# Lecture 10: Lyapunov/Sums-of-Squares for Control.

$V(x) > 0 \qquad \dot{V}(x) < 0$

$$\frac{\partial V}{\partial x} f(x)$$

If we choose Lyapunov candidates which are polynomials and our dynamics is polynomial
(i.e., if $V(x)$ polynomial
$f(x)$ polynomial)

then Sum-of-squares (SoS) optimization.

But,
Eqn of motion

Simple pendulum

$ml^2\ddot{\theta} + b\dot{\theta} + mgl\sin\theta = 0$ is not polynomial in the original coordinates.

they are polynomial, but they are not actually polynomial in original coordinates

$\sin\theta_i \rightarrow s_i$ (new decision variable)

$\cos\theta_i \rightarrow c_i$

In order to make sure, these are related, add another constraint

$$s_i^2 + c_i^2 = 1$$

Note
The eqn we get out of our mechanical systems have sines and cosines that enter in a particular way.
We can get $\sin(\theta_1 + \theta_2)$ which we can break into polys. of sin & cos $\theta_1$ or $\theta_2$ with trigonometry.

In pendulum

$x = \begin{bmatrix} s \\ c \\ \dot{\theta} \end{bmatrix}$

Coordinate System

$$\dot{x} = f(x) = \begin{bmatrix} c\dot{\theta} \\ -s\dot{\theta} \\ \dfrac{b\dot{\theta} + mgls}{ml^2} \end{bmatrix}$$

(Still polynomial in x coordinates)

These special structures allows us to do above substitution $\sin\theta_i \rightarrow s_i$, $\cos\theta_i \rightarrow c_i$ $\leftarrow$ But, we will never get something like $\sin 10$ or $\sin\dot{\theta}$, etc. These trigonometric terms are direct function of one variable which is $\theta$.

Screw joints are not polynomial because they mix translation and rotation in the same coordinate.

find (P, $\alpha$) s.t. $\longrightarrow$ decision variables

$$V = m^T(x) \, P \, m(x)$$

$\hookrightarrow$ monomial is a basis of $[1, s, c, \dots]$

sine cosine

$$-\dot{V}(x) + \lambda_\alpha(x)(s^2+c^2-1)$$

is SoS.

x are indeterminate

another polynomial

$V(0) = 0$ (to set the scaling).

S-procedure

$g(x) \le 0 \Rightarrow$ ___

$g(x) = 0 \Rightarrow$ ___

(s, c are not decision variable, these are indeterminates they are never actually handed to the solver)

* A decision variable has to take on a particular value at the optimal solution whereas an indeterminant is a quantity we want to hold for all values — undetermined value / indeterminate value.

$s^2+c^2=1$ is not a constraint. It makes the problem easier in some sense but it's a language that says when $s^2+c^2=1$ that implies $\dot{V}(x)$ is −ve

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} = \tau_g(q) + \cancel{Bu}^{\,0}$$

$M(q)$, $C(q,\dot{q})$, $\tau_g$ are polynomial in $\dot{q}$, s, c

$$\dot{x} = f(x) = \begin{bmatrix} \dot{q} \\ M^{-1}(q)[\tau_g(q) - C\dots] \end{bmatrix}$$

inverse makes it a rational polynomial

In the scalar case

$$\frac{P_1(x)}{P_2(x)}$$

$\dot{x} = f(x)$ — an explicit way to write eqn of motion when the derivative is an explicit function of the state.

more general description of system of equation

$g(x,\dot{x}) = 0 = f(x) - \dot{x}$ (implicit form of dynamics)

$M(q)\ddot{q} + C(q,\dot{q})\dot{q} = \tau_g(q) + \cancel{Bu}^{0}$

(implicit form of dynamics)

$$\dot{x} = f(x) = \begin{bmatrix} \dot{q} \\ M^{-1}(q)[\tau_g(q) - C \ldots] \end{bmatrix}$$

explicit form of dynamics

$g(q,\dot{q},\ddot{q}) - M(q)\ddot{q} - C - \tau_g - \cancel{Bu}^{0} = 0$ (implicit form of dynamics)

## Lyapunov conds. in implicit form

$V(x)$ is SoS

Indeterminates $x, z$ $\quad -\frac{\partial V}{\partial x} z$ is SoS

$\uparrow$ placeholder for $\dot{x}$

**Quotient Ring**

$g(x,z) = 0 \Rightarrow -\frac{\partial V}{\partial x} z$ is SoS.

$-\frac{\partial V}{\partial x} z + \lambda^T(x) g(x,z)$ is SoS.

finally reveals the fundamental structure of the eqn of motion in our optimization.

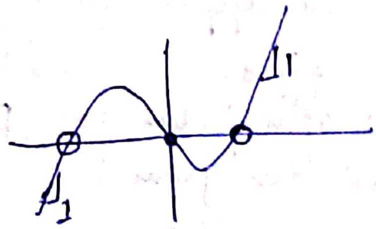
polynomial of some fixed degree

**Example**

Eq. of motion

$$\dot{x} = \frac{-x + x^3}{1 + x^2}$$

Rational polynomial func.



when $g(x,z) = 0$

$\Rightarrow \frac{\partial V}{\partial x} z > 0$

## Lyapunov Control (we are going to design controllers and Lyapunov function that ~~xxxx~~ proves it's stable)

$\dot{x} = f_1(x) + f_2(x)u$ (dynamics are control affine)

fine assumptions for multibody system

find a controller
$u = K(x)$ (polynomial in $x$), $V(x)$
that makes above system
satisfy the Lyapunov ~~xxxx~~ condition in closed loop.

$K_\alpha(x)$, $V_\beta(x)$

Scalar ~~vector~~ polynomial

Vector of polynomial

Coeff. vector $\alpha$ & $\beta$ that we are trying to search over.

$$\underset{\alpha,\beta}{\text{find}} \quad V_\beta(x) > 0$$

$$\dot{V} = \frac{\partial V_\beta}{\partial x}\left[f_1(x) + f_2(x) K_\alpha(x)\right] < 0 \quad \text{is SoS}$$

requires linear in decision variable.

we want this to be a polynomial linear in decision variables $\alpha$ & $\beta$.

Non-convex constraint (bad) since this will have terms like $\alpha_1\beta_2, \ldots$ which ~~xxxx~~ are not jointly convex in $\alpha$ & $\beta$.

Above problem is Bilinear in decision variables

$\alpha_i B_j(\ldots)$
indeterminants

A standard approach would be ~~to~~ alternation.

- Fix $V_\beta(x)$, optimize $K_\alpha(x)$
- Fix $K_\alpha(x)$, optimize $V_\beta(x)$

repeat until convergence

(if $\alpha$ is fixed, then remaining term is convex in $\beta$ and vice-versa)

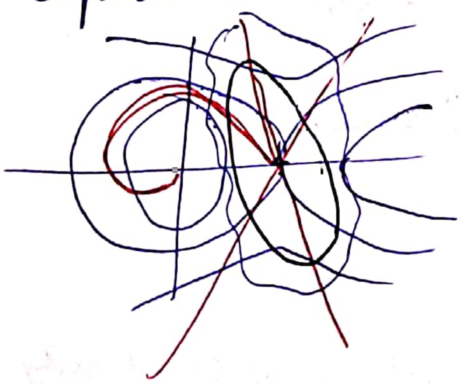
(if our controller is fixed, we know that the optimization is convex for looking for Lyapunov functions and similarly if our Lyapunov func is fixed, the problem of finding good K's that satisfy that is convex in parameters of K.)

- Can have local minima.
- Recursive feasibility (once we find a Lyapunov/Controller, they will only improve).
- Monotonic improvement

(if we start with an initial ~~K~~ and optimize K.
~~eg~~ our initial guess at K is already a feasible solution
So, optimizing we can only do better
at each of these steps.

✓ we can get a guess at K, try to find a better Lyapunov function maybe one that goes downhill faster, for instance.
we can only do better at each of these steps.)

☆ Hard part is we have to find an initial Lyapunov function that is feasible. once we have that we can expect monotonic improvement until convergence.



powerful recipe

- Linearize nonlinear dynamics at some fixed point $x^*, u^*$
- Solve LQR: gives $u = -Kx$, $\hat{J} = x^T S x$ (cost-to-go)

If we can compute
ROA using SoS of $u = -Kx$ controller
with $\hat{V} = x^T S x$ ~~$\hat{J}$~~ lyapunov guess,
Now start Alternation to
Search $K(x), V(x)$ to prove bigger RoA.

offline step

$u = Kx + \alpha x^2 + \beta x^3$

# Sums-of-Squares DP

## Recall L.P.DP (Linear Programming for DP)

Discrete setting. $\forall s$ $\quad J(s) = \min_a \left[ l(s,a) + J(f(s,a)) \right]$

discrete state

$\Downarrow$ L.P. formulation

$$\forall s, \forall a \quad J(s) \le l(s,a) + J(f(s,a))$$

$$\max \sum_s J(s)$$

## New in Continuous time state action

$$\forall s \quad 0 = \min_u \left[ l(s,a) + \frac{\partial J}{\partial x} f(x,u) \right]$$

$$\forall x \, \forall u \quad 0 \le l(x,u) + \frac{\partial \hat{J}}{\partial x} f(x,u)$$

$$\Rightarrow l(x,u) + \frac{\partial \hat{J}}{\partial x} f(x,u) \text{ is SOS.}$$

$$\max \int_x \hat{J}(x)\, dx \qquad \hat{J} \Rightarrow \text{estimate}$$

---

$\dot{x} = -x + x^3$ (origin is already stable, so not an interesting control problem)

$\dot{x} = x - 4x^3 + u$ (origin is unstable)

## HJB vs Lyapunov

$\dot{J} = -l(x, u^*)$ $\qquad \dot{V} \le 0$

now,

(Not a convex optimization)

$\dot{J} \le -l(x, u^*)$ (upper bound on cost-to-go and certify Lyapunov cond.)

But what we did above $\dot{J} \ge -l(x, u^*)$ (lower bound)

(Convex optimisation) minimize $u$ can be replaced by all $u$.