Lecture 9: Computing Lyapunov Functions II

Big picture from Lyapunov (the Computational perspective of Lyapunov)

$$\frac{dJ}{dt} = -l(x, u^*) \quad \Rightarrow \frac{d}{dt} V(x) < 0$$

First version – Pick a bunch of sample points x_i's and say

$\underline{\forall_i \; x_i}$ $\dot{V}(x_i) < 0$

(Some number of samples) (enough for the simple case of pendulum)

Convex optimization version $\underline{\forall x}$ $V(x) > 0$ $\dot{V}(x) < 0$

(all x)

Linear system

$$\dot{x} = Ax$$

$$V = x^T P x \qquad P > 0$$

$$\dot{V} = x^T P A x + x^T A^T P x < 0$$

Search for P as a semi-definite program.

→ optimization that fits into the class of semi-definite programming are convex optimization (If a solution exists, then guaranteed to be found by the solver except for the numerical limits or running out of memory. etc).

→ Above extends beautifully to the non-linear systems

General form $V = m^T(x) P m(x)$, $m(x) \Rightarrow$ non-linear basis function

In ML → Kernel trick.

Linear function Approximator

$$\sum_i \alpha_i \phi_i(x) = \alpha^T \vec{\phi}(x)$$

Sums-of-Squares

(Still linear in the parameters) $\sum_i \alpha_i \phi_i^2(x)$ $\quad m^T(x) P m(x)$, $P \succ 0$

(more dimension way to write this)

A general tool for polynomial optimization

$p(x) \geq 0 \;\forall x$ polynomial $\quad p(x) = m^T(x) P m(x)$, $P \succ 0$

(Coeff. matching and hand it to a convex optimisation)

No matter what $m(x)$ is, if $P \succ 0$ (positive definite matrix), then the quadratic form $m^T(x) P m(x) \geq 0$ (it's sum of squares)

(this is what we need to satisfy the Lyapunov condition)

$$P_\alpha(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \dots$$

(Coeff. matching gives us eqns that are linear in the decision variables and turns out it just adds linear equality constraints and we can search over the polynomials).

How does this work for multivariable polynomial?

Every positive univariate polynomial has a sum-of-squares decomposition and that's not true in the multivariate case. But, the one direction is true. If we can find a sum-of-square decomposition then we are guaranteed that it's positive. It's the converse that's not true.

$$\min_{x} p(x)$$

$$\max_{\gamma, P} \gamma \quad \text{s.t.}$$

gamma

$\underbrace{p(x) - \gamma}_{P_2(x)}$ is SOS (sums-of-squares)

$p(x) \geq \gamma$

Total decision variable

decision variable

$P_2(x)$ is SOS

$P_2(x) = m_2^T(x) G m(x), \; G \succeq 0$

γ, P (matrix being positive definite).

Example Global Stability

2D system $\dot{x} = \begin{bmatrix} -x_1 - 2x_2^2 \\ -x_2 - x_1 x_2 - 2x_2^3 \end{bmatrix}$

(1) certifying Lyapunov cond.

$V(x) = x_1^2 + 2x_2^2$

Newton Polytopes

$\forall x \in D$ (some domain), $p(x) \geq 0$

$D : \{x \mid g(x) \leq 0\}$

also polynomial

multiplier polynomial

$p(x) + \lambda^T(x) g(x)$ is SOS

famously called S-procedure: $\lambda(x)$ is SOS

$\lambda^T(x) g(x)$ is only making $p(x)$ looks more negative

In the region of interest where we are trying to certify, $g(x)$ is negative ($g(x) < 0$) which means having $g(x)$ in $[p(x) + \lambda^T(x) g(x)]$ only making it harder for ~~LHS~~ L.H.S. to be SOS especially $\lambda(x)$ is guaranteed to be true.

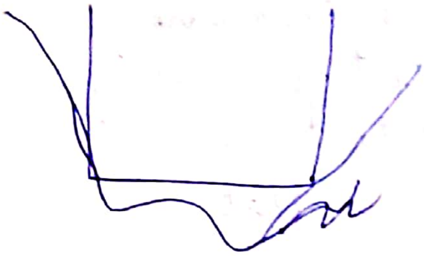
So, compared to $p(x)$ inside the region $\lambda^T(x) g(x)$ is making it harder to be Sum. positive.

So, if $p(x) + \lambda^T(x) g(x) \ni$ SOS is true then $\forall x \in D, \; p(x) \geq 0$

What's interesting is when it's outside the region and $g(x)$ is ~~negative~~ +ve, $p(x)$ could be -ve and $\lambda^T(x)g(x)$ can help us remove the -ve.

So, $\boxed{\lambda^T(x)g(x)}$ helps us when outside the region & hurts us when inside the region.

Lagrange's multiplier

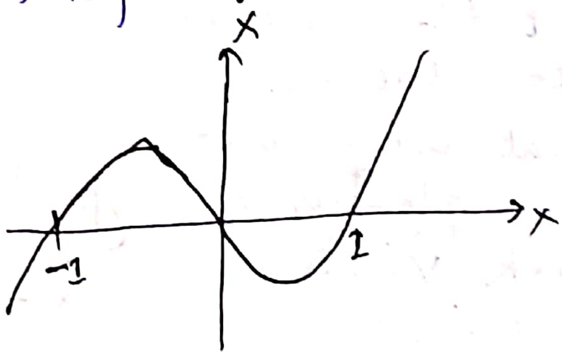


What if $D\{x \mid g(x)=0\}$.

$p(x)+\lambda^T(x)g(x)$ is SoS

Now, $\lambda(x)$ is free.

Ex $\dot{x}=-x+x^3$



$V(x)=x^2$

$V(x)\leq 1 \Rightarrow \dot{V}(x)\leq 0$

Make a multiplier $-\dot{V}(x)+\lambda(x)(V(x)-1)$ is SoS.

$\lambda(x)$ is SoS.

Conditions for Region of Attraction — Have to be an invariant set in addition to being a region.

So, if we are certified over some region in space ($\dot{V}$) that wouldn't say it's a region of attraction yet. We also have to show that the sets is an invariant set. So the choices of domain for RoA are almost always sublevel subsets of V because then they are self consistent. Once we are inside V and we have certified the condn inside V, we won't leave V.

S-procedure example

$V(x) = x^2$

$V(x) < 1 \Rightarrow \dot{V}(x) \le 0$

$-\dot{V}(x) + \lambda(x)(V(x) - \rho)$ is SoS

$\lambda(x)$ is SoS. $\rho = 1$

Trying to find ways to write conditions that certify something about the dynamics in a way that is compatible with our convex optimization toolbox.

$\lambda(x)(V(x) - \rho)$

Let's take the example of balancing an acrobot at the top. We compute an LQR controller. We know it's going to work around the linearization but if it gets too far, it's going to fall down. So, asking for that cost-to-go to be a Lyapunov for globally is not going to work. $\dot{V}(x)$ is going to be increasing in some places. We want to say that at least in some invariant sub-level set, $\dot{V}(x)$ is going downhill. That would give us a Region of Attraction. That's why, we have to do above multiplier [$\lambda(x)(V(x) - \rho)$] trick.

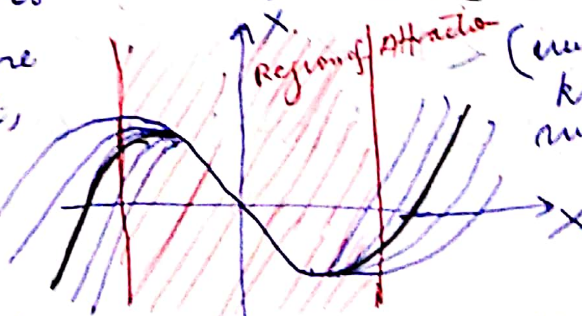
$\dot{V}(x)$ only must be < 0 inside this region.

Gain Uncertainty

$\dot{x} = -x + \Delta x^3$,

$\Delta \in [.5, 1.5]$

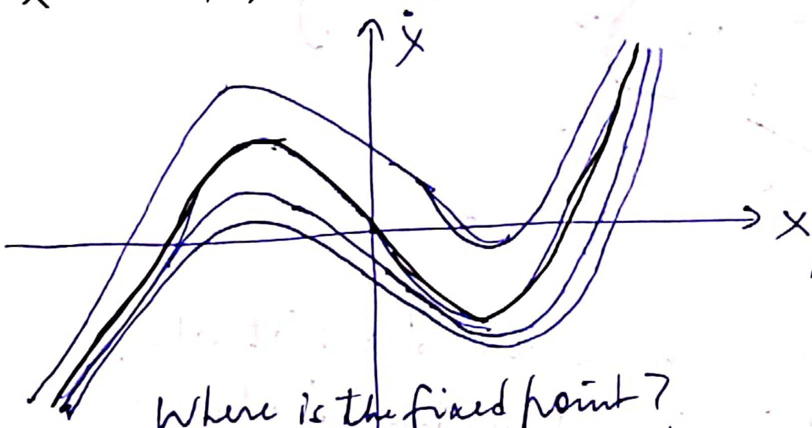
(we don't know what Δ is)



It could be anyone of the above curve. But, if we find a common Lyapunov function that works for all Δ in that region which we can do with another S-procedure, we can have Δ be an indeterminate an put another S-procedure to say for all α for all x in some region, then we can find the robust region of attraction with SoS program.

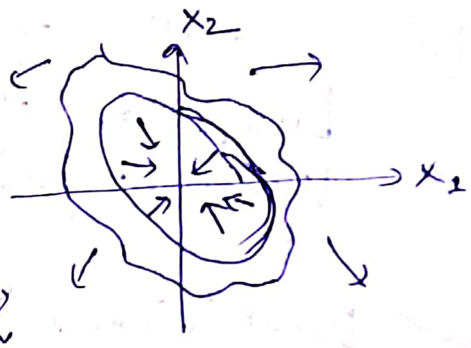
Additive Uncertainty

$$\dot{x} = -x + x^3 + \Delta, \quad \Delta \in [-.25, .25]$$



No RoA to the origin because depending on the parameters, the fixed point moves.

Where is the fixed point?



$$V(x) \leq \rho \Rightarrow \dot{V}(x) < 0$$

If we relaxed our condition to say,

$$V(x) = \rho \Rightarrow \dot{V}(x) < 0$$

Similar to this

At least on the boundary, we are going in. we are not saying we are going all the way to the origin. we are just saying it's an invariant set. Everytime we get to $V(x) \leq \rho$, we know we are pointing in. And we can find the largest invariant set for a set even if we don't know where the fixed point is.

Ex 2D (vanderpaul oscillator) Non-linear System

$\ddot{q} = -q - (q^2 - 1)\dot{q}$

2nd order system

phase portrait

great example for ROA analysis ~~[illegible]~~

Note – Not all +ve polynomials are SoS. Not all Lyapunov polynomial system have polynomial Lyapunov function.

What is the ROA of the linear controller on the non-linear system?

Robots aren't polynomial. They are Sin & Cosin.

Rigid body dynamics are (rational) polynomials.

$ml^2\ddot{\theta}$ ~~[illegible]~~ $+ b\dot{\theta} + mgl\sin\theta = u$

s (substitution)

$\cos\theta \rightarrow c$

eqn will be polynomial in s & c

$D \; \{ X \mid s^2 + c^2 = 1 \}$

$x_1 \quad x_2$

$\|x_1 - x_2\|^2 = c$

(rigid body)

In general,

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

poly in Sin & cos

poly. in Sin, cos & $\dot{\theta}$

But, ~~[illegible]~~ mass matrix inverse will ~~convert~~ turn it from poly. eqn to rational polynomial eqn.

So, we can operate directly on sine & cosine. It is polynomial as long as we avoid mass matrix inverse