6.832 Midterm

Name: SOLUTIONS

November 3, 2015

Please do not open the test packet until you are asked to do so.

- You will be given 85 minutes to complete the exam.
- Please write your name on this page, and on any additional pages that are in danger of getting separated.
- We have left workspace in this booklet. Scrap paper is available from the staff. Any scrap paper should be handed in with your exam.
- YOU MUST WRITE ALL OF YOUR ANSWERS IN THIS BOOKLET (not the scrap paper).
- The test is open notes.
- The test is out of 37 points. (The 4th problem is omitted from this version.)

Good luck!

Problem 1 (10 pts) *Lyapunov analysis*
Consider the system described by:

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

In this problem we will investigate the stability of the fixed point at $x^* = 1$.

a) *Draw the function* $\dot{x}$ *as a function of* x *in the space below. Circle the fixed points, and label each fixed point as stable or unstable.*

Solution: *Take figure 10.4 from the textbook and horizontally reverse it. Stable fixed points at -1,1, unstable at 0.*

b) *Linearize the system about the fixed point at* $x^* = 1$. *Write your answer in the form:* $\dot{\bar{x}} = A\bar{x}$, *where* $\bar{x} = x - x^*$.

$$\dot{\bar{x}} \approx (1 - 3)\bar{x} = -2\bar{x}$$

c) *Solve the Lyapunov equation* $(PA + A^T P = -Q, P = P^T \succ 0)$ *using* $Q = 1$ *to find a candidate Lyapunov function. Write the resulting Lyapunov function as a polynomial in terms of* x*:*

$$V = \frac{1}{4}\bar{x}^2 = \frac{1}{4}(x-1)^2 = \frac{1}{4}x^2 - \frac{1}{2}x + \frac{1}{4}$$

d) *Compute* $\dot{V}$ *using the original nonlinear dynamics.*

$$\dot{V} = \frac{1}{2}(x-1)(x-x^3) = \frac{1}{2}(x^2 + x^3 - x - x^4)$$

$\dot{V} < 0$ *for all* $x > 0$*, with the obvious exception* $\dot{V}(x = x^*) = 0$.

e) *What initial conditions* x *does this Lyapunov candidate prove are inside the region of attraction of the fixed point? (Hint: all roots of the resulting polynomial are at integer values)*

Unless we invoke our extra knowledge of invariance in the one DOF systems, we can still only argue for a sub-level set of $V \leq \frac{1}{4}$*. So* $(0, 2)$.

f) *Is this estimate of the region of attraction tight? (If no, then demonstrate this by naming another initial condition in the region of attraction which is outside the certified region).*

Not tight. $x = 3$

g) *With the Lyapunov candidate,* $V(\mathbf{x})$ *given, is it possible to certify this region of attraction using a single* ***convex*** *sums-of-squares optimization?*

YES or NO

- *If yes, what are the decision variables (you need not provide the decision variables required to write the SOS program as an SDP)?*
- *If no, then explain why the problem is not convex.*

[You may wish to write down the optimization to make your answer clear]

This follows Example 10.7 in the book – the answer is yes. We'd want to apply the S-procedure here to find a positive definite multiplier polynomial λ to certify that $\dot{V}(x) \succ 0$ where $(\rho - V(x))$ is negative, i.e. where $V(x) < \rho$ that we calculated in a previous part.

$$\begin{aligned} &find_\lambda \\ -\dot{V}(x) - \lambda(x)(\rho - V(x)) &\text{ is SOS} \\ \lambda(x) &\text{ is SOS} \end{aligned}$$

Since only λ is unknown here, this optimization is convex, as it can be converted into an SDP. (It wouldn't be if we didn't know e.g. ρ, as we'd wind up with a bilinear term multiplying $\rho * \lambda$.)

This page intentionally left blank for workspace

Problem 2 (10 pts) *Planar quadrotors*

Consider a planar quadrotor model with two ideal thrusters and a simple model of bluff-body drag and all non-gravity constants set to 1*, given by*

$$\ddot{x} = -\sin\theta(u_1 + u_2) - \dot{x}^2$$
$$\ddot{z} = -g + \cos\theta(u_1 + u_2) - \dot{z}^2$$
$$\ddot{\theta} = -u_1 + u_2.$$

This model has two inputs to control three degrees of freedom; intuitively we should be able to control two of them with feedback linearization (but note that the coupling in the first two equations prevents us from commanding an arbitrary $\ddot{x}_d, \ddot{z}_d$*). Assume* u *is unbounded unless otherwise noted.*

a) *Give a partial-feedback linearizing controller that imposes the closed-loop dynamics*

$$\ddot{\theta} = \ddot{\theta}_d, \quad \ddot{z} = \ddot{z}_d.$$

$$u_2 = \ddot{\theta}_d + u_1$$
$$u_1$$
$$= \frac{\ddot{z}_d + g + \dot{z}^2}{2\cos\theta} - \ddot{\theta}_d$$

b) *Does your feedback controller have any singularities? If so, in what states?*

yes. $\cos\theta = 0$ *(props full sideways can't regulate z)*

Now imagine that you've lost a propellor $(u_2 = 0)$*. You're quadrotor is going down. Fortunately, you have been given the optimal cost-to-go function,* $J^*(\mathbf{x})$*, for a damage-minimizing cost function of the form*

$$J = \int_0^\infty g(\mathbf{x})dt.$$

This time you have to consider your input limits, $0 \leq u_1 \leq u_{max}$.

c) *What is the optimal policy,* $u_1 = \pi^*(\mathbf{x})$ *as a function of* g*,* J^**, and its partial derivatives?*

$\min_{u_1} \left[g(x) + \frac{\partial J}{\partial x}\dot{x} + \frac{\partial J}{\partial z}\dot{z} + \frac{\partial J}{\partial \theta}\dot{\theta} + \frac{\partial J}{\partial \dot{x}}\ddot{x} + ... \right]$

the only terms that include u_1 *are* $\left[-\frac{\partial J^*}{\partial \dot{x}}\sin\theta + \frac{\partial J^*}{\partial \dot{z}}\cos\theta - \frac{\partial J^*}{\partial \dot{\theta}} \right] u_1$

so $u_1 = 0$ *if the inside is positive, and* $u_1 = u_max$ *if the inside is negative*

This page intentionally left blank for workspace

Problem 3 (7 pts) *Scaling up*

As a part of your latest robotic art installation, you've assembled a robot with 284 degrees of freedom, and you want it to be dynamic, but you could only afford 35 actuators. Which of the following techniques from class might you have a hope of applying to your contraption? Justify your answers. (Note: the problem is subjective; your job is to write something for each method to convince us you understand)

a) *Value iteration using barycentric interpolation*

Probably not a good idea – value iteration on a grid (e.g. with barycentric interpolation) will have to approximate your state space with a grid, so getting reasonable resolution will require exponentially many grid points in the dimension of your state space.

b) *Linear Quadratic Regulator (LQR), using a linearization of the system in the vicinity of a fixed point*

This should probably work great – at least, in the vicinity of the fixed point. The algebraic Riccati solve required to compute the LQR controller is reasonably efficient (low-order polynomial time) and should work OK on a problem of this scale.

c) *Partial feedback linearization*

Like LQR, PFL is probably still useful! Consider the general form of either collocated or non-collocated feedback linearization for the manipulator equations (see Chapter 3 of the textbook): computing the new linearized dynamics requires a reasonably large matrix inverse or pseudo-inverse, but those operations again have low-order polynomial complexity (in the ballpark of $O(n^2)$ or $O(n^3)$ depending on algorithm) that ought to be OK for a system of this order.

However, practically, PFL is only a building block for a complete control system (it'll linearize some set of your dynamics, but won't tell you what to do with those linearized dynamics) – and it also doesn't guarantee that the control inputs required to linearize the dynamics are reasonable, which might cause practical problems for your robot.

Just as LQR works consistently and well as long as your linearization is a reasonable approximation of your system dynamics, PFL is going to be sensitive to how your actuators enter into your system dynamics.

d) *Lyapunov's method for linear systems* $(PA + A^T P = -Q, P = P^T \succ 0)$ *This tooling would likely work for a system of this complexity (this problem is starting to reach into the "moderate" size class for SDP solvers), but what it will provide will be of limited use, as your system is almost certainly not linear. Hence, your certificates won't mean very much.*

e) *Sums-of-squares optimization for estimating regions of attraction*

SOS for ROA estimation will be very hard to deploy for a system of this scale. The size of the monomial basis you'll want to employ (and thus the side-length *of your decision variable matrix Q in* $m(x)^T Q m(x), \quad Q \succeq 0$*) will grow in something that looks like* $\left(\frac{d+n}{n}\right)$*, for order* d *and state dimension* n [1]*. Your dynamics are likely to be of really high order, so this scaling will really bite you.*

It's possible that you could reduce your problem size to some degree by replacing polynomial representations of your system dynamics with samples and hoping you can find a relatively low-order Lyapunov function (that is, try to solve this problem with small enough d *that it's still tractable), but such high state dimension will require way too many samples.*

f) *Sums-of-squares optimization for controller design*

Hard for similar reasons to above – plus the added trouble that you controller is now indeterminate, too! You're likely to wind up in an even harder problem class (e.g. an optimization with semidefiniteness constraints, and also bilinear constraints) that will be even harder to solve.

[1] You can save complexity by picking your basis more carefully than taking all polynomials of order d, via e.g. "Basis selection for SOS programs via facial reduction and polyhedral approximations" (Permenter and Parrilo, 2014), but your savings will be problem-dependent.

g) *Nonlinear trajectory optimization using direct collocation and sequential quadratic programming*

This approach will be computationally tractable, but will suffer from the high dimension of the system in another way: nonlinear optimization may have difficulty with getting stuck in local optima.

h) *Linear trajectory optimization using quadratic programming, using a linearization of the system in the vicinity of a fixed point*

This'll be perfectly computational tractable, though it doesn't offer any improvements over LQR (other than allowing more complex, nonlinear cost functions).

This page intentionally left blank for workspace