k$ is a SOS iff there exists $\alpha \in \mathbb{R}^m$ and $\lambda \in \mathbb{R}^h$ such that $Q_0 + \sum_{k=1}^{m} \alpha_k Q_k + \sum_{i=1}^{h} \lambda_i N_i \geq 0$.

SOS Programming

SOS Programming: Given $c \in \mathbb{R}^m$ and polynomials $\{f_k\}_{k=0}^m$, solve:

$$\min_{\alpha \in \mathbb{R}^m} c^\top \alpha \;\; \text{subject to:} \;\; f_0 + \sum_{k=1}^{m} \alpha_k f_k \in \Sigma[x]$$

This SOS programming problem is an SDP.

- The cost is a linear function of α.
- The SOS constraint can be replaced with either the primal or dual form LMI constraint.
- A more general SOS program can have many SOS constraints.

There is freely available software (e.g. SOSTOOLS, YALMIP, SOSOPT) that: (i) Converts the SOS program to an SDP, (ii) Solves the SDP with available codes, and (iii) Converts the SDP results back into polynomial solutions.

Complexity of SOS Feasiblity Problem

Let p be a degree $2d$ polynomial in n variables. The complexity of the LMI to test if p is an SOS grows rapidly in (n, d).

For example, the Gram matrix $Q = Q^\top$ is $l_z \times l_z$ where the dependence of l_z on (n, d) is shown below.

$l_z = \binom{n+d}{d}$	$2d$=4	6	8	10
$n = 2$	6	10	15	21
5	21	56	126	252
9	55	220	715	2002
14	120	680	3060	11628
16	153	969	4845	20349

SOS Programming Example

Problem: Minimize α subject to $f_0 + \alpha\, f_1 \in \Sigma[x]$ where

$$f_0(x) := -x_1^4 + 2x_1^3x_2 + 9x_1^2x_2^2 - 2x_2^4$$

$$f_1(x) := x_1^4 + x_2^4$$

For every $\alpha, \lambda \in \mathbb{R}$, the Gram Matrix Decomposition equality is:

$$f_0 + \alpha f_1 = z^\top \left(Q_0 + \alpha Q_1 + \lambda N_1\right) z$$

where

$$z := \begin{bmatrix} x_1^2 \\ x_1x_2 \\ x_2^2 \end{bmatrix}, Q_0 = \begin{bmatrix} -1 & 1 & 4.5 \\ 1 & 0 & 0 \\ 4.5 & 0 & -2 \end{bmatrix}, Q_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, N_1 = \begin{bmatrix} 0 & 0 & -0.5 \\ 0 & 1 & 0 \\ -0.5 & 0 & 0 \end{bmatrix}$$

Thus the problem is equivalent to the SDP

$$\min_{\alpha,\lambda\in\mathbb{R}} \alpha \ \text{ subject to: } \ Q_0 + \alpha Q_1 + \lambda N_1 \succeq 0$$

SOS Programming Example

Use SOSTOOLs to minimize α subject to $f_0 + \alpha\, f_1 \in \Sigma[x]$.

```
% Define polynomials in the SOS optimization
pvar x1 x2 alpha;
f0 = -x1^4 + 2*x1^3*x2 + 9*x1^2*x2^2 - 2*x2^4;
f1 = x1^4 + x2^4;

% Solve the SOS optimization
prog = sosprogram([x1;x2]);            % Define polynomial variables
prog = sosdecvar(prog,alpha);          % Define decision variable
prog = sosineq(prog,f0+alpha*f1);      % Define SOS constraint
prog = sossetobj(prog,alpha);          % Define objective function
prog = sossolve(prog);                 % Solve optimization

alphaOPT = sosgetsol(prog,alpha)       % Get optimal solution
alphaOPT =   2
```

Global Stability Conditions Using SOS

Revisit the autonomous nonlinear dynamical system

$$\dot{x}(t) = f(x(t))$$

where $x(t) \in \mathbb{R}^n$ is the state at time t and $f: \mathbb{R}^n \to \mathbb{R}^n$. Assume f is a polynomial function of x and $f(0) = 0$

Theorem: Let $l_1, l_2 \in \mathbb{R}[x]$ be given with $l_i(0) = 0$ and $l_i(x) > 0 \ \forall x \ (i = 1,2)$. The point $x = 0$ is a globally asymptotically stable (GAS) equilibrium if $\exists \, V \in \mathbb{R}[x]$ such that:

- $V(0) = 0$
- $V - l_1 \in \Sigma[x]$
- $-\nabla V \cdot f - l_2 \in \Sigma[x]$

Proof: The conditions imply that V is pos. def, decrescent, and radially unbounded. Moreover, $-\nabla \mathrm{V} \cdot f$ is a positive definite. Hence V is a Lyapunov function that proves $x = 0$ is GAS.

Global Stability Example

The following example is sosdemo2 in SOSTOOLs. See Section 4.2 of SOSTOOLS User's Manual.

```
% Constructing the vector field dx/dt = f
pvar x1 x2 x3;  vars = [x1; x2; x3];
f = [(-x1^3-x1*x3^2)*(x3^2+1);  (-x2-x1^2*x2)*(x3^2+1); ...
   (-x3+3*x1^2*x3)*(x3^2+1)-3*x3];
% SOS Program
prog = sosprogram(vars);
[prog,V] = sospolyvar(prog,[x1^2; x2^2; x3^2],'wscoeff');
prog = sosineq(prog,V-(x1^2+x2^2+x3^2));
expr = -(diff(V,x1)*f(1)+diff(V,x2)*f(2)+diff(V,x3)*f(3));
prog = sosineq(prog,expr);
solver_opt.solver = 'sedumi';
prog = sossolve(prog,solver_opt);
SOLV = sosgetsol(prog,V)
    SOLV =   6.6589*x1^2 + 4.6277*x2^2 + 2.0734*x3^2
```

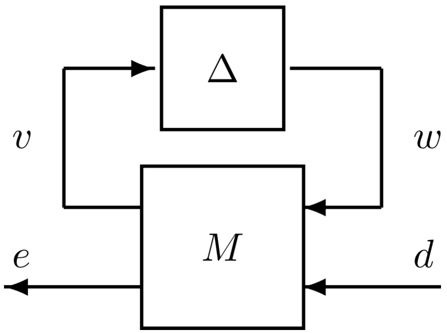
Constructing Storage Functions With SOS & IQC

We can combine SOS and IQC techniques. Consider an uncertain system $F_U(M,\Delta)$ where:

1. M is described by polynomial dynamics:

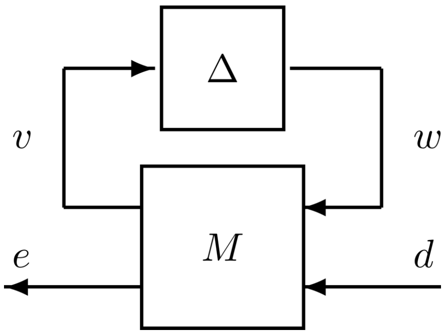
$$\dot{x} = f(x,w,d), \quad v = g_1(x,w,d) \quad e = g_2(x,w,d)$$

2. Δ satisfies the QC defined by $J = J^\top$.



Constructing Storage Functions With SOS & IQC

We can combine SOS and IQC techniques. Consider an uncertain system $F_U(M,\Delta)$ where:



1. M is described by polynomial dynamics:

$$\dot{x} = f(x,w,d), \quad v = g_1(x,w,d) \quad e = g_2(x,w,d)$$

2. Δ satisfies the QC defined by $J = J^\top$.

Theorem: $F_U(M,\Delta)$ has L_2 gain $\leq \gamma$ if $\exists\, V \in \mathbb{R}[x]$ such that:

- $V(0) = 0$
- $V \in \Sigma[x]$
- $-\left(\nabla V^\top f + (e^\top e - \gamma^2 d^\top d) + \begin{bmatrix} v \\ w \end{bmatrix}^\top J \begin{bmatrix} v \\ w \end{bmatrix}\right) \in \Sigma[x,w,d]$

Comments:

- The last condition is a dissipation ineq. with IQC. It is a polynomial in (x,w,d) after substituting for (v,e) using the output equations of M.
- These conditions can be checked as an SOS optimization.

Equilibrium Independent Dissipativity

Recall the Equilibrium Independent Dissipativity (EID) conditions

$$V(\bar{x},\bar{x}) = 0, \quad \nabla_x V(x,\bar{x})^\top f(x,u) \leq s(u-\bar{u}, y-\bar{y}) \qquad \textbf{(1)}$$

where $\bar{u}, \bar{y}$ are functions of $\bar{x}$ through $f(\bar{x},\bar{u}) = 0,\ \bar{y} = h(\bar{x},\bar{u})$.

Assume system defined by polynomial f, h and the supply rate s is also polynomial. Recall $\mathbb{R}[x]$ set of all polynomials and $\Sigma[x]$ all SOS polynomials in x. SOS formulation for EID:

$$\begin{aligned}
-\nabla_x V(x,\bar{x})^T f(x,u) + s(u-\bar{u}, h(x,u) - h(\bar{x},\bar{u})) \\
+ r(x,u,\bar{x},\bar{u}) f(\bar{x},\bar{u}) \in \Sigma[x,u,\bar{x},\bar{u}] \\
r(x,u,\bar{x},\bar{u}) \in \mathbb{R}[x,u,\bar{x},\bar{u}]
\end{aligned}$$

To enforce $V(\bar{x},\bar{x}) = 0$ take $V(x,\bar{x}) = (x-\bar{x})^T Q(x,\bar{x})(x-\bar{x})$ where $Q(x,\bar{x})$ is a pos.def. symmetric matrix of polynomials.

Note: $\bar{x}, \bar{u}$ are independent variables in the SOS program, but the term $r(x,u,\bar{x},\bar{u}) f(\bar{x},\bar{u})$ ensures (1) holds when $f(\bar{x},\bar{u}) = 0$.

Delta Dissipativity

Recall the delta dissipativity conditions:

$$S(x,u)=0 \Leftrightarrow f(x,u)=0$$

$$\nabla_x S(x,u)^\top f(x,u) + \nabla_u S(x,u)^\top v \leq s(v,w) \quad \forall x,u,v$$

where $w := \nabla_x h(x,u)^\top f(x,u) + \nabla_u h(x,u)^\top v.$

SOS formulation:

$$s(v,w(x,u,v)) - \left(\nabla_x S(x,u)^\top f(x,u) + \nabla_u S(x,u)^\top v\right) \in \Sigma[x,u,v]$$

where $S(x,u) = \psi(x,u)^\top P(x,u)\psi(x,u)$ with user-specified ψ s.t.

$$\psi(x,u)=0 \Leftrightarrow f(x,u)=0$$

and P symmetric matrix of polynomials, enforced to be pos.def. by

$$l^\top (P(x,u)-\delta I)l \in \Sigma[x,u,l],\ \ \delta > 0$$

Generalizations

- Dynamic IQCs (Ψ, J) can be combined with the search for polynomial storage functions.

- "Local" conditions can be constructed to estimate regions of attraction or local input/output gains.
  - These conditions involve set containment constraints that can be relaxed via Lagrange multipliers (S-procedure).
  - This typically leads to non-convex, bilinear SOS conditions.
  - Various heuristic iterations have been developed to approximately solve these conditions.

Time-Varying Systems

Wind Turbine
Periodic /
Parameter-Varying

Flexible Aircraft
Parameter-Varying

Vega Launcher
Time-Varying
(Source: ESA)

Robotics
Time-Varying
(Source: ReWalk)

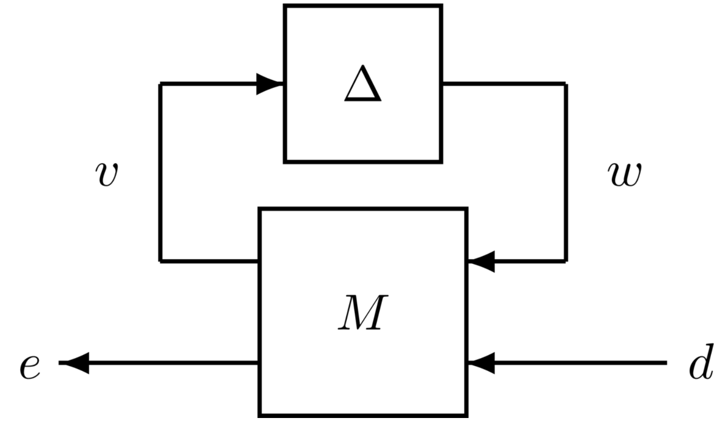
The IQC/DI results can be extended to assess the robustness of time-varying systems.

(Robust) Finite-Horizon Analysis

Uncertain LTV System

$$\begin{bmatrix} \dot{x}(t) \\ v(t) \\ e(t) \end{bmatrix} = \begin{bmatrix} A(t) & B_1(t) & B_2(t) \\ C_1(t) & D_1(t) & D_2(t) \\ C_2(t) & 0 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ w(t) \\ d(t) \end{bmatrix}$$

$$x(0) = 0$$



Uncertainty set $\boldsymbol{\Delta}$ can be block-structured with parametric / non-parametric uncertainties and nonlinearities.

Analysis Objective

Derive bound on $\|e(T)\|_2$ that holds for all disturbances $\|d\|_{2,[0,T]} \leq 1$ and uncertainties $\Delta \in \boldsymbol{\Delta}$ on the horizon $[0, T]$.

Integral Quadratic Constraints (IQCs)

The robustness analysis uses constraints on the I/O behavior of Δ expressed as (time-domain) IQCs.

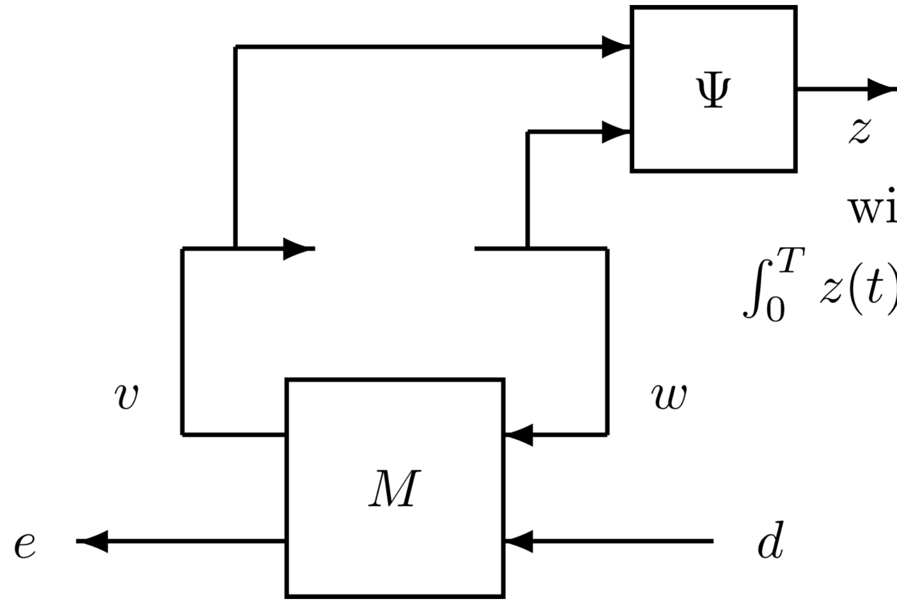
Definition: Δ satisfies the finite-horizon IQC defined by a stable filter Ψ and a matrix $J = J^\top \in \mathbb{R}^{(n_v+n_w)\times(n_v+n_w)}$ if every $v \in L_2[0,T]$ and $w = \Delta(v)$ satisfies:

$$\int_0^T z(t)^\top J z(t)\, dt \geq 0$$

Comments:

- The analysis that follows only requires the IQC to hold over the finite horizon [0,T].
- The filter Ψ and matrix J can be time-varying with only notational changes, e.g. we could have a QC with time-varying sector bounds.

Robustness Analysis



with $\int_0^T z(t)^T J z(t)\, dt \geq 0$

The robustness analysis is performed on the extended (LTV) system of (J, Ψ) using the constraint on z.

$$\begin{bmatrix} \dot{x}_e(t) \\ z(t) \\ e(t) \end{bmatrix} = \begin{bmatrix} \mathcal{A}(t) & \mathcal{B}_1(t) & \mathcal{B}_2(t) \\ \mathcal{C}_1(t) & \mathcal{D}_1(t) & \mathcal{D}_2(t) \\ \mathcal{C}_2(t) & 0 & 0 \end{bmatrix} \begin{bmatrix} x_e(t) \\ w(t) \\ d(t) \end{bmatrix}$$

Robust Finite Horizon Analysis

Theorem [1,2]

Assume Δ satisfies the IQC defined by (Ψ, J).
If there exists $P(\cdot) = P(\cdot)^\top$ such that
(i) $P(T) = \mathcal{C}_2(T)^\top \mathcal{C}_2(T)$, and
(ii) $V(x,t) := x^\top P(t) x$ satisfies

$$\frac{d}{dt}V(x,t) - \gamma^2 d(t)^\top d(t) + z(t)^\top J z(t) \leq 0 \;\; \forall t \in [0,T]$$

then $\|e(T)\|_2 \leq \gamma \|d\|_{2,[0,T]}$

Proof

Integrate dissipation inequality from $t=0$ to $t=T$:

$$\underbrace{V(x(T),T)}_{=e(T)^\top e(T)} - \underbrace{V(x(0),0)}_{=0} - \gamma^2 \int_0^T d(t)^\top d(t) dt + \underbrace{\int_0^T z(t)^\top J z(t) dt}_{\geq 0} \leq 0$$

[1] Moore, Finite Horizon Robustness Analysis using IQCs, MS Thesis, Berkeley, 2015.
[2] Seiler, Moore, Meissen, Arcak, Packard, Finite Horizon Robustness Analysis of LTV Systems Using IQCs, arXiv 2018 and Automatica 2019.

Robust Finite Horizon Analysis

Theorem [1,2]

Assume Δ satisfies the IQC defined by (Ψ, J).
If there exists $P(\cdot) = P(\cdot)^\top$ such that
(i) $P(T) = \mathcal{C}_2(T)^\top \mathcal{C}_2(T)$, and
(ii) $V(x,t) := x^\top P(t) x$ satisfies

$$\frac{d}{dt}V(x,t) - \gamma^2 d(t)^\top d(t) + z(t)^\top J z(t) \leq 0 \;\; \forall t \in [0,T]$$

then $\|e(T)\|_2 \leq \gamma \|d\|_{2,[0,T]}$

Dissipation inequality can be recast as a differential LMI:

$$\begin{bmatrix} \dot{P} + \mathcal{A}^\top P + P\mathcal{A} & P\mathcal{B}_1 & P\mathcal{B}_2 \\ \mathcal{B}_1^\top P & 0 & 0 \\ \mathcal{B}_2^\top P & 0 & -\gamma^2 I \end{bmatrix} + (\cdot)^\top J \begin{bmatrix} \mathcal{C}_1 & \mathcal{D}_1 & \mathcal{D}_2 \end{bmatrix} \preceq 0$$

$\forall t \in [0,T]$

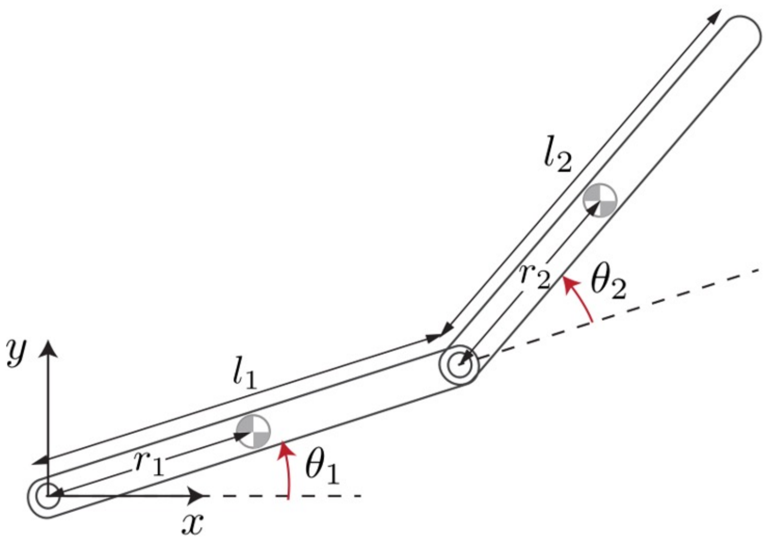
[1] Moore, Finite Horizon Robustness Analysis using IQCs, MS Thesis, Berkeley, 2015.
[2] Seiler, Moore, Meissen, Arcak, Packard, Finite Horizon Robustness Analysis of LTV Systems Using IQCs, arXiv 2018 and Automatica 2019.

Numerical Algorithms and Software

- **Robustness Algorithms**
  - Differential LMI can be "solved" via convex optimization using basis functions for $P(\cdot)$ and gridding on time [1].
  - A more efficient algorithm mixes the differential LMI and a related Riccati Differential Equation condition [2].
  - Similar methods developed for LPV [4,5] and periodic systems [6].
- **LTVTools Software [3]**
  - Time-varying state space system objects, e.g. obtained from Simulink snapshot linearizations.
  - Includes functions for nominal and robustness analyses.

[1] Moore, Finite Horizon Robustness Analysis using IQCs, MS Thesis, Berkeley, 2015.
[2] Seiler, Moore, Meissen, Arcak, Packard, Finite Horizon Robustness Analysis of LTV Systems Using IQCs, arXiv 2018 and Automatica 2019.
[3] https://z.umn.edu/LTVTools
[4] Pfifer & Seiler, Less Conservative Robustness Analysis of LPV Systems Using IQCs, IJRNC, 2016.
[5] Hjartarson, Packard, Seiler, LPVTools: A Toolbox for Modeling, Analysis, & Synthesis of LPV Systems, 2015.
[6] Fry, Farhood, Seiler, IQC-based robustness analysis of discrete-time LTV systems, IJRNC 2017.

Two-Link Robot Arm



Two-Link Diagram [1]

Nonlinear dynamics [MZS]:

$$\dot{\eta} = f(\eta, \tau, d)$$

where

$$\eta = [\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2]^T$$
$$\tau = [\tau_1,\ \tau_2\ ]^T$$
$$d = [d_1,\ d_2\ ]^T$$

τ and d are control torques and disturbances at the link joints.

[1] R. Murray, Z. Li, and S. Sastry. *A Mathematical Introduction to Robot Manipulation*, 1994.

Nominal Trajectory in Cartesian Coordinates

Analysis

Nonlinear dynamics:

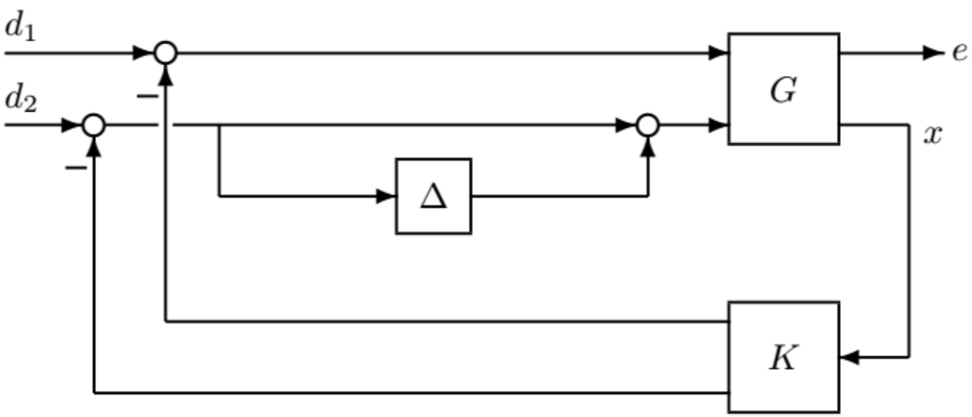
$$\dot{\eta} = f(\eta, \tau, d)$$

Linearize along the finite –horizon trajectory $(\bar{\eta}, \bar{\tau}, d = 0)$

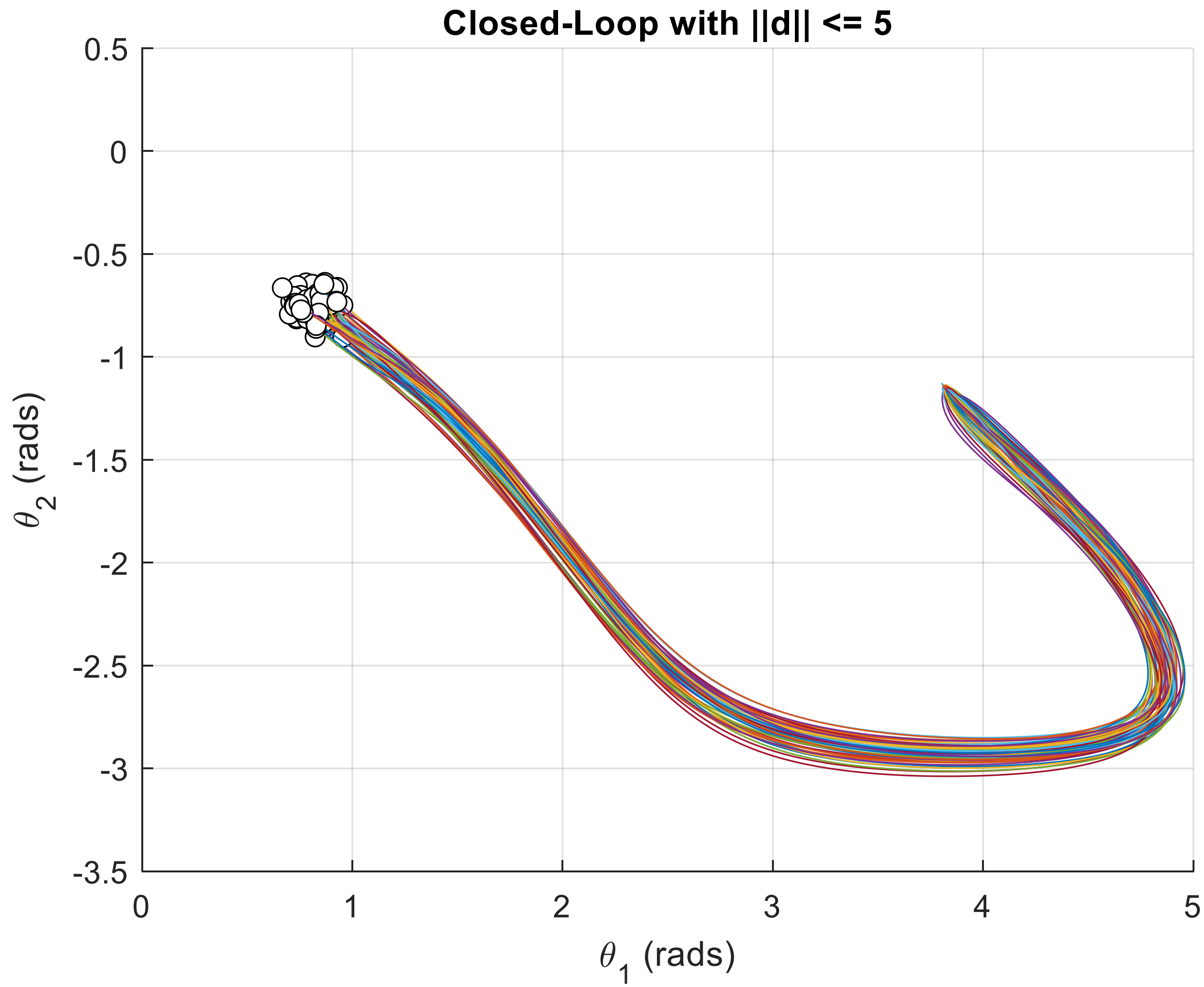
$$\dot{x} = A(t)x + B(t)u + B(t)d$$

Design finite-horizon state-feedback LQR gain.

Goal: Compute bound on the final position accounting for disturbances and LTI uncertainty Δ at 2nd joint.

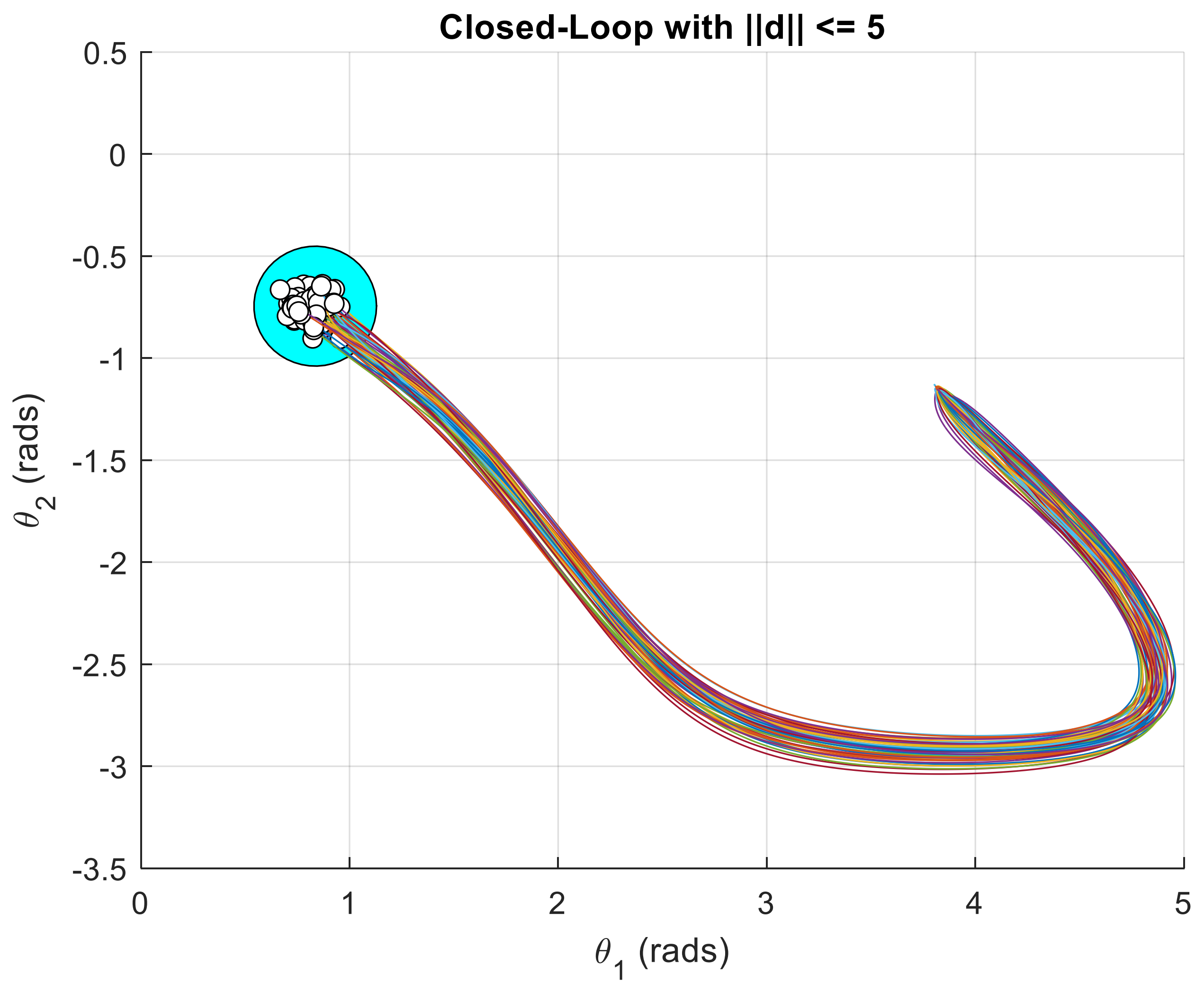


Monte-Carlo Simulations



LTV simulations with randomly sampled disturbances and uncertainties (overlaid on nominal trajectory).

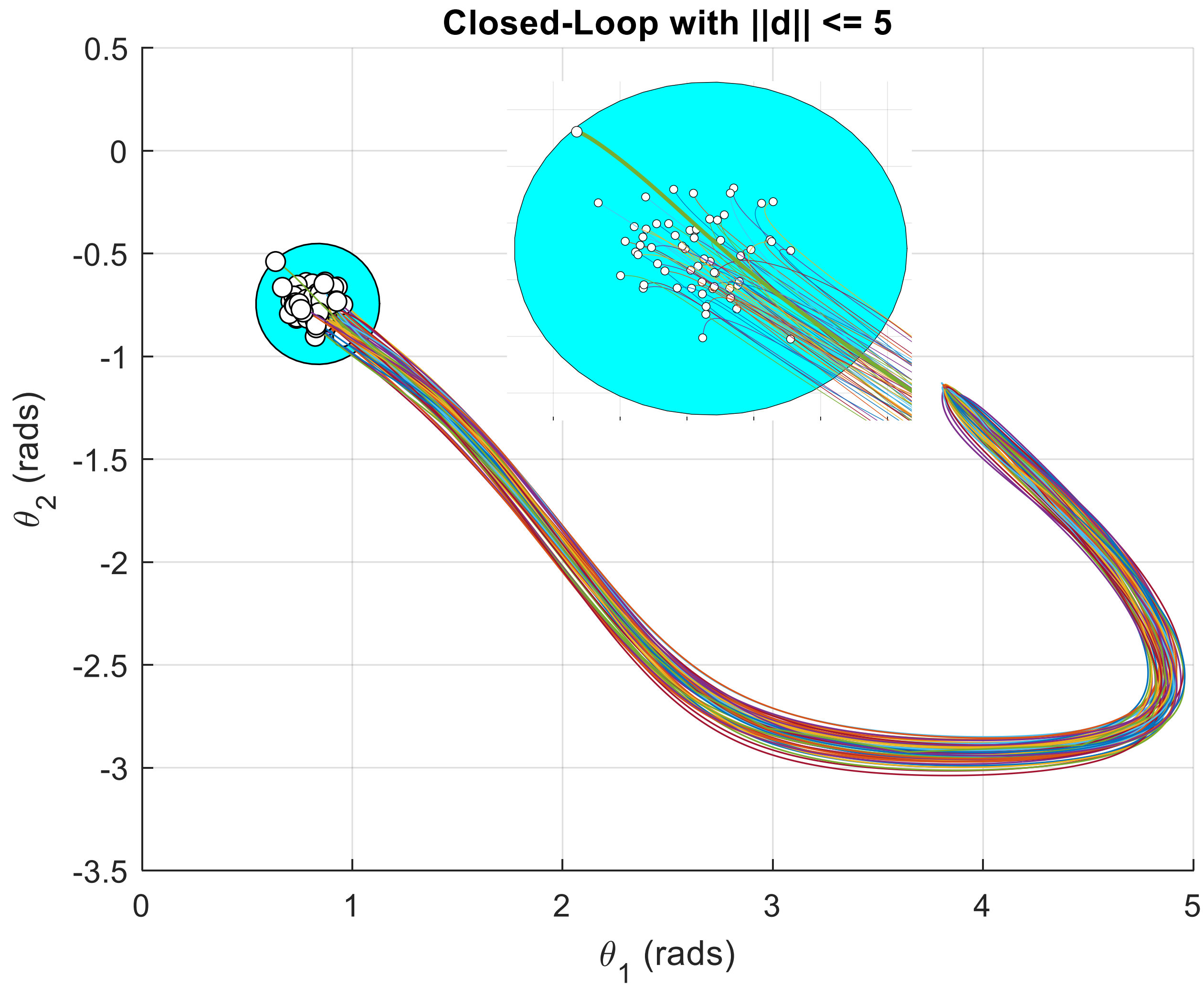
Robustness Bound



Cyan disk is bound computed in 102 sec using IQC/DI method

Bound accounts for disturbances $\|d\| \leq 5$ and $\|\Delta\| \leq 0.8$

Worst-Case Uncertainty / Disturbance



Randomly sample Δ to find “bad” perturbation and compute corresponding worst-case disturbance using method in [1].

[1] Iannelli, Seiler, Marcos, Construction of worst-case disturbances for LTV systems…, 2019.

Intermezzo

Given two functions $p, q : \mathbb{R}^n \to \mathbb{R}$ how can we show

$$\{x : p(x) \leq 0\} \subseteq \{x : q(x) \leq 0\} \qquad \textbf{(1)}$$

i.e., $p(x) \leq 0 \;\Rightarrow\; q(x) \leq 0$?

If we can find $\lambda : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ such that

$$\lambda(x)p(x) - q(x) \geq 0 \quad \forall x \in \mathbb{R}^n \qquad \textbf{(2)}$$

then $q(x) \leq \lambda(x)p(x)$ and, since $\lambda(x) \geq 0$, $p(x) \leq 0 \;\Rightarrow\; q(x) \leq 0$.

The idea of using the nonnegativity property (2) to show the set containment (1) is called the **S-procedure** in control theory.

When p, q are polynomials we can apply the S procedure with SOS programming: find polynomial $\lambda(x)$ such that

$$\lambda(x) \in \Sigma[x]$$

$$\lambda(x)p(x) - q(x) \in \Sigma[x]$$

Nonlinear Reachability Analysis

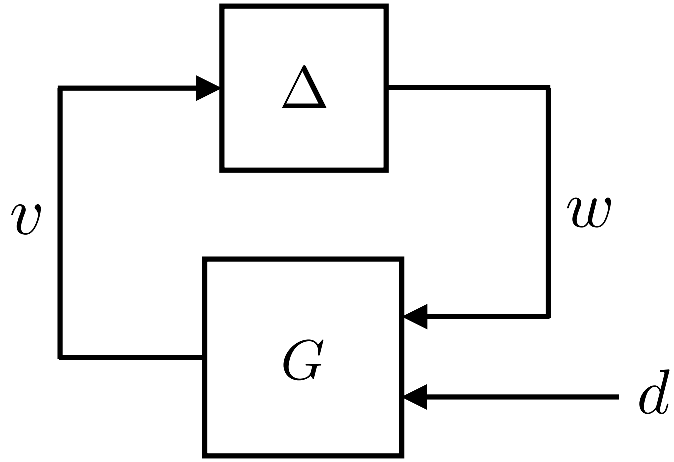
1. Forward Reachability: Bounding trajectories from a set of initial conditions in the presence disturbances and unmodeled dynamics

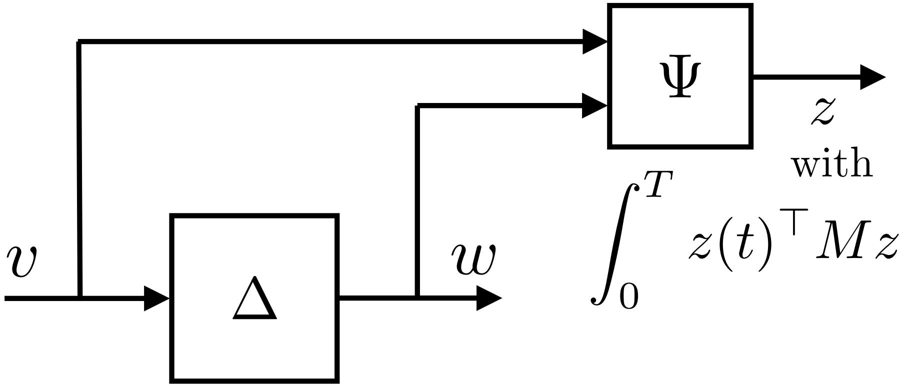
G: nominal plant model

$$\dot{x}_G = f(x_G, w, d)$$
$$v = h(x_G, w, d)$$

d: disturbance

Δ: unmodeled dynamics characterized by IQC:

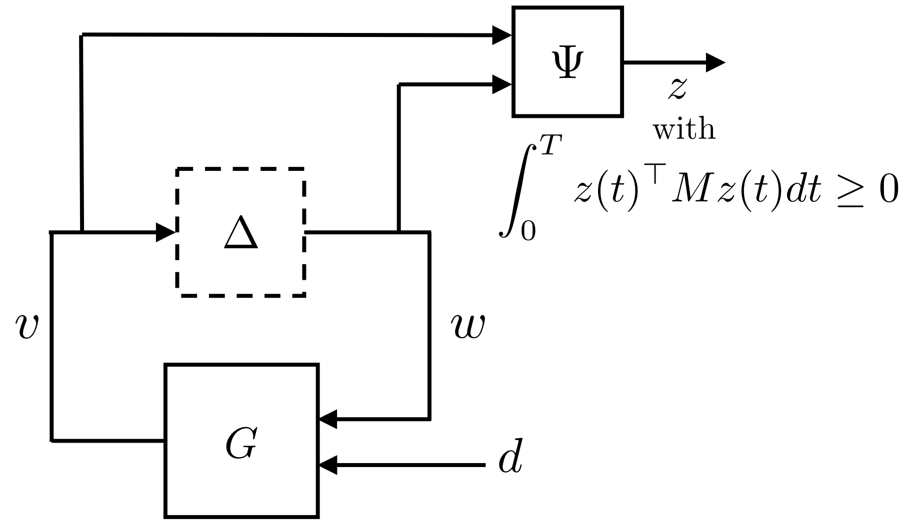




$$\int_0^T z(t)^\top M z(t) dt \geq 0$$

Nonlinear Reachability Analysis

Goal: Given set of initial conditions X_0 find an outer bound on trajectories at time T for all Δ satisfying the IQC and for all d s.t. $\|d\|_{\mathcal{L}_2,[0,T]} \leq R$.

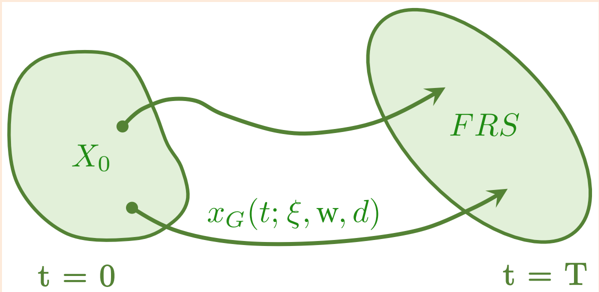


Lump plant and filter into single model with state $x = [x_G; x_\Psi]$:

$$\dot{x} = F(x, w, d)$$
$$z = H(x, w, d)$$

If ∃ storage function $(t, x) \mapsto V(t, x)$ s.t.

$$\dot{V}(t, x, w, d) + z^\top M z \leq d^\top d \quad \forall t \in [0, T]$$
$$X_0 \times \{0_{n_\Psi}\} \subseteq \{x : V(0, x) \leq 0\}$$



then projection of $\{x : V(T, x) \leq R^2\}$ onto x_G subspace ⊃ FRS.

Nonlinear Reachability Analysis

Proof by integrating dissipation inequality from 0 to T:

$$V(T, x(T)) - V(0, x(0)) + \underbrace{\int_0^T z(\tau)^\top M z(\tau) d\tau}_{\geq 0} \leq \underbrace{\int_0^T d(t)^\top d(t) dt}_{\leq R^2}$$

Then, $x_G(0) \in X_0, x_\Psi(0) = 0 \;\Rightarrow\; V(0, x(0)) \leq 0 \;\Rightarrow\; V(T, x(T)) \leq R^2$

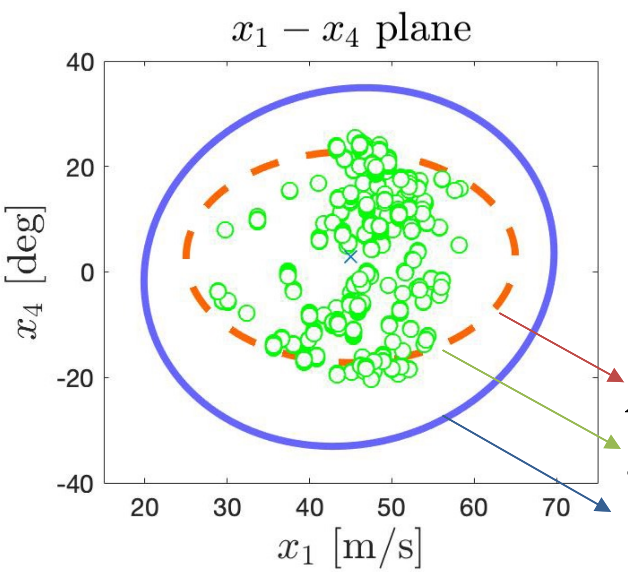
SOS procedure to find V:

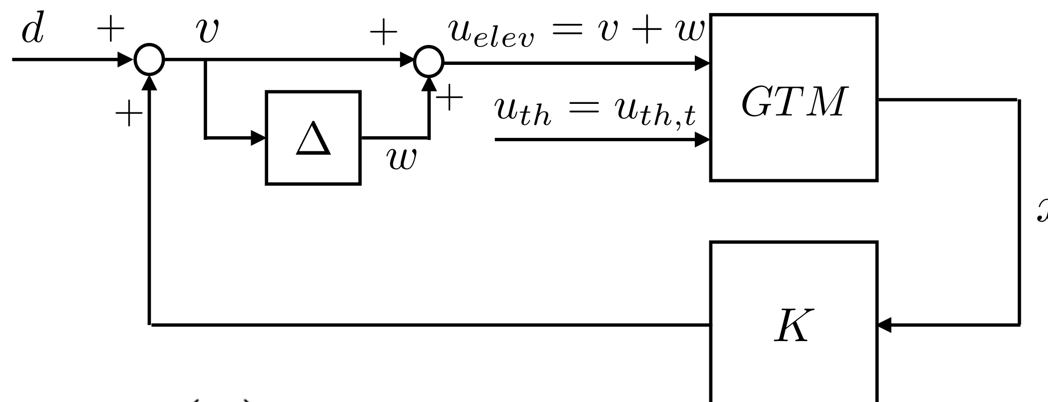
- Use semi-algebraic (sublevel set of polynomial) representation of X_0 and polynomial approximation of system model
- View dissipation inequality as a nonnegativity constraint
- Turn set containment condition $X_0 \times \{0_{n_\Psi}\} \subseteq \{x : V(0, x) \leq 0\}$ to nonnegativity constraint with S-procedure
- SOS relaxation for nonnegativity; SOS then translated into SDP

Nonlinear Reachability Analysis

Example: Generic Transport Model (GTM)

- 5.5% scale commercial aircraft
- State variables: airspeed (x_1), angle of attack (x_2), pitch rate (x_3), pitch angle (x_4)
- Controls: elevator deflection (u_{elev}), engine throttle (u_{th})
- X_0: ellipsoid around the equilibrium
- Disturbance and unmodeled dynamics in elevator control channel:





X_0

samples of $x_G(T)$

FRS estimate for $\|d\|_{\mathcal{L}_2,[0,T]} \le 0.004$, $\|\Delta\|_{\mathcal{L}_2 \to \mathcal{L}_2} \le 0.4$

Nonlinear Reachability Analysis

2. Backward Reachability: Given target set X_T find a set of initial states (BRS) and a controller that drives states from BRS to X_T

G: nominal plant model

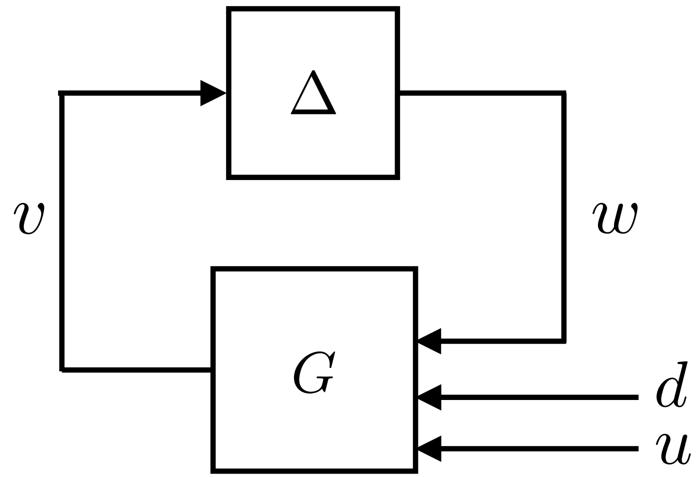
$$\dot{x}_G = f(x_G, w, d) + g(x_G, w, d)u$$
$$v = h(x_G, w, d)$$

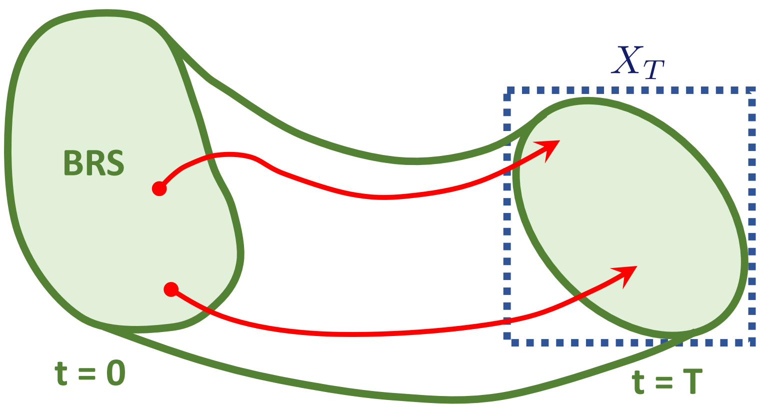
d: disturbance

Δ: unmodeled dynamics characterized by IQC as before

Control design now part of the formulation:

$$u(t) = k(t, x_G(t))$$





Nonlinear Reachability Analysis

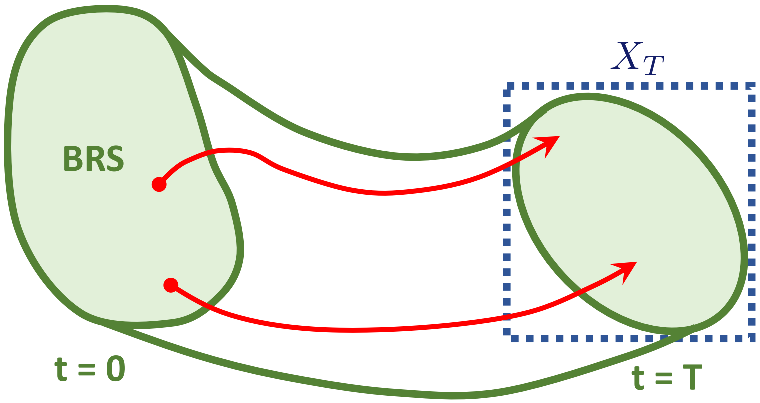
If $\exists$ storage function $(t,x) \mapsto V(t,x)$ and control $u(t) = k(t, x_G(t))$:

$$\partial_t V(t,x) + \partial_x V(t,x) \cdot F(x, w, d, k(t, x_G)) + z^\top M z \le d^\top d \quad \forall t \in [0,T]$$

$$\{x_G : V(T, [x_G; x_\Psi]) \le R^2 \; \exists x_\Psi\} \subseteq X_T$$

then $\{x_G : V(0, [x_G; 0]) \le 0\}$ is a BRS inner approximation.

- Can use SOS to search for V and k by restricting V, k, F, H to be polynomials and X_T to be semi-algebraic
- The dissipation inequality is bilinear in V and k. Alternate the search between the two.
- BRS inner-approximation is useful even if we don't commit to using the control k obtained along with the approximation.



Nonlinear Reachability Analysis

Example: Six-state quadrotor model

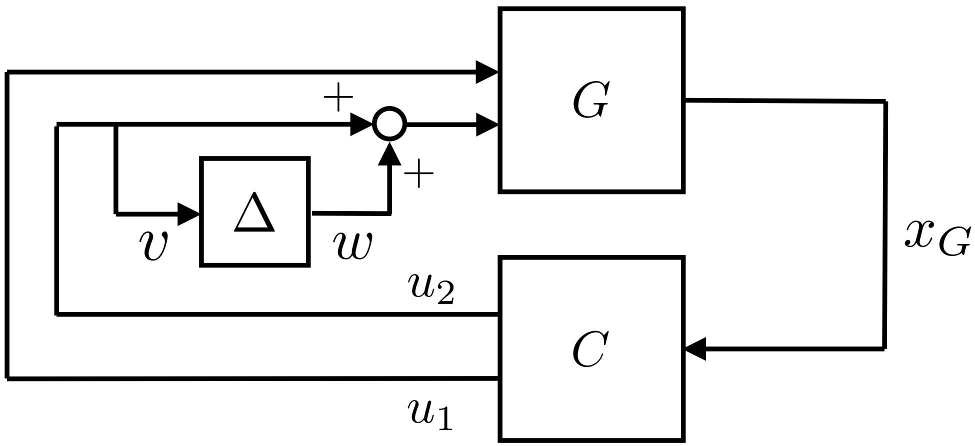
Credit: DJI

$$\dot{x}_1 = x_3,$$
$$\dot{x}_2 = x_4,$$
$$\dot{x}_3 = u_1 K \sin(x_5),$$
$$\dot{x}_4 = u_1 K \cos(x_5) - g_n,$$
$$\dot{x}_5 = x_6,$$
$$\dot{x}_6 = -d_0 x_5 - d_1 x_6 + n_0 u_2$$

x_1 : horizontal position
x_2 : vertical position
x_3 : horizontal velocity
x_4 : vertical velocity
x_5 : roll
x_6 : roll velocity

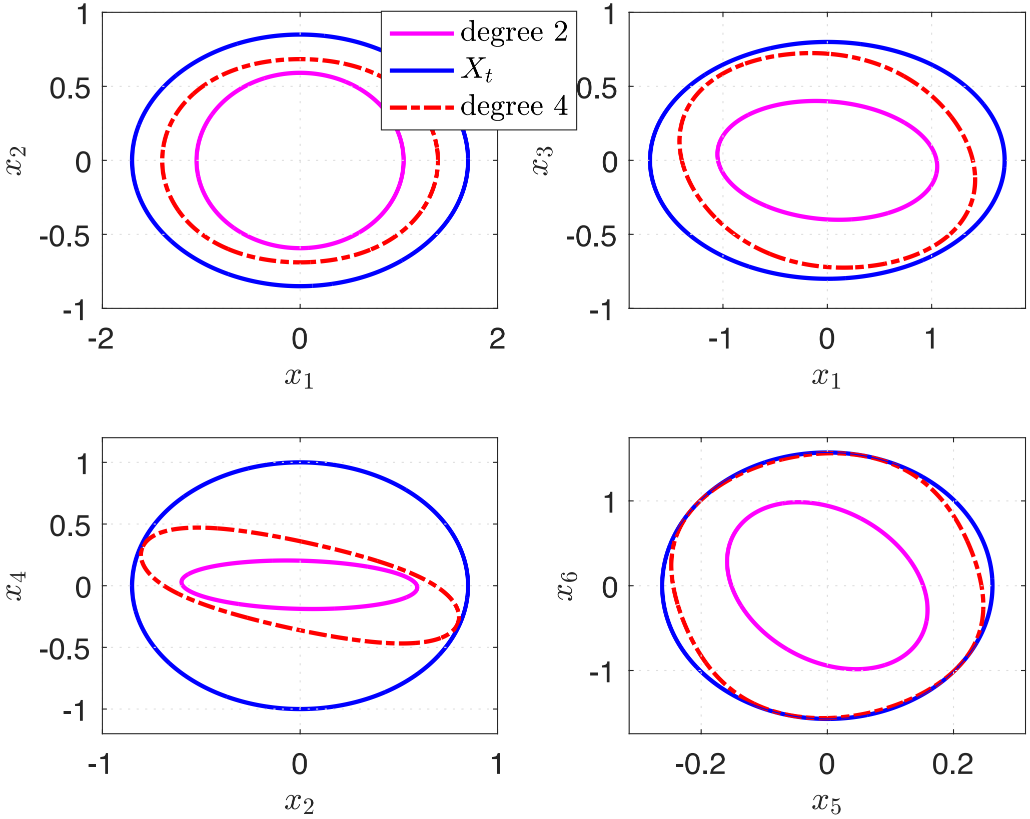
$u_1 \in [-1.5, 1.5] + g_n/K$: total trust
$u_2 \in [-\pi/12, \pi/12]$: desired roll angle

Additive uncertainty Δ acting on u_2:



Nonlinear Reachability Analysis

Example: Six-state quadrotor model BRS inner-approximation with degree-2 and degree-4 polynomial storage functions:



Credit: DJI

- $\|\Delta\|_{\mathcal{L}_2 \to \mathcal{L}_2} \leq 0.2$
- Computation times: 18 min. for degree-2; 60 min for degree-4
- Higher degree: tighter approximation but longer computation

Nonlinear Reachability Analysis

Key take-aways:

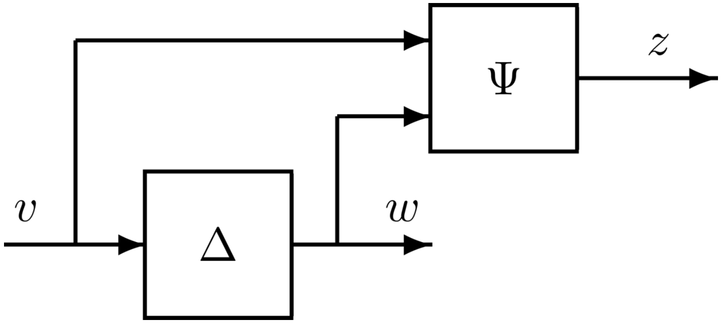
- We can account for dynamic uncertainty in reachability analysis
- Dissipation formulation played a key role:
  - to accommodate dynamic uncertainty (described by IQCs) and disturbances simultaneously
  - to translate analysis/synthesis to optimization problems, via S-procedure and SOS programming
- No gridding of state space required (unlike Hamilton-Jacobi or symbolic control methods, which suffer exponential growth in complexity with state dimension).
- However, scalability is still a challenge for the SOS procedures mentioned

Discrete-Time IQCs

We can define discrete-time IQCs analogously to their continuous-time counterparts. In this case, "summation quadratic constraints" (SQCs) would be more appropriate terminology but we'll continue to use "IQCs ".

Definition: A discrete-time system Δ satisfies the IQC defined by a stable filter Ψ and a matrix $J = J^\top \in \mathbb{R}^{(n_v+n_w)\times(n_v+n_w)}$ if every $v \in \ell_2$ and $w = \Delta(v)$ satisfies:

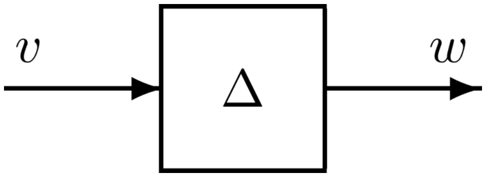
$$\Sigma_{t=0}^{T} z(t)^\top J z(t) \geq 0 \ \forall T \geq 0$$



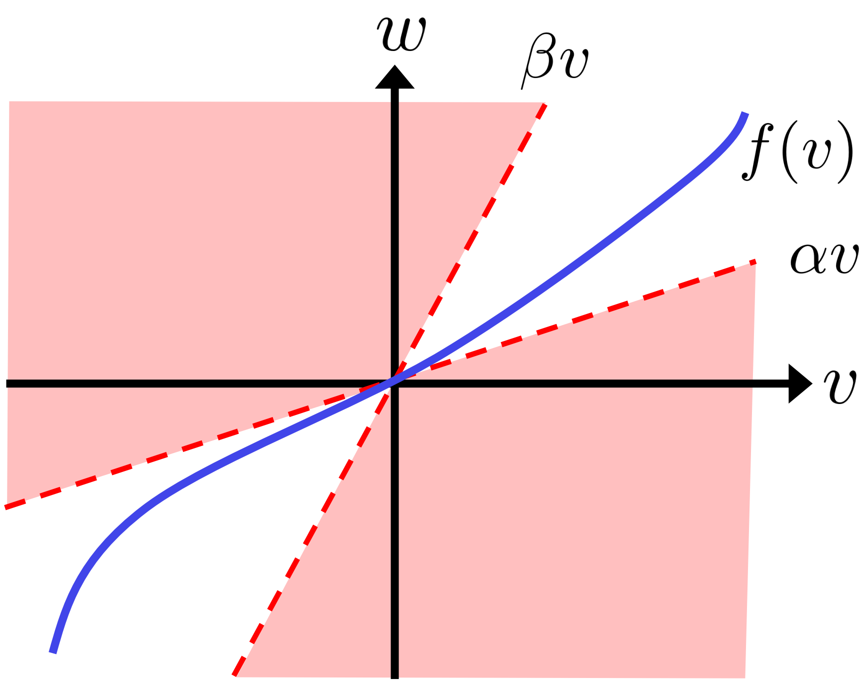
Most continuous-time IQCs have similar discrete-time versions.

We'll briefly discuss a few cases on the next slides.

Example: Sector-bounded Nonlinearity



Suppose Δ is a nonlinearity, $w = f(v)$, whose graph lies in the sector $[\alpha, \beta]$.



$$(w(t) - \alpha v(t)) \cdot (\beta v(t) - w(t)) \geq 0$$

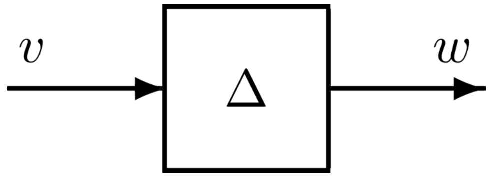
⇕

$$\begin{bmatrix} v(t) \\ w(t) \end{bmatrix}^\top \underbrace{\begin{bmatrix} -2\alpha\beta & \alpha + \beta \\ \alpha + \beta & -2 \end{bmatrix}}_{:=J} \begin{bmatrix} v(t) \\ w(t) \end{bmatrix} \geq 0$$

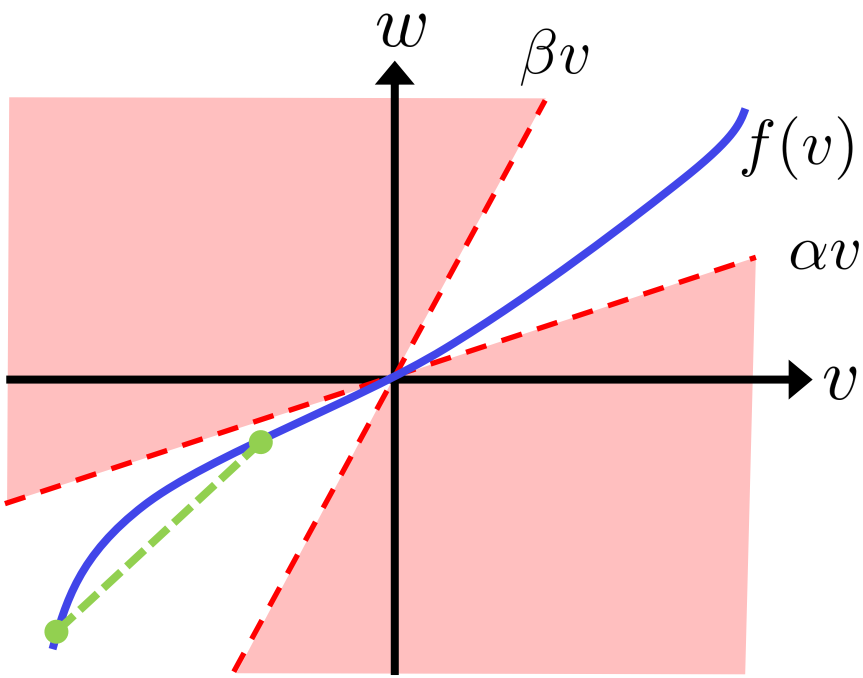
⇕

Δ satisfies the static QC defined by J.

Example: Slope-Restricted Nonlinearity



Suppose Δ is a nonlinearity, $w = f(v)$, whose slope lies in $[\alpha, \beta]$ and $f(0) = 0$.



$$\alpha \leq \frac{w(t_1)-w(t_2)}{v(t_1)-v(t_2)} \leq \beta \quad \forall v(t_1) \neq v(t_2)$$

$$(\delta_w - \alpha\delta_v) \cdot (\beta\delta_v - \delta_w) \geq 0$$

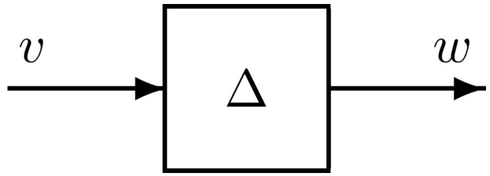
where $\delta_w := w(t_1) - w(t_2)$

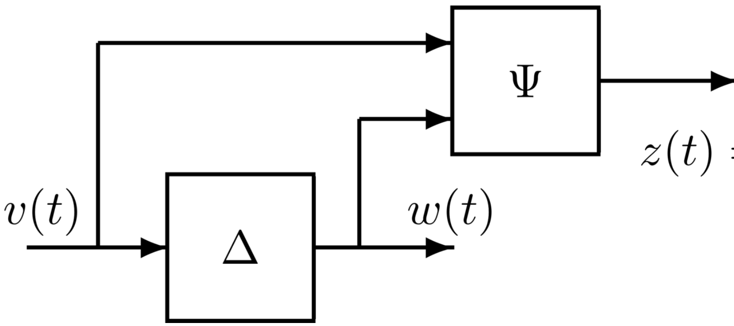
and $\delta_v := v(t_1) - v(t_2)$

$$\begin{bmatrix} v(t_1) \\ v(t_2) \\ w(t_1) \\ w(t_2) \end{bmatrix}^\top J \begin{bmatrix} v(t_1) \\ v(t_2) \\ w(t_1) \\ w(t_2) \end{bmatrix} \geq 0$$

$$J := (\cdot)^\top \begin{bmatrix} -2\alpha\beta & \alpha+\beta \\ \alpha+\beta & -2 \end{bmatrix} \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

Example: Slope-Restricted Nonlinearity





$$z(t) = \begin{bmatrix} v(t) \\ v(t-1) \\ w(t) \\ w(t-1) \end{bmatrix}$$

Suppose Δ is a nonlinearity, $w = f(v)$, whose slope lies in $[\alpha, \beta]$ and $f(0) = 0$.

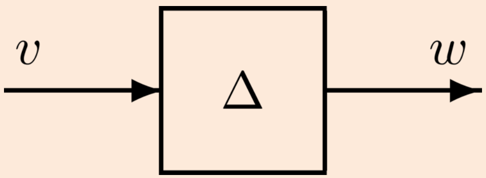
$$J := (\cdot)^\top \begin{bmatrix} -2\alpha\beta & \alpha+\beta \\ \alpha+\beta & -2 \end{bmatrix} \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

Define Ψ as the system shown above. It contains delays to store $v(t-1)$ and $w(t-1)$. Then, Δ satisfies the IQC defined by (Ψ, J).

- This is called the "off-by-one" IQC [Lessard, Recht, Packard, 2016].
- This leads to the more general Zames-Falb IQC [Carrasco, et al, 2019; Scherer, 2022; Zames, Falb, 1968].

Slope-Restricted Nonlinearity

Wake-up Problems



Suppose Δ is a nonlinearity, $w = f(v)$, whose slope lies in $[\alpha, \beta] := [0,1]$ and $f(0) = 0$.

1) Write a quadratic constraints on $[v(t), w(t)]^\top$ representing the sector constraint at time t.

2) Write a quadratic constraints on $[v(t-1), w(t-1)]^\top$ representing the sector constraint at time $t-1$.

3) Write a quadratic constraints on $[v(t), v(t-1), w(t), w(t-1)]^\top$ representing the slope constraint at times t and $t-1$.

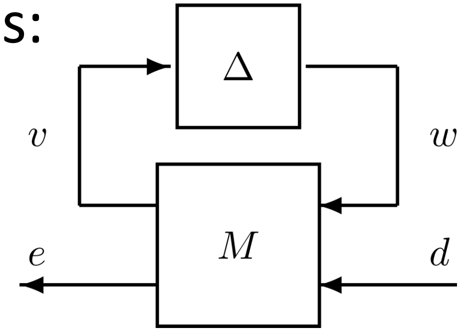
4) Write a general QC formed by the conic combination of the QCs created in parts a)-c). Note that you can scale QC i by a non-negative constant λ_i for $i = 1,2,3$.

Constructing storage functions using IQCs

The analysis procedure consists of the following steps:

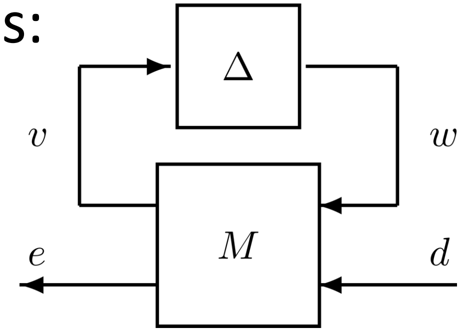
1. Express the uncertain system as an LFT $F_U(M, \Delta)$ with the uncertainty/nonlinearity in Δ.

2. Specify an IQC (J, Ψ) for Δ. This bounds the Input/output characteristics of Δ.



Constructing storage functions using IQCs

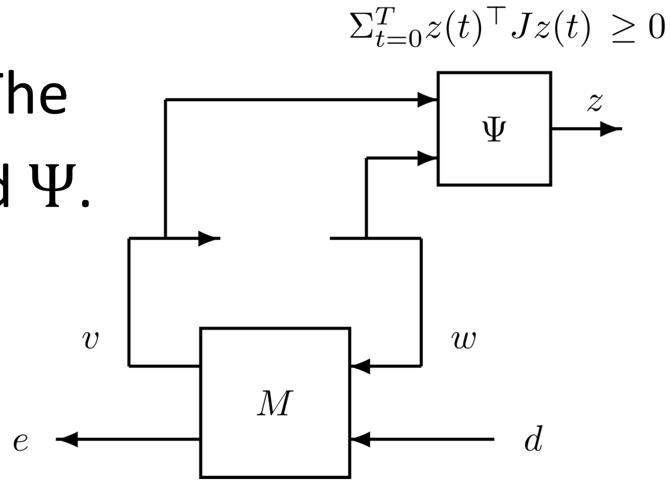
The analysis procedure consists of the following steps:



1. Express the uncertain system as an LFT $F_U(M, \Delta)$ with the uncertainty/nonlinearity in Δ.

2. Specify an IQC (J, Ψ) for Δ. This bounds the Input/output characteristics of Δ.

3. Append the IQC dynamics to the system. The appended system has the dynamics of M and Ψ.



$$\begin{bmatrix} x_e(t+1) \\ z(t) \\ e(t) \end{bmatrix} = \begin{bmatrix} \mathcal{A} & \mathcal{B}_1 & \mathcal{B}_2 \\ \mathcal{C}_1 & \mathcal{D}_{11} & \mathcal{D}_{12} \\ \mathcal{C}_2 & \mathcal{D}_{21} & \mathcal{D}_{22} \end{bmatrix} \begin{bmatrix} x_e(t) \\ w(t) \\ d(t) \end{bmatrix}$$

4. Write a dissipation inequality on the appended system exploiting the IQC. (See next slide.)

Note: Multiple uncertainties/nonlinearities can be combined into $\Delta = \text{diag}(\Delta_1, \dots, \Delta_n)$ and each block can have multiple IQCs.

Constructing storage functions using IQCs

The appended system has the form:
$$\begin{bmatrix} x_e(t+1) \\ z(t) \\ e(t) \end{bmatrix} = \begin{bmatrix} \mathcal{A} & \mathcal{B}_1 & \mathcal{B}_2 \\ \mathcal{C}_1 & \mathcal{D}_{11} & \mathcal{D}_{12} \\ \mathcal{C}_2 & \mathcal{D}_{21} & \mathcal{D}_{22} \end{bmatrix} \begin{bmatrix} x_e(t) \\ w(t) \\ d(t) \end{bmatrix}$$

Suppose there is a storage function $V(x_e) = x_e^\top P\, x_e$ with $P \geq 0$ such that the dissipation inequality (DI) holds along trajectories:

$$V(x_e(t+1)) - V(x_e(t)) + \begin{bmatrix} e(t) \\ d(t) \end{bmatrix}^\top \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} \begin{bmatrix} e(t) \\ d(t) \end{bmatrix} + z(t)^\top J z(t) \leq 0$$

Constructing storage functions using IQCs

The appended system has the form: $\begin{bmatrix} x_e(t+1) \\ z(t) \\ e(t) \end{bmatrix} = \begin{bmatrix} \mathcal{A} & \mathcal{B}_1 & \mathcal{B}_2 \\ \mathcal{C}_1 & \mathcal{D}_{11} & \mathcal{D}_{12} \\ \mathcal{C}_2 & \mathcal{D}_{21} & \mathcal{D}_{22} \end{bmatrix} \begin{bmatrix} x_e(t) \\ w(t) \\ d(t) \end{bmatrix}$

Suppose there is a storage function $V(x_e) = x_e^\top P\, x_e$ with $P \geq 0$ such that the dissipation inequality (DI) holds along trajectories:

$$V(x_e(t+1)) - V(x_e(t)) + \begin{bmatrix} e(t) \\ d(t) \end{bmatrix}^\top \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} \begin{bmatrix} e(t) \\ d(t) \end{bmatrix} + z(t)^\top J z(t) \leq 0$$

Summing from $t = 0$ to $t = T$ yields:

$$\underbrace{V(x_e(T+1))}_{\geq 0} - V(x_e(0)) + \Sigma_{t=0}^{T}\, e(t)^\top e(t) + \underbrace{\Sigma_{t=0}^{T}\, z(t)^\top J z(t)}_{\geq 0} \leq \gamma^2 \Sigma_{t=0}^{T}\, d(t)^\top d(t)$$

If $x_e(0) = 0$, $d \in \ell_2$ then we can let $T \to \infty$ to obtain $||e||_2 \leq \gamma ||d||_2$.

The DI + IQC verifies the uncertain system $F_U(M, \Delta)$ has ℓ_2 gain $\leq \gamma$.

With a few additional technical details, we can prove $x_e(t) \to 0$.

Constructing storage functions using IQCs

The appended system has the form: $\begin{bmatrix} x_e(t+1) \\ z(t) \\ e(t) \end{bmatrix} = \begin{bmatrix} \mathcal{A} & \mathcal{B}_1 & \mathcal{B}_2 \\ \mathcal{C}_1 & \mathcal{D}_{11} & \mathcal{D}_{12} \\ \mathcal{C}_2 & \mathcal{D}_{21} & \mathcal{D}_{22} \end{bmatrix} \begin{bmatrix} x_e(t) \\ w(t) \\ d(t) \end{bmatrix}$

Suppose there is a storage function $V(x_e) = x_e^\top P\, x_e$ with $P \geq 0$ such that the dissipation inequality (DI) holds along trajectories:

$$V(x_e(t+1)) - V(x_e(t)) + \begin{bmatrix} e(t) \\ d(t) \end{bmatrix}^\top \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} \begin{bmatrix} e(t) \\ d(t) \end{bmatrix} + z(t)^\top J z(t) \leq 0$$

This DI can be expressed as an LMI:

$$(\cdot)^\top P \begin{bmatrix} \mathcal{A} & \mathcal{B}_1 & \mathcal{B}_2 \end{bmatrix} - \begin{bmatrix} P & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + (\cdot)^\top \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} \begin{bmatrix} \mathcal{C}_2 & \mathcal{D}_{21} & \mathcal{D}_{22} \\ 0 & 0 & I \end{bmatrix}$$

$$+ (\cdot)^\top J \begin{bmatrix} \mathcal{C}_1 & \mathcal{D}_{11} & \mathcal{D}_{12} \end{bmatrix} \preceq 0$$

Summary

In this lesson:

- We introduced sum-of-squares (SOS) optimization.
- We merged SOS methods with our dissipation inequality/IQC formalism to assess the stability and performance of polynomial systems. This included results for nonlinear reachability.
- We generalized our dissipation inequality / IQC results to systems that are linear time-varying (LTV) or discrete-time.

Next lesson: Application of the methods to optimization algorithms and games.

Further Reading

Sum of Squares

- Lasserre, Global optimization with polynomials and the problem of moments, SIAM Journal on optimization, 2001.
- Parrilo, Semidefinite programming relaxations for semialgebraic problems, Mathematical programming, 2003.
- Balas, Packard, Seiler, Topcu, Robustness analysis of nonlinear systems, ACC Workshop, 2009.
- Summers, Chakraborty, Tan, Topcu, Seiler, Balas, Packard, Quantitative local L2-gain and Reachability analysis for nonlinear systems, IJRNC, 2013.
- Iannelli, Seiler, Marcos, Region of attraction analysis with integral quadratic constraints, Automatica, 2019.
- Papachristodoulou, Anderson, Valmorbida, Prajna, Seiler, Parrilo, Peet, Jagt, SOSTOOLS: Sum of squares optimization toolbox for MATLAB, 2013.

Further Reading

IQCs for LTV Systems

- Jönsson, Robustness of trajectories with finite time extent, Automatica, 2002.
- Petersen, Ugrinovskii, Savkin, Robust control design using H_∞ methods, Springer, 2012.
- Moore, Finite Horizon Robustness Analysis using IQCs, MS Thesis, Berkeley, 2015.
- Seiler, Moore, Meissen, Arcak, Packard, Finite Horizon Robustness Analysis of LTV Systems Using IQCs, Automatica, 2019.
- Biertümpfel, Theis, Pfifer, Robustness analysis of nonlinear systems along uncertain trajectories, IFAC, 2023.
- Biertümpfel, Pholdee, Bennani, Pfifer, Finite Horizon Worst Case Analysis of Linear Time-Varying Systems Applied to Launch Vehicle, TCST, 2023.
- Seiler, Venkataraman, Trajectory-based robustness analysis for nonlinear systems, IJRNC, 2024.

Further Reading

Nonlinear Reachability Analysis

- Yin, Arcak, Packard, Seiler, Backward reachability for polynomial systems on a finite horizon, IEEE Trans. Automatic Control, vol.66, no.12, pp. 6025-6032, 2021.
- Yin, Seiler, Arcak, Backward reachability using integral quadratic constraints for uncertain nonlinear systems, IEEE L-CSS, vol.5, no.2, pp. 707-712, 2021.
- Yin, Packard, Arcak, Seiler. Reachability analysis using dissipation inequalities for uncertain nonlinear dynamical systems, Systems and Control Letters, vol.142, pp. 104736, 2020.

Further Reading

Discrete-Time IQCs

- Hu, Lacerda, Seiler, Robustness analysis of uncertain discrete-time systems with dissipation inequalities and integral quadratic constraints, IJRNC, 2017.
- Jaoude, Farhood, Customized analytic center cutting plane methods for the discrete-time integral quadratic constraint problem, 2022.
- Fry, Farhood, Seiler, IQC-based robustness analysis of discrete-time linear time-varying systems, IJRNC, 2017.
- Palframan, Fry, Farhood, Robustness analysis of flight controllers for fixed-wing unmanned aircraft systems using integral quadratic constraints, TCST, 2017.
- Lessard, Recht, Packard, Analysis and Design of Optimization Algorithms via Integral Quadratic Constraints, SIAM, 2016.
- Carrasco, Heath, Zhang, Ahmad, Wang. Convex searches for discrete-time Zames–Falb multipliers. IEEE Transactions on Automatic Control, 2019.
- Zames, Falb, Stability conditions for systems with monotone and slope-restricted nonlinearities, SIAM, 1968.
- Scherer, Dissipativity, convexity and tight O'Shea-Zames-Falb multipliers for safety guarantees, IFAC, 2022.

Self-Study Problems

See Web site for problems and solutions.

sites.google.com/berkeley.edu/dissipation-iqc