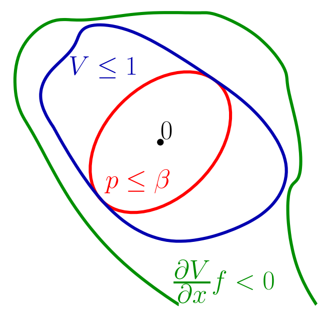
CDS 270 (Fall 09) - Assignment 5 (Due Friday, Oct. 30)

Consider a dynamical system governed by $\dot{x} = f(x)$ where $x \in \mathbb{R}^n$ and $f \in \mathbb{R}[x]$. Let's assume that $f(0) = 0$ and the origin is a locally, asymptotically stable equilibrium point for $\dot{x} = f(x)$. Let $l_1(x) := \epsilon x^T x$, $l_2(x) := \epsilon x^T x$, and $p \in \mathbb{R}[x]$ be positive definite. We proposed the following optimization problem to increase the "size" of the estimate of the region of attraction around the origin.

$$\max_{V \in \mathbb{R}[x],\ \beta \in \mathbb{R}} \beta$$

subject to:

$$\Omega_{p,\beta} \subseteq \Omega_{V,1}$$
$$\Omega_{V,1} \subseteq \{\nabla V \cdot f(x) \le -l_2(x)\} \cup \{0\}$$
$$V - l_1 \in \Sigma[x], V(0) = 0$$



where $\Omega_{p,\beta}$ is the β-sublevel set of p and $\Omega_{V,1}$ is the 1-sublevel set of V.

Then, we replaced the problem by an S-procedure and sum-of-squares relaxation whose solution provides a lower bound on the optimal value of β obtained from the problem above.

$$\max_{s_1, s_2 \in \Sigma[x],\ V \in \mathbb{R}[x],\ \beta \in \mathbb{R}} \beta$$

subject to:

$$-((V-1) + s_1(\beta - p)) \in \Sigma[x]$$
$$-((\nabla V \cdot f + l_2) + s_2(1 - V)) \in \Sigma[x]$$
$$V - l_1 \in \Sigma[x], V(0) = 0,$$

Even though this problem is a sum-of-squares programming problem, it cannot be directly solved using, say, SeDuMi because some constraints are not affine in the decision variables. We have discussed the following iterative procedure to compute sub-optimal solutions for the SOS program above.

Initialization: Find V which proves local stability in some neighborhood of $x = 0$.

1. γ Step: Hold $V(x)$ fixed and use `pcontain` to solve for s_2:

$$\gamma^* = \max_{s_2 \in \Sigma[x], \gamma \in \mathbb{R}} \gamma \quad \text{s.t.} \ -(\nabla V \cdot f + l_2 + s_2(\gamma - V)) \in \Sigma[x]$$

2. β Step: Hold $V(x)$ fixed and use `pcontain` to solve for s_1:

$$\beta^* = \max_{s_1 \in \Sigma[x], \beta \in \mathbb{R}} \beta \quad \text{s.t.} \ -((V - \gamma) + s_1(\beta - p)) \in \Sigma[x]$$

3. <u>V step</u>: Hold $s_1, s_2, \beta^*, \gamma^*$ fixed and solve the following feasibility problem using `sosopt`:

$$
\begin{aligned}
&-\left(\frac{\partial V}{\partial x} f + l_2 + s_2(\gamma - V)\right) \in \Sigma[x] \\
&-((V-\gamma) + s_1(\beta - p)) \in \Sigma[x] \\
&V - l_1 \in \Sigma[x], V(0) = 0
\end{aligned}
$$

4. <u>V Scaling</u>: Replace V with $\frac{V}{\gamma^*}$.

5. Repeat until a pre-specified stopping criterion is satisfied.

In this assignment, you will implement this iterative procedure.

Hand in a Matlab script/function that solves the SOS optimization problem in the previous page by using the iterative procedure above. Take the following into consideration in the implementation.

- Use a quadratic Lyapunov function for the linearized dynamics in the initialization step. You may want to check the function `linstab`.
- Repeat the iterations until a maximum number of iterations is reached or the increase in β^* from an iteration to the next one becomes less than a pre-specified tolerance.
- Use $\epsilon = 10^{-6}$ in the definition of l_1 and l_2.

Questions: Use the Van der Pol (VDP) dynamics

$$
\begin{aligned}
\dot{x}_1 &= -x_2 \\
\dot{x}_2 &= x_1 + (x_1^2 - 1)x_2.
\end{aligned}
$$

- Test your script/function on the following cases. Take $p(x) = x^T x$. Use 25 as the maximum number of iterations and $5 \cdot 10^{-3}$ as the tolerance for the increase in β^*.

	Initialization	$deg(V)$	$deg(s_1)$	$deg(s_2)$
(a)	quadratic V for linearization	2	0	2
(b)	quadratic V for linearization	4	2	2
(c)	quadratic V for linearization	4	2	4
(c)	optimal V from part(a)	4	2	4
(e)	quadratic V for linearization	6	4	2
(f)	quadratic V for linearization	6	4	4
(g)	optimal V from part(c)	6	4	4

For a polynomial π, deg(π) in the above table means that you should include all appropriate monomials up to degree deg(π) in forming π.

- Plot the ("largest" - measured by the sets $\{x : p(x) \leq \beta\}$) sublevel set of V for cases (a), (b), (c), (e), and (f). Overlay all of them on the same plot. Try to use different colors and/or different line types. Remember to use `pcontour`. (You can pull out two outputs from `pcontain`. One of them is the handle to the contour object. You can play with the line types using this handle.)

- Find the limit cycle for the VDP dynamics and plot it on the figures in the previous part. This limit cycle is the boundary of the actual region of attraction.

- Is there any difference

  - between the optimal values of β in (b) and (c)
  - between the optimal values of β in (e) and (f)?

  If there is any difference, can you explain the difference?