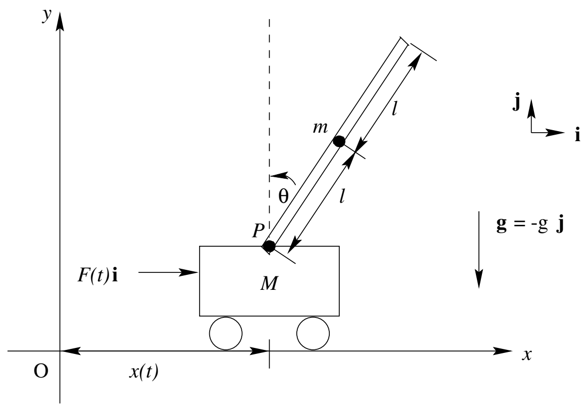
Linear control of inverted pendulum

Deep Ray, Ritesh Kumar, Praveen. C, Mythily Ramaswamy, J.-P. Raymond

IFCAM Summer School on Numerics and Control of PDE
22 July - 2 August 2013
IISc, Bangalore
http://praveen.cfdlab.net/teaching/control2013

1 Inverted pendulum



$$I = \text{Moment of inertia of pendulum about its cg} = \frac{1}{3}ml^2$$

Lagrangian

$$L = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m(\dot{x} + l\dot{\theta}\cos\theta)^2 + \frac{1}{2}(I + ml^2)\dot{\theta}^2 - mgl\cos\theta$$

Euler-Lagrange equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = F, \qquad \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = 0$$

gives

$$\begin{aligned} (M+m)\ddot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta + k\dot{x} &= F \\ ml\ddot{x}\cos\theta + (I + ml^2)\ddot{\theta} - mgl\sin\theta + c\dot{\theta} &= 0 \end{aligned}$$

This is a coupled system

$$\begin{bmatrix} M+m & ml\cos\theta \\ ml\cos\theta & I+ml^2 \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} ml\dot{\theta}^2\sin\theta - k\dot{x} + F \\ mgl\sin\theta - c\dot{\theta} \end{bmatrix}$$

Define determinant

$$D = (M+m)(I+ml^2) - m^2l^2\cos^2\theta$$

Solving for $\ddot{x}$, $\ddot{\theta}$

$$\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \frac{1}{D} \begin{bmatrix} ml\cos\theta(c\dot{\theta} - mgl\sin\theta) + (I+ml^2)(F - k\dot{x} + ml\dot{\theta}^2\sin\theta) \\ (M+m)(-c\dot{\theta} + mgl\sin\theta) - ml\cos\theta(F - k\dot{x} + ml\dot{\theta}^2\sin\theta) \end{bmatrix}$$

Define

$$z = (z_1, z_2, z_3, z_4)^\top = (x, \dot{x}, \theta, \dot{\theta})^\top$$

Then

$$\dot{z}_1 = z_2, \qquad \dot{z}_3 = z_4$$

we can write the first order system

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{bmatrix} = \begin{bmatrix} z_2 \\ \frac{1}{D}[ml\cos z_3(cz_4 - mgl\sin z_3) + (I+ml^2)(F - kz_2 + mlz_4^2\sin z_3)] \\ z_4 \\ \frac{1}{D}[(M+m)(-cz_4 + mgl\sin z_3) - ml\cos z_3(F - kz_2 + mlz_4^2\sin z_3)] \end{bmatrix}$$

In the experimental setup of Landry et al. [1], the force F on the cart is

$$F(t) = \alpha u(t) - \beta\dot{x}, \qquad \alpha > 0, \qquad \beta > 0$$

where u is the voltage in the motor driving the cart, and the second term represents the electrical resistance in the motor. Let us write the non-linear pendulum model as

$$\frac{\mathrm{d}z}{\mathrm{d}t} = N(z, u)$$

1.1 Excercises

The matlab programs are in the directory `pendulum`; go into this directory and start matlab. E.g. if you are working on the Unix/Linux command line

```
$ cd control2013/pendulum
$ matlab
```

If you have started matlab through some other way, then change the directory inside matlab

```
>> cd <Full path to>/control2013/pendulum
```

Or use the matlab gui to change the working directory.

The solution of the non-linear pendulum is implemented in `nlp.m` file and ths right hand side function without any control ($F = 0$) is implemented in `rhs_nlp.m` file. Study these two programs.

The pendulum has two equilibrium positions, one upright ($\theta = 0$) and another in the down position ($\theta = \pi$).

1. Using `nlp.m` solve the non-linear pendulum problem with initial condition close to upright position
$$z(0) = \left[0, 0, \tfrac{5\pi}{180}, 0\right]$$
and final time $T = 100$. Is the upright position stable ? What happens to the four variables ? What is the time asymptotic value of x, $\dot{x}$, θ, $\dot{\theta}$?

2. Solve the non-linear pendulum problem with initial condition close to downward position
$$z(0) = \left[0, 0, \tfrac{170\pi}{180}, 0\right]$$
and final time $T = 100$. Is the downward position stable ?

3. Now set the value of friction coefficients k and c to zero in `parameters.m` file and run `nlp.m` again. How does the solution behave qualitatively ? Is it converging to a stationary state ?

Note Set the value of k and c back to their initial non-zero values.

1.2 Linearised system

We linearize around $z = (0, 0, 0, 0)$

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -(k+\beta)v_2 & -\frac{m^2l^2gv_2}{I+ml^2} & \frac{mlcv_2}{I+ml^2} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{ml(k+\beta)v_2}{M+m} & mlgv_1 & -cv_1 \end{bmatrix}\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \alpha v_2 \\ 0 \\ -\frac{\alpha mlv_1}{M+m} \end{bmatrix} u$$

where
$$v_1 = \frac{M+m}{I(M+m)+mMl^2}, \qquad v_2 = \frac{I+ml^2}{I(M+m)+mMl^2}$$

then we have the linear model
$$\frac{\mathrm{d}z}{\mathrm{d}t} = Az + Bu$$

1.3 Excercises

1. Generate matrices for linear system

```
>> parameters
>> [A,B] = get_system_matrices()
```

2. Compute eigenvalues

```
>> e = eig(A)
>> plot(real(e), imag(e), 'o')
```

   Is there an unstable eigenvalue ? Is there a zero eigenvalue ?

3. The linear pendulum system is controllable if rank of controllability matrix is four. Check that (A, B) is controllable by computing rank of controllability matrix

$$P_c = \begin{bmatrix} B & AB & A^2B & A^3B \end{bmatrix}$$

```
>> Pc = [B, A*B, A^2*B, A^3*B];
>> rank(Pc)
```

4. Check the controllability using Hautus criterion. Compute all the eigenvectors and eigenvalues of $A^\top$. For each eigenvalue λ and eigenvector V of $A^\top$

$$A^\top V = \lambda V$$

   check if

$$B^\top V \neq 0$$

   If the above condition is true for each unstable eigenvalue, i.e., with $\text{real}(\lambda) \geq 0$, then the system is stabilizable. Is the linear pendulum stabilizable ?

5. Solve linear pendum model (see `lp.m` and `rhs_lp.m`) with initial condition

$$z(0) = \left[0, 0, \tfrac{5\pi}{180}, 0\right]$$

   upto a final time of $T = 5$. Compare this solution with solution of non-linear model.

2 Matlab function: `lqr`

The solution X of algebraic Riccati equation

$$A^\top X + XA - XBR^{-1}B^\top X + Q = 0, \qquad Q = C^\top C \tag{1}$$

can be computed in matlab using

```
>> [K,X,E] = lqr(A, B, Q, R);
```

Here K is the feedback gain and can be computed as

$$K = B^\top X$$

and E contains the eigenvalues of $A - BK$.

3 Minimal norm feedback control

The minimal norm control is given by

$$u(t) = -Kz(t) \qquad \text{with} \qquad K = B^\top X$$

where X is the maximal solution of Algebraic Bernoulli Equation (ABE)

$$A^\top X + XA - XBB^\top X = 0$$

For inverted pendulum, matrix A has a zero eigenvalue. We replace A with $A+\omega I$, $\omega > 0$, for determining the control.

$$(A + \omega I)^\top X + X(A + \omega I) - XBB^\top X = 0, \qquad K = B^\top X$$

Model with feedback

$$\frac{\mathrm{d}z}{\mathrm{d}t} = (A - BK)z, \qquad z(0) = z_0$$

3.1 Excercises

1. Compute the minimal norm feedback matrix K using `lqr` function

```
>> Q = zeros(4,4);
>> R = 1;
>> om = 0.01;
>> As = A + om*eye(4);
>> [K,X,E] = lqr(As, B, Q, R);
```

Compute feedback gain from solution `X` of Riccati equation and check that it is same as `K` given by `lqr` function

```
>> K1 = B'*X
>> K-K1          % This should give you zero matrix
```

E contains the eigenvalues of $As - BK$. Compare eigenvalues of `As` and `As-B*K`

```
>> e = eig(As);
>> plot(real(e), imag(e), 'o', real(E), imag(E), '*')
```

Observe that unstable eigenvalues of `As` have been reflected symmetrically about the imaginary axis.

2. Also check that $A - BK$ is stable by computing its eigenvalues.

```
>> e1 = eig(A);
>> e2 = eig(A-B*K);
>> plot(real(e1), imag(e1), 'o', real(e2), imag(e2), '*')
```

4 Feedback control using LQR approach

Output

$$y_m = Cz, \qquad \text{e.g.} \qquad C = I_4$$

Performance measure

$$\begin{aligned} J &= \frac{1}{2}\int_0^\infty y_m^\top Q_m y_m \mathrm{d}t + \frac{1}{2}\int_0^\infty u^\top R u \mathrm{d}t \\ &= \frac{1}{2}\int_0^\infty z^\top Q z \mathrm{d}t + \frac{1}{2}\int_0^\infty u^\top R u \mathrm{d}t, \qquad Q = C^\top Q_m C \end{aligned}$$

Find feedback law

$$u = -Kz$$

which minimizes J. The feedback or gain matrix K is given by

$$K = R^{-1} B^\top X$$

where X is solution of algebraic Riccati equation (ARE)

$$A^\top X + XA - XBR^{-1}B^\top X + Q = 0$$

4.1 Excercises

1. We may measure different output quantities, e.g.,

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}, \qquad C = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}, \qquad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \qquad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

These correspond to controlling (x), (θ), (x, θ) and $(x, \dot{x}, \theta, \dot{\theta})$ respectively. In each case, check if (A, C) is observable by computing rank of observability matrix

$$P_o = \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \end{bmatrix} \qquad \text{Is } \mathrm{rank}(P_o) = 4 \text{ or is } \mathrm{rank}(P_o) < 4\ ?$$

For which C is the rank condition satisfied ? Which of the four state variables is important ?

2. Compute feedback operator K using `lqr` function. We will control position and angle.

```
>> C = [1 0 0 0; 0 0 1 0];
>> Q = C'*C;
>> R = 1;
>> [K,X,E] = lqr(A, B, Q, R);
>> plot(real(E), imag(E), 'o')
```

E contains the eigenvalues of $A - BK$. Compare eigenvalues of A and $A - BK$

```
>> e = eig(A);
>> plot(real(e), imag(e), 'o', real(E), imag(E), '*')
```

Is $A - BK$ stable ?

3. Solve with LQR feedback control. This is implemented in program `pend_lqr.m` which solves both linear and non-linear model. Compare the linear and non-linear solutions. Try with $R = 0.01$ and $R = 0.1$ and initial condition

$$z(0) = \begin{bmatrix} 0 & 0 & \frac{50\pi}{180} & 0 \end{bmatrix}$$

Which is better, $R = 0.01$ or $R = 0.1$?

4. Run the program `pend_lqr.m` with different initial angles

$$z(0) = \begin{bmatrix} 0 & 0 & \frac{50\pi}{180} & 0 \end{bmatrix}, \qquad z(0) = \begin{bmatrix} 0 & 0 & \frac{58\pi}{180} & 0 \end{bmatrix}$$

and for $R = 0.01$ and $R = 0.1$. Which value of R gives faster stabilization ? What is the maximum magnitude of control in each case ?

5. If the initial angle θ_0 is too large, then it may not be possible to stabilize the non-linear pendulum. The linear pendulum is always stabilized. Find angle beyond which non-linear model cannot be stabilized. Do this for $R = 0.01$, $R = 0.1$ and $R = 0.5$. Use final time of $T = 3$ and initial conditions of the form

$$z(0) = \begin{bmatrix} 0 & 0 & \theta_0 & 0 \end{bmatrix}$$

Use the evolution of energy as an indicator for finding the threshold angle. If the energy starts to increase monotonically, then conclude that it is not possible to stabilize the system.

5 Linear system with noise and partial information

Consider the system with noise in the model and initial condition

$$\frac{\mathrm{d}z}{\mathrm{d}t} = Az + Bu + w, \qquad z(0) = z_0 + \eta$$

where w and η are error/noise terms. We may not have access to the full state but only some partial information which is also corrupted by noise.

$$y_o = Hz + v, \qquad \text{e.g.} \qquad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad \Longrightarrow \qquad y_o = \begin{bmatrix} x \\ \theta \end{bmatrix} + v$$

which corresponds to observing the position of cart x and the position of pendulum θ only.

Since we have partial information, we have to estimate to estimate the remaining state variables somehow.

5.1 A simple "estimator"

Suppose we can observe only $z_1 = x$ and $z_3 = \theta$ but this information is also contaminated by some error. Then we can try to estimate $\dot{x}$ and $\dot{\theta}$ by using finite difference in time.

$$\dot{x}(t_n) = \frac{x(t_n) - x(t_{n-1})}{\Delta t}, \qquad \dot{\theta}(t_n) = \frac{\theta(t_n) - \theta(t_{n-1})}{\Delta t}$$

Then we can try the following feedback control scheme using backward Euler time discretization

$$\begin{aligned} y^n &= Hz^n + v^n \\ u^n &= -K \begin{bmatrix} y_1^n \\ \frac{1}{\Delta t}(y_1^n - y_1^{n-1}) \\ y_2^n \\ \frac{1}{\Delta t}(y_2^n - y_2^{n-1}) \end{bmatrix} \\ \frac{z^{n+1} - z^n}{\Delta t} &= Az^{n+1} + Bu^n + w^n \end{aligned}$$

This is implemented in `lp_noest.m`.

5.2 Excercise

Run the program `lp_noest.m` and observe that the state and control are very noisy and hence this approach is not of practical utility.

5.3 Estimation problem

Linear estimator

$$\frac{\mathrm{d}z_e}{\mathrm{d}t} = Az_e + Bu + L(y_o - Hz_e)$$

We determine the filtering gain L by minimizing

$$J = \frac{1}{2}\int_0^\infty (y_o - Hz_e)^\top R_v^{-1}(y_o - Hz_e)\mathrm{d}t + \frac{1}{2}\int_0^\infty w^\top R_w^{-1} w \mathrm{d}t$$

The best L is given by

$$L = X_e H^\top R_v^{-1} \tag{2}$$

where X_e is solution of algebraic Riccati equation

$$AX_e + X_e A^\top - X_e H^\top R_v^{-1} H X_e + R_w = 0$$

This equation is similar to equation (1) if we replace

$$\begin{array}{lll} A & \text{with} & A^\top \\ B & \text{with} & H^\top \\ Q & \text{with} & R_w \\ R & \text{with} & R_v \end{array}$$

5.4 Excercises

1. Compute the filtering matrix L

```
>> Rw = (5e-2)^2 * diag([1, 1, 1, 1]);
>> Rv = (5e-2)^2 * diag([1, 1]);
>> [L, Xe] = lqr(A', H', Rw, Rv);
>> L = real(L');
```

2. Compute eigenvalues of $A - LH$ and check that they are all stable.

```
>> eig(A-L*H)
```

3. Compute L from equation (2) and check that it is same as L obtained in step 1 above.

```
>> L1 = Xe * H' / Rv;
>> L - L1
```

5.5 Linear system with linear estimator

The feedback is based on estimated solution $u = -Kz_e$

$$\begin{aligned}\frac{\mathrm{d}z}{\mathrm{d}t} &= Az - BKz_e + w \\ \frac{\mathrm{d}z_e}{\mathrm{d}t} &= LHz + (A - BK - LH)z_e + Lv\end{aligned}$$

or in matrix form

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} z \\ z_e \end{bmatrix} = \begin{bmatrix} A & -BK \\ LH & A - BK - LH \end{bmatrix}\begin{bmatrix} z \\ z_e \end{bmatrix} + \begin{bmatrix} I & 0 \\ 0 & L \end{bmatrix}\begin{bmatrix} w \\ v \end{bmatrix}$$

The initial condition is given by

$$z(0) = z_0 + \eta, \qquad z_e(0) = z_0$$

This is implemented in program `lp_est.m`

5.6 Excercises

1. Run program `lp_est.m`. Observe that initially z and z_e differ but after some time, z_e approches z so that the estimation becomes more accurate. You may have to zoom the plot near the initial time to see the difference.

2. Vary the magnitude of the errors (GIVE SOME CHOICES) in the covariance matrices `Rw, Ru`. How does each type of control perform?

5.7 Non-linear system with linear estimator

The feedback is based on estimated solution using the linear estimator which is used to add the control into the non-linear model.

$$\begin{aligned}\frac{\mathrm{d}z}{\mathrm{d}t} &= N(z, u) \\ \frac{\mathrm{d}z_e}{\mathrm{d}t} &= Az_e + Bu + L(y_o - Hz_e) \\ y_o &= Hz \\ u &= -Kz_e\end{aligned}$$

with initial condition

$$z(0) = z_0 + \eta, \qquad z_e(0) = z_0$$

Since we do not know the precise initial condition, the estimator contains an error in the initial condition. This is implemented in program `nlp_est.m` and `rhs_nlpest.m`

5.8 Excercises

1. Run program `nlp_est.m`. Zoom the plot near initial time and compare the non-linear solution with the linear estimator. Note that z_e approaches z for larger times.

2. Increase the initial angle and check if the feedback can be stabilize the non-linear system. Have the threshold angles evaluated for $R = 0.01$, $R = 0.1$ and $R = 0.5$ changed? If yes, then what are the new threshold angles?

3. How does the control vary as R is changed? Is the behaviour the same as that observed in the earlier cases?

4. How does the value of R influence the stabilization of the state variables?

5.9 Excercises

Here we will assume that we do not know the value of the friction coefficient k and c. So the linear estimator will be constructed taking $k = 0$ and $c = 0$ whereas the non-linear model will include the friction. We want to see if the estimator is still able to give a good estimate of the real state and if the system is stabilized by the feedback.

1. In `parameters.m` set k and c to be zero.

2. Copy `nlp_est.m` as `nlp_est2.m`

3. Modify the initial part like this

   ```
   parameters;
   [A,B] = get_system_matrices();
   k = 0.05;
   c = 0.005;
   ```

4. Run the program

   ```
   >> nlp_est2
   ```

   Does the estimate and control still succeed ?

6 Inverted pendulum: Parameters

M	Mass of cart	2.4
m	Mass of pendulum	0.23
l	Length of pendulum	0.36
k	Friction in cart	0.05
c	Friction in pendulum	0.005
α	Coefficient of voltage	1.7189
β	Friction in motor	7.682
a	Acceleration due to gravity	9.81

7 List of Programs

1. `parameters.m`: Set parameters for pendulum
2. `get_system_matrices.m`: Computes matrices for linear model
3. `hautus.m`: Checks the stabilizability of the system using Hautus criterion
4. `rhs_nlpc.m`: Computes right hand side of non-linear model, with feedback
5. `rhs_lp.m`: Computes right hand side of linear model, without feedback
6. `rhs_nlp.m`: Computes right hand side of non-linear model, without feedback
7. `rhs_lpc.m`: Computes right hand side of linear model, with feedback
8. `rhs_nlpc.m`: Computes right hand side of non-linear model, with feedback
9. `rhs_nlpest.m`: Computes right hand side of non-linear model coupled with the linear estimator and feedback
10. `lp.m`: Solves the linear model without feedback
11. `nlp.m`: Solves the linear model without feedback
12. `pend_lqr.m`: Solves the linear and non-linear models with feedback and full state information
13. `lp_est.m`: Solves the linear model with noise and partial information, coupled estimator and feedback stabilization
14. `lp_noest.m`: Solves the linear model with noise and partial information, feedback without estimation

15. `nlp_est.m`: Solves the non-linear model with noise and partial information, coupled estimator and feedback stabilization

References

[1] M. Landry, S. A. Campbell, L. Morris and C. O. Aguilar, "Dynamics of an inverted pendulum with delayed feedback control", SIAM J. Applied Dynamical Systems, Vol. 4, No. 2, pp. 333-351, 2005.