

# امتحان پایان ترم کنترل بهینه ۱

علی بنی اسد ۹۶۱۰۸۳۷۸

۱۵ تیر ۱۴۰۰

## سوال اول

برای حل این سوال از کد ارسالی درس استفاده شده است ولی برای این سوال تغییراتی اعمال شده است که در ادامه به بررسی آن پرداخته می شود.  
سیستم به فرم زیر نوشته شده است که در کد با استفاده از تابع ode45 شبیه سازی می شود.

$$a = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -0.4x_1^2 - 0.2x_2^2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

برای افزودن قیود صورت سوال از توابع جریمه به شکل Quadratic Extended استفاده شده است. این کار باعث عوض شدن ماتریس همیتونین می شود.

$$g_1(x_2) = x_2 - 0.5$$

$$g_1(u) = u - 0.5$$

$$g_2(x_2) = -x_2 - 0.5$$

$$g_2(u) = -u - 0.5$$

$$G_1(x_2) = \begin{cases} -\frac{1}{g_1(x_2)} & g_1(x_2) \leq \epsilon \\ -\frac{1}{\epsilon} \left( 3 - \frac{3g_1(x_2)}{\epsilon} + \left( \frac{g_1(x_2)}{\epsilon} \right)^2 \right) & g_1(x_2) > \epsilon \end{cases}$$

$$G'_1(x_2) = \begin{cases} \frac{1}{(x_2 - 0.5)^2} & g_1(x_2) \leq \epsilon \\ -\frac{1}{\epsilon} \left( -\frac{3}{\epsilon} + \frac{2x_2 - 1}{\epsilon^2} \right) & g_1(x_2) > \epsilon \end{cases}$$

$$G_2(x_2) = \begin{cases} -\frac{1}{g_2(x_2)} & g_2(x_2) \leq \epsilon \\ -\frac{1}{\epsilon} \left( 3 - \frac{3g_2(x_2)}{\epsilon} + \left( \frac{g_2(x_2)}{\epsilon} \right)^2 \right) & g_2(x_2) > \epsilon \end{cases}$$

$$G'_2(x_2) = \begin{cases} \frac{1}{(x_2 + 0.5)^2} & g_2(x_2) \leq \epsilon \\ -\frac{1}{\epsilon} \left( \frac{3}{\epsilon} + \frac{2x_2 + 1}{\epsilon^2} \right) & g_2(x_2) > \epsilon \end{cases}$$

$$G_1(u) = \begin{cases} -\frac{1}{g_1(u)} & g_1(u) \leq \epsilon \\ -\frac{1}{\epsilon} \left( 3 - \frac{3g_1(u)}{\epsilon} + \left( \frac{g_1(u)}{\epsilon} \right)^2 \right) & g_1(u) > \epsilon \end{cases}$$

$$G'_1(u) = \begin{cases} \frac{1}{(u - 0.5)^2} & g_1(u) \leq \epsilon \\ -\frac{1}{\epsilon} \left( -\frac{3}{\epsilon} + \frac{2u - 1}{\epsilon^2} \right) & g_1(u) > \epsilon \end{cases}$$

$$G_2(x_2) = \begin{cases} -\frac{1}{g_2(u)} & g_2(u) \leq \epsilon \\ -\frac{1}{\epsilon} \left( 3 - \frac{3g_2(u)}{\epsilon} + \left( \frac{g_2(u)}{\epsilon} \right)^2 \right) & g_2(u) > \epsilon \end{cases}$$

$$G'_2(u) = \begin{cases} \frac{1}{(u + 0.5)^2} & g_2(u) \leq \epsilon \\ -\frac{1}{\epsilon} \left( \frac{3}{\epsilon} + \frac{2u + 1}{\epsilon^2} \right) & g_2(u) > \epsilon \end{cases}$$

توابع جریمه دارای ضریب نیز هستند که در هر دوره محاسبات به روز می شود البته در کد حداقل ضریب 0.001 در نظر گرفته شده است که از این مقدار کوچکتر نشود.

$$\epsilon = -c(r_k)^2, \quad a = 0.5, \quad r_{k+1} = cr_k, \quad c = 0.9, \quad \min(r_k) = 0.001$$

$$\mathcal{H} = \vec{P}^T a(\vec{X}, u, t) + \frac{1}{2} (x_1^2 + x_2^2 + u^2 + r_k G(u) + r_k G(x_2))$$

با تغییر ماتریس همیلتونین مشتق آن نسبت به  $\vec{X}$  تغییر می کند که در کد لحاظ شده است.

$$\dot{\vec{P}} = -\frac{\partial \mathcal{H}}{\partial \vec{X}} = \begin{bmatrix} -x_1 + 0.4p_2 \\ -x_2 - p_1 + 0.4p_2x_2 - \frac{r_k}{2}G'(x_2) \end{bmatrix}$$

$$\begin{bmatrix} \dot{p}_1 \\ \dot{p}_2 \end{bmatrix} = \begin{bmatrix} -x_1 + 0.4p_2 \\ -p_1 + x_2(0.4p_2 - 1) - \frac{r_k}{2}G'(x_2) \end{bmatrix}$$

بر اساس روند بالا و موارد قبلی که در کد پیاده سازی شده بود می توان مشتق همیلتونین نسبت به تلاش کنترلی ( $u$ ) بدست آورد سپس جریمه تلاش کنترلی را نیز اضافه می کنیم.

$$\frac{\partial \mathcal{H}}{\partial u} = Ru + PB + r_k G'(u)$$

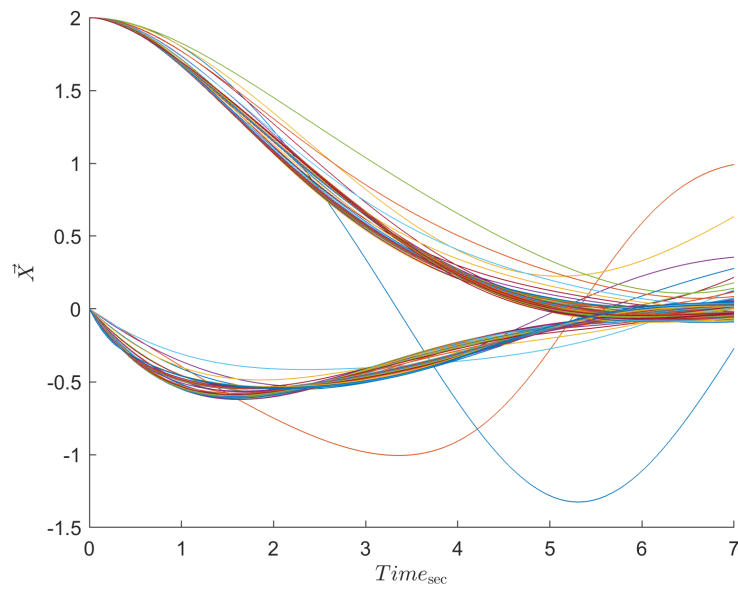
در بالا روش تحلیلی بدست آوردن گرادیان توضیح داده شد. تابع هزینه هم از انتگرال زیر بدست می آید.

$$J = 10(x_{1(t_f)}^2 + x_{2(t_f)}^2) + \frac{1}{2} \int_0^7 x_1^2 + x_2^2 + u^2 + r_k G(x_2) + r_k G(u)$$

کد پیوست شده توانایی حل با چهار روش را دارد آگه از منو می توان روش را انتخاب کرد. نتایج چهار روش آورده شده است.

- Steepest Descent + Quadratic Interpolation

Figure 1: Steepest Descent + Quadratic Interpolation



- Steepest Descent + Golden Section

Figure 2: Steepest Descent + Golden Section

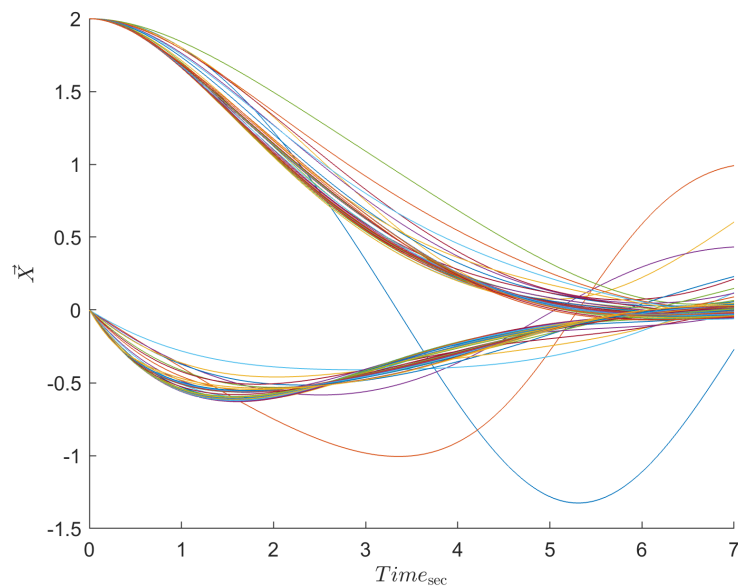


Figure 3: BFGS + Quadratic Interpolation

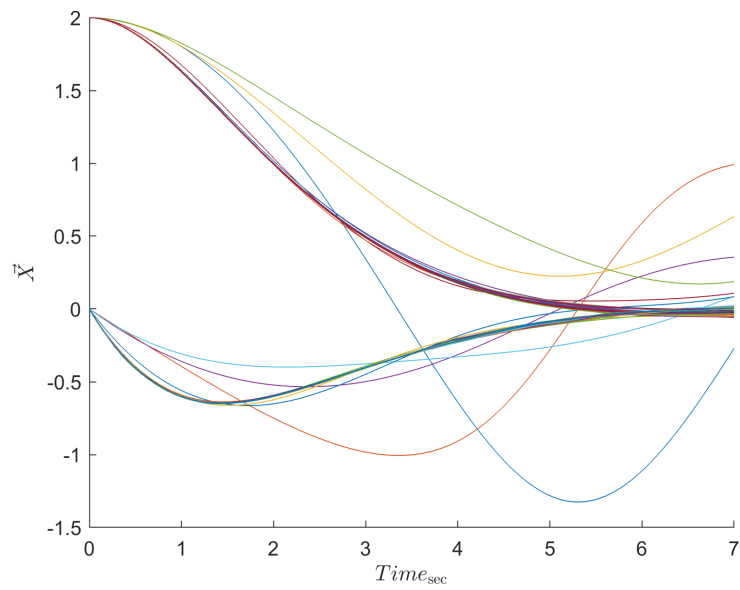
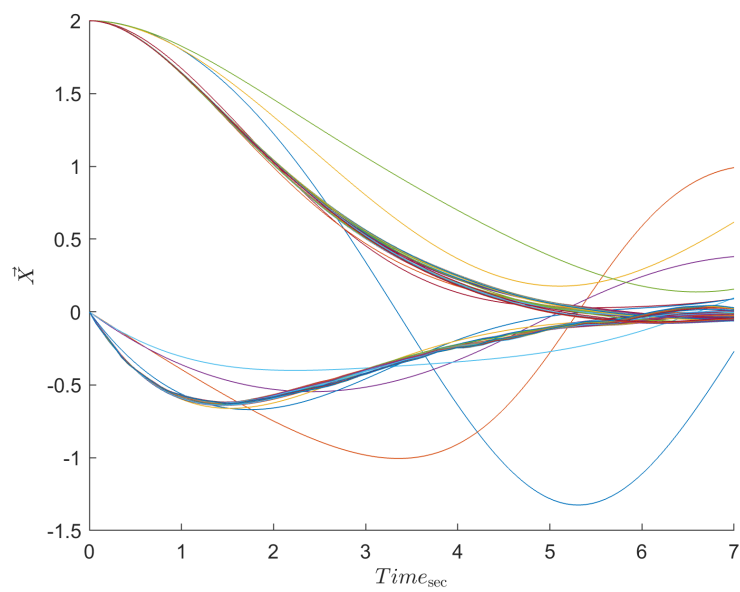


Figure 4: BFGS + Golden Section



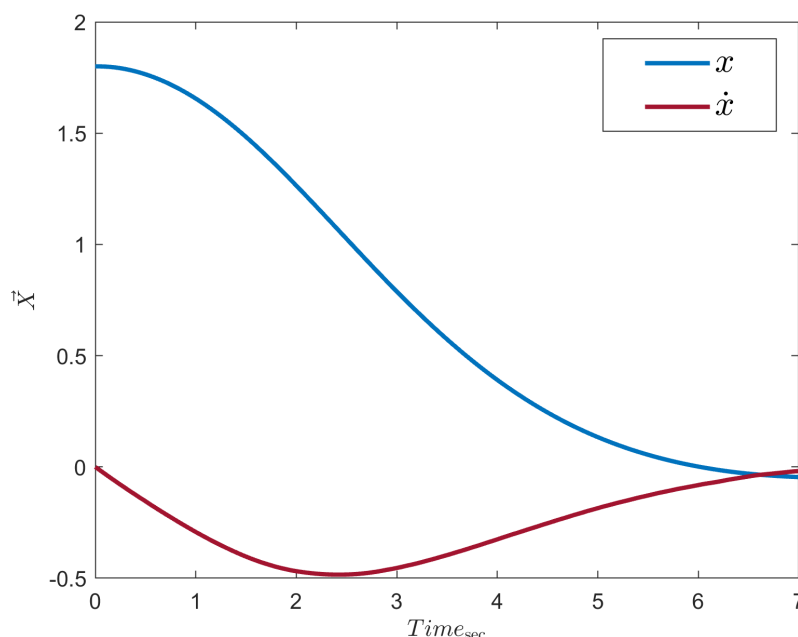
## سوال دوم

در این قسمت با توجه به نقطه اولیه ممکن است جواب درستی وجود نداشته باشد برای این کار اگر از قید رد شد تابع هزینه به شدت افزایش می‌یابد که با اینکار می‌توان به جواب حدودی همراه با قید رسید. مزیت اینکار نسبت به عدم در نظر گرفتن حالت خارج از آن یا قرار دادن Inf برای آن این است که تابع interpolation عملکرد بهتری خواهد داشت و اگر شرایط اولیه مطلوب نبود برنامه دچار گمراهی نمی‌شود و بازهم بهترین مسیر را انتخاب می‌کند ولی اگر شرایط اولیه درست باشد با توجه به تفاوت بسیار زیاد تابع هزینه در شرایط خارج از قید اصلا از قید خارج نمی‌شود. معدلات سیستم به فرم زیر نوشته شده‌است.

$$a = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -0.4x_1^2 - 0.2x_2^2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} x_2(k) \\ -0.4x_1(k)^2 - 0.2x_2(k)^2 + u(k) \end{bmatrix} \Delta t + \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}$$

Figure 5: Dynamic Programming result



گسسته سازی به صورت زیر انجام شد و کل برنامه در ۸ ثانیه اجرا شد. البته با توجه به گسسته سازی محدود interpolation با ماتریس‌های نسبتاً بزرگی انجام شده‌است و دقت کار را به شکل خوبی بالا می‌برد.