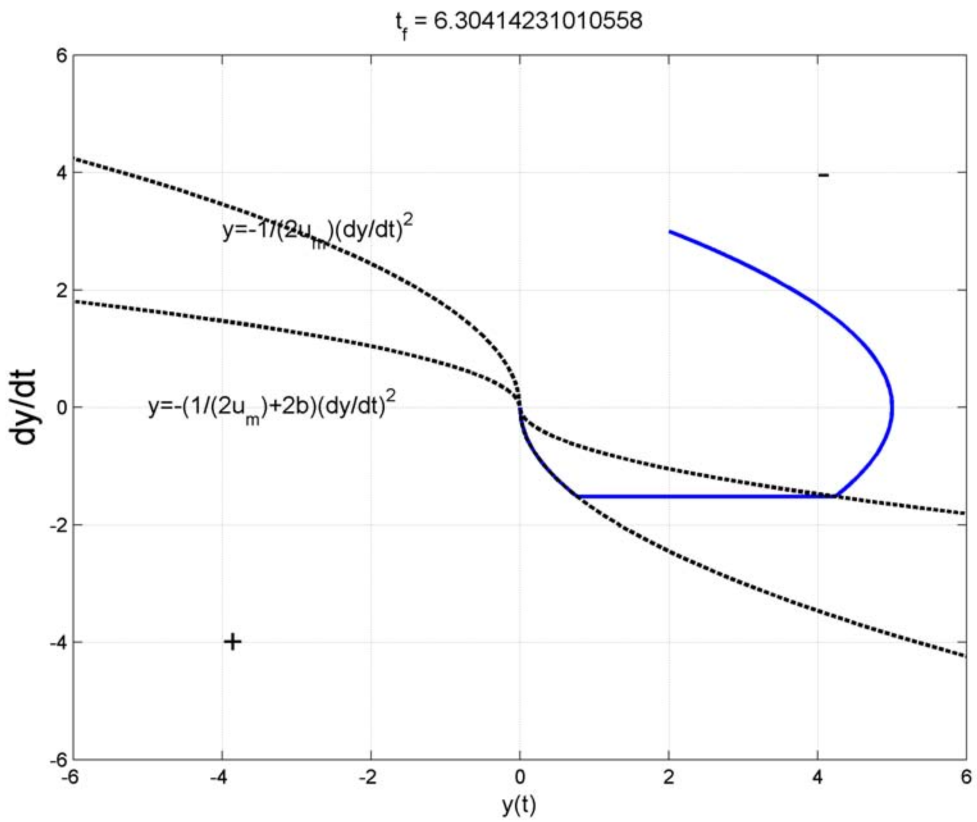
16.323 Principles of Optimal Control
Spring 2008

16.323 Lecture 9

Constrained Optimal Control

Bryson and Ho – Section 3.x and Kirk – Section 5.3



Constrained Optimal Control

- First consider cases with constrained control inputs so that $\mathbf{u}(t) \in \mathcal{U}$ where $\mathcal{U}$ is some bounded set.
  - Example: inequality constraints of the form $\mathbf{C}(\mathbf{x}, \mathbf{u}, t) \leq 0$
  - Much of what we had on 6–3 remains the same, but algebraic condition that $H_{\mathbf{u}} = 0$ must be replaced
  - Note that $\mathbf{C}(\mathbf{x}, t) \leq 0$ is a much harder case

- Augment constraint to cost (along with differential equation constraints)

$$J_a = h(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} \left[H - \mathbf{p}^T \dot{\mathbf{x}} + \boldsymbol{\nu}^T \mathbf{C} \right] dt$$

- Find the variation (assume t_0 and $x(t_0)$ fixed):

$$\begin{aligned} \delta J_a \;=\; & h_{\mathbf{x}} \delta \mathbf{x}_f + h_{t_f} \delta t_f + \int_{t_0}^{t_f} \Big[H_{\mathbf{x}} \delta \mathbf{x} + H_{\mathbf{u}} \delta \mathbf{u} + (H_{\mathbf{p}} - \dot{\mathbf{x}}^T) \delta \mathbf{p}(t) \\ & - \mathbf{p}^T(t) \delta \dot{\mathbf{x}} + \mathbf{C}^T \delta \boldsymbol{\nu} + \boldsymbol{\nu}^T \left\{ \mathbf{C}_{\mathbf{x}} \delta \mathbf{x} + \mathbf{C}_{\mathbf{u}} \delta \mathbf{u} \right\} \Big] dt \\ & + \left[H - \mathbf{p}^T \dot{\mathbf{x}} + \boldsymbol{\nu}^T \mathbf{C} \right] (t_f) \delta t_f \end{aligned}$$

- Now IBP

$$-\int_{t_0}^{t_f} \mathbf{p}^T(t) \delta \dot{\mathbf{x}} dt = -\mathbf{p}^T(t_f) \left(\delta \mathbf{x}_f - \dot{\mathbf{x}}(t_f) \delta t_f \right) + \int_{t_0}^{t_f} \dot{\mathbf{p}}^T(t) \delta \mathbf{x} dt$$

then combine and drop terminal conditions for simplicity:

$$\begin{aligned} \delta J_a \;=\; & \int_{t_0}^{t_f} \Big\{ \left[H_{\mathbf{x}} + \dot{\mathbf{p}}^T + \boldsymbol{\nu}^T \mathbf{C}_{\mathbf{x}} \right] \delta \mathbf{x} + \left[H_{\mathbf{u}} + \boldsymbol{\nu}^T \mathbf{C}_{\mathbf{u}} \right] \delta \mathbf{u} \\ & + (H_{\mathbf{p}} - \dot{\mathbf{x}}^T) \delta \mathbf{p}(t) + \mathbf{C}^T \delta \boldsymbol{\nu} \Big\} dt \end{aligned}$$

- Clean up by defining augmented **Hamiltonian**

$$H_a(\mathbf{x}, \mathbf{u}, \mathbf{p}, t) = g + \mathbf{p}^T(t)\mathbf{a} + \boldsymbol{\nu}^T(t)\mathbf{C}$$

where (see 2–12)

$$\nu_i(t) \begin{cases} \geq 0 & \text{if} \quad C_i = 0 \quad \text{active} \\ = 0 & \text{if} \quad C_i < 0 \quad \text{inactive} \end{cases}$$

– So that $\nu_i C_i = 0 \; \forall \; i$.

- So necessary conditions for $\delta J_a = 0$ are that for $t \in [t_0, t_f]$

$$\boxed{\begin{aligned} \dot{\mathbf{x}} &= \mathbf{a}(\mathbf{x}, \mathbf{u}, t) \\ \dot{\mathbf{p}} &= -(H_a)_{\mathbf{x}}^T \\ (H_a)_{\mathbf{u}} &= 0 \end{aligned}}$$

– With appropriate boundary conditions and $\nu_i C_i(\mathbf{x}, \mathbf{u}, t) = 0$

- Complexity here is that typically will have sub-arcs to the solution where the inequality constraints are active (so $C_i(\mathbf{x}, \mathbf{u}, t) = 0$) and then not (so $\nu_i = 0$).
  – Transitions between the sub-arcs must be treated as corners that are at unspecified times - need to impose the equivalent of the Erdmann-Weirstrass corner conditions for the control problem, as in Lecture 8.

Constrained Example

- Design the control inputs that minimize the cost functional

$$\min_u J = -x(4) + \int_0^4 u^2(t)dt$$

with $\dot{x} = x + u$, $x(0) = 0$, and $u(t) \leq 5$.

- Form augmented Hamiltonian:

$$H = u^2 + p(x+u) + \nu(u-5)$$

- Note that, independent of whether the constraint is active or not, we have that

$$\dot{p} = -H_x = -p \quad \Rrightarrow \quad p(t) = ce^{-t}$$

and from transversality BC, know that $p(4) = \partial h/\partial x = -1$, so have that $c = -e^4$ and thus $p(t) = -e^{4-t}$

- Now let us assume that the control constraint is initially active for some period of time, then $\nu \geq 0$, $u = 5$, and

$$H_u = 2u + p + \nu = 0$$

so we have that

$$\nu = -10 - p = -10 + e^{4-t}$$

  – Question: for what values of t will $\nu \geq 0$?

$$\begin{aligned} \nu &= -10 + e^{4-t} \geq 0 \\ &\rightarrow e^{4-t} \geq 10 \\ &\rightarrow 4 - t \geq \ln(10) \\ &\rightarrow 4 - \ln(10) \geq t \end{aligned}$$

  – So provided $t \leq t_c = 4 - \ln(10)$ then $\nu \geq 0$ and the assumptions are consistent.

- Now consider the inactive constraint case:

$$H_u = 2u + p = 0 \Rrightarrow u(t) = -\frac{1}{2}p(t)$$

- The control inputs then are

$$u(t) = \begin{cases} 5 & t \leq t_c \\ \frac{1}{2}e^{4-t} & t \geq t_c \end{cases}$$

which is continuous at t_c.

- To finish the solution, find the state in the two arcs $x(t)$ and enforce continuity at t_c, which gives that:

$$x(t) = \begin{cases} 5e^t - 5 & t \leq t_c \\ -\frac{1}{4}e^{4-t} + (5 - 25e^{-4})e^t) & t \geq t_c \end{cases}$$

- Note that since the corner condition was not specified by a state constraint, continuity of λ and H at the corner is required – but we did not need to use that in this solution, it will occur naturally.

Pontryagin's Minimum Principle

- For an alternate perspective, consider general control problem statement on 6–1 (free end time and state). Then on 6–2,

$$\delta J_a = \left(h_{\mathbf{x}} - \mathbf{p}^T(t_f)\right)\delta \mathbf{x}_f + \left[h_{t_f} + H\right](t_f)\delta t_f \tag{9.13}$$
$$+ \int_{t_0}^{t_f} \left[\left(H_{\mathbf{x}} + \dot{\mathbf{p}}^T\right)\delta\mathbf{x} + H_{\mathbf{u}}\delta\mathbf{u} + (H_{\mathbf{p}} - \dot{\mathbf{x}}^T)\delta\mathbf{p}(t)\right] dt$$

now assume we have a trajectory that satisfies all other differential equation and terminal constraints, then all remains is

$$\Rightarrow \delta J_a = \int_{t_0}^{t_f} \left[H_{\mathbf{u}}(t)\delta\mathbf{u}(t)\right] dt \tag{9.14}$$

- For the control to be minimizing, need $\delta J_a \geq 0$ for all admissible variations in $\mathbf{u}$ (i.e., $\delta\mathbf{u}$ for which $C_{\mathbf{u}}\delta\mathbf{u} \leq 0$)
  - Equivalently, need $\delta H = H_{\mathbf{u}}(t)\delta\mathbf{u}(t) \geq 0$ for all time and for all admissible $\delta\mathbf{u}$
  - Gives condition that $H_{\mathbf{u}} = 0$ if control constraints not active
  - However, at the constraint boundary, could have $H_{\mathbf{u}} \neq 0$ and whether we need $H_{\mathbf{u}} > 0$ or $H_{\mathbf{u}} < 0$ depends on the direction (sign) of the admissible $\delta\mathbf{u}$.

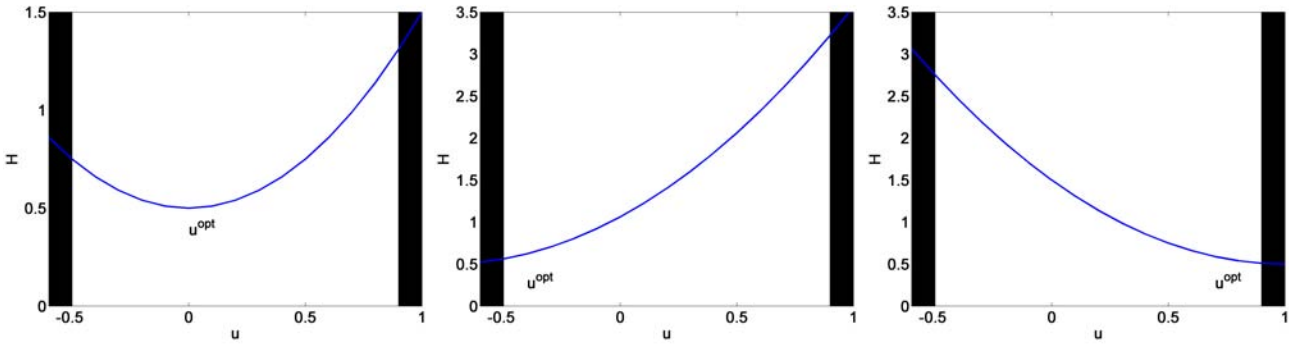


Figure 9.1: Examples of options for $\delta H = H_{\mathbf{u}}(t)\delta\mathbf{u}(t)$. Left: unconstrained min, so need $H_{\mathbf{u}} = 0$. Middle: constraint on left, so at min value, must have $\delta u \geq 0$ $\Rightarrow$ need $H_{\mathbf{u}} \geq 0$ so that $\delta H \geq 0$. Right: constraint on right, so at min value, must have $\delta u \leq 0 \Rightarrow$ need $H_{\mathbf{u}} \leq 0$ so that $\delta H \geq 0$.

- The requirement that $\delta H \geq 0$ says that δH must be non-improving to the cost (recall trying to minimize the cost) over the set of possible $\delta \mathbf{u}$.
  - Can actually state a stronger condition: H must be minimized over the set of all possible $\mathbf{u}$

- Thus for control constrained problems, third necessary condition

$$H_{\mathbf{u}} = 0$$

must be replaced with a more general necessary condition

$$\mathbf{u}^{\star}(t) = \arg\left\{\min_{\mathbf{u}(t)\in\mathcal{U}} H(\mathbf{x},\mathbf{u},\mathbf{p},t)\right\}$$

  - So must look at H and explicitly find the minimizing control inputs given the constraints - not as simple as just solving $H_{\mathbf{u}} = 0$
  - Known as **Pontryagin's Minimum Principle**
  - Handles "edges" as well, where the admissible values of $\delta \mathbf{u}$ are "inwards"

- PMP is very general and applies to all constrained control problems – will now apply it to a special case in which the performance and the constraints are linear in the control variables.

PMP Example: Control Constraints

- Consider simple system $y = G(s)u$, $G(s) = 1/s^2$ with $|u(t)| \leq u_m$
  - Motion of a rigid body with limited control inputs – can be used to model many different things
- Want to solve the **minimum time-fuel problem**

$$\min J = \int_0^{t_f} (1 + b|u(t)|)dt$$

  - The goal is to drive the state to the origin with minimum cost.
  - Typical of many spacecraft problems – $\int |u(t)|dt$ sums up the fuel used, as opposed to $\int u^2(t)dt$ that sums up the power used.

- Define $x_1 = y$, $x_2 = \dot{y}$ $\Rightarrow$ dynamics are $\dot{x}_1 = x_2$, $\dot{x}_2 = u$

- First consider the response if we apply ± 1 as the input. Note:
  - If $u = 1$, $x_2(t) = t + c_1$ and

$$x_1(t) = 0.5t^2 + c_1 t + c_2 = 0.5(t + c_1)^2 + c_3 = 0.5x_2(t)^2 + c_3$$

  - If $u = -1$, $x_2(t) = -t + c_4$ and

$$x_1(t) = -0.5t^2 + c_4 t + c_5 = -0.5(t + c_4)^2 + c_6 = -0.5x_2(t)^2 + c_6$$

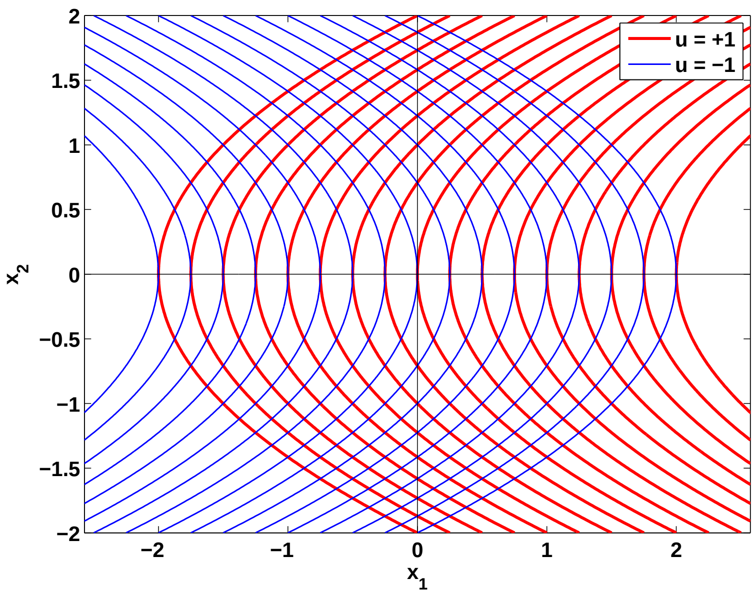


Figure 9.2: Possible response curves – what is the direction of motion?

- Hamiltonian for the system is:

$$\begin{aligned} H &= 1 + b|u| + \begin{bmatrix} p_1 & p_2 \end{bmatrix} \left\{ \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \right\} \\ &= 1 + b|u| + p_1 x_2 + p_2 u \end{aligned}$$

- First find the equations for the co-state:

$$\dot{\mathbf{p}} = -H_{\mathbf{x}}^T \qquad \Rightarrow \qquad \begin{cases} \dot{p}_1 = -H_{x_1} = 0 & \rightarrow \quad p_1 = c_1 \\ \dot{p}_2 = -H_{x_2} = -p_1 & \rightarrow \quad p_2 = -c_1 t + c_2 \end{cases}$$

  – So p_2 is linear in time

- To find optimal control, look at the parts of H that depend on u:

$$\tilde{H} = b|u| + p_2 u$$

  – **Recall PMP:** given constraints, goal is to find u that minimizes H (or $\tilde{H}$)
  – Sum of two functions $|u|$ and u - sign of which depends on sign and relative size of p_2 compared to $b > 0$

- Three cases to consider (plots use $u_m = 1.5$):

1. $p_2 > b > 0 \longrightarrow$ choose $u^\star(t) = -u_m$

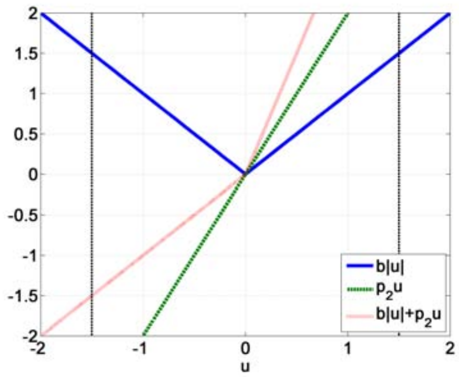


Figure 9.3: $b = 1, p_2 = 2$, so $p_2 > b > 0$
fopt1

2. $p_2 < -b \longrightarrow$ choose $u^\star(t) = u_m$

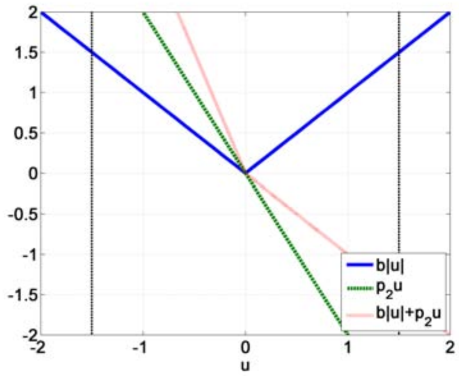


Figure 9.4: $b = 1, p_2 = -2$, so $p_2 < -b$

3. $-b < p_2 < b \longrightarrow$ choose $u^\star(t) = 0$

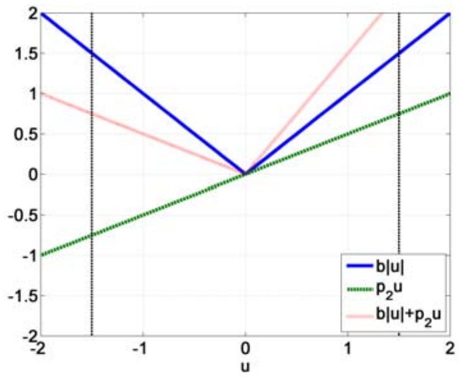


Figure 9.5: $b = 1, p_2 = 1$, so $-b < p_2 < b$

- The resulting control law is:

$$u(t) = \begin{cases} -u_m & b < \ p_2(t) \\ 0 & -b < \ p_2(t) < \quad b \\ u_m & \qquad p_2(t) < \ -b \end{cases}$$

- So the control depends on $p_2(t)$ - but since it is a linear function of time, it is only possible to get at most 2 switches
  - Also, since $\dot{x}_2(t) = u$, and since we must stop at t_f, then must have that $u = \pm u_m$ at t_f

- To complete the solution, impose the boundary conditions (transversality condition), with $x_2(t_f) = 0$

$$H(t_f) + h_t(t_f) = 0 \quad \rightarrow \quad 1 + b|u(t_f)| + p_2(t_f)u(t_f) = 0$$

  - If $u = u_m$, then $1 + bu_m + p_2(t_f)u_m = 0$ implies that

$$p_2(t_f) = -(b + \frac{1}{u_m}) < -b$$

    which is consistent with the selection rules.
  - And if $u = -u_m$, then $1 + bu_m - p_2(t_f)u_m = 0$ implies that

$$p_2(t_f) = (b + \frac{1}{u_m}) > b$$

    which is also consistent.
  - So the terminal condition does not help us determine if $u = \pm u_m$, since it could be either

- So first look at the case where $u(t_f) = u_m$. Know that

$$p_2(t) = c_2 - c_1 t$$

and $p_2(t_f) = -(b + \frac{1}{u_m}) < -b$.
  - Assume that $c_1 > 0$ so that we get some switching.

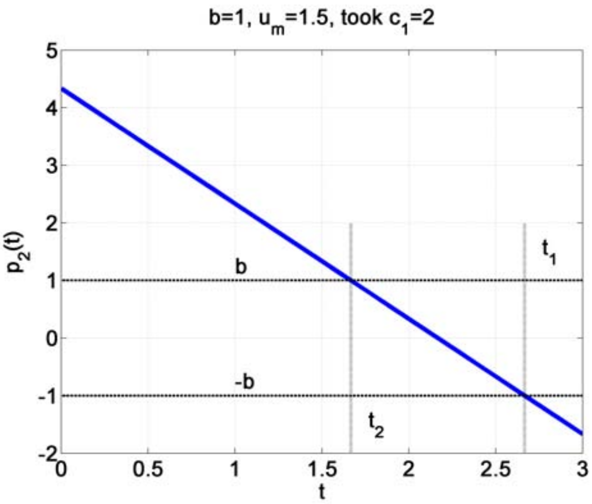


Figure 9.6: Possible switching case, but both t_f and c_1 are unknown at this point.

  - Then set $p_2(t_1) = -b$ to get that $t_1 = t_f - 1/(u_m c_1)$
  - And $p_2(t_2) = b$ gives $t_2 = t_f - (2b + 1/u_m)/c_1$

- Now look at the state response:
  - Starting at the end: $\ddot{y} = u_m$, gives $y(t) = u_m/2t^2 + c_3 t + c_4$, where $\dot{y} = y = 0$ at t_f gives us that $c_3 = -u_m t_f$ and $c_4 = u_m/2t_f^2$, so

$$y(t) = \frac{u_m}{2}t^2 - u_m t_f t + \frac{u_m}{2}t_f^2 = \frac{u_m}{2}(t - t_f)^2$$

  - But since $\dot{y}(t) = u_m t + c_3 = u_m(t - t_f)$, then $\boxed{y(t) = \dfrac{\dot{y}(t)^2}{2u_m}}$

  - State response associated with $u = u_m$ is in lower right quadrant of the $y/\dot{y}$ phase plot

- Between times t_2–t_1, control input is zero $\Rightarrow$ **coasting phase**.
  - Terminal condition for coast same as the start of the next one:

$$y(t_1) = \frac{u_m}{2}(t_1 - t_f)^2 = \frac{1}{2u_m c_1^2}$$

and $\dot{y}(t_1) = -1/c_1$

  - On a coasting arc, $\dot{y}$ is a constant (so $\dot{y}(t_2) = -1/c_1$), and thus

$$y(t_2) - \frac{(t_1 - t_2)}{c_1} = \frac{1}{2u_m c_1^2}$$

which gives that

$$\begin{aligned} y(t_2) &= \frac{1}{2u_m c_1^2} + \frac{1}{c_1}\left(t_f - \frac{1}{u_m c_1} - (t_f - (\frac{2b}{c_1} + \frac{1}{u_m c_1}))\right) \\ &= (2b + \frac{1}{2u_m})\frac{1}{c_1^2} = (2b + \frac{1}{2u_m})\dot{y}(t_2)^2 \end{aligned}$$

- So the first transition occurs along the curve $$y(t) = (2b + \frac{1}{2u_m})\dot{y}(t)^2$$

- For the first arc, things get a bit more complicated.
Clearly $u(t) = -u_m$, with IC $y_0,\ \dot{y}_0$ so

$$\begin{aligned} \dot{y}(t) &= -u_m t + c_5 = -u_m t + \dot{y}_0 \\ y(t) &= -\frac{u_m}{2}t^2 + c_5 t + c_6 = -\frac{u_m}{2}t^2 + \dot{y}_0 t + y_0 \end{aligned}$$

  - Now project forward to t_2

$$\begin{aligned} \dot{y}(t_2) &= -u_m t_2 + \dot{y}_0 = \dot{y}(t_1) = -\frac{1}{c_1} \quad \rightarrow \quad c_1 = \frac{2(b + 1/u_m)}{t_f - \dot{y}_0/u_m} \\ y(t_2) &= -\frac{u_m}{2}t_2^2 + \dot{y}_0 t_2 + y_0 \end{aligned}$$

and use these expressions in the quadratic for the switching curve to solve for c_1, t_1, t_2

- The solutions have a very distinctive **Bang–Off–Bang** pattern
  - Two parabolic curves define switching from $+u_m$ to 0 to $-u_m$

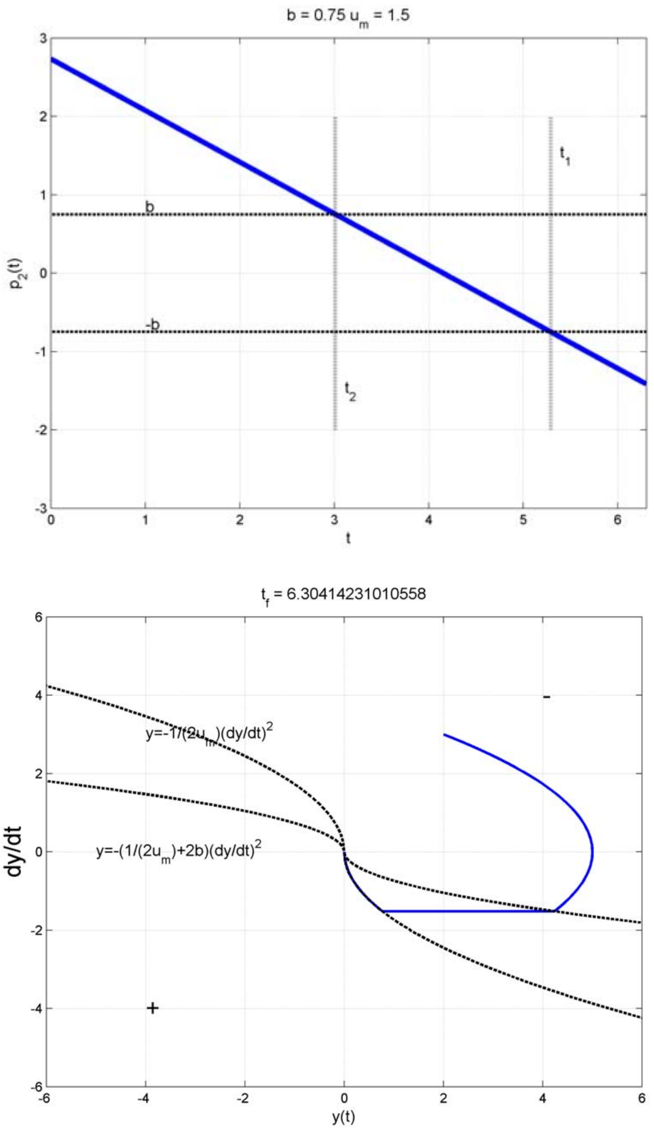


Figure 9.7: $y_0 = 2$ $\dot{y}_0 = 3$ $b = 0.75$ $u_m = 1.5$

- Switching control was derived using a detailed evaluation of the state and costate
  - But final result is a switching law that can be written wholly in terms of the system states.

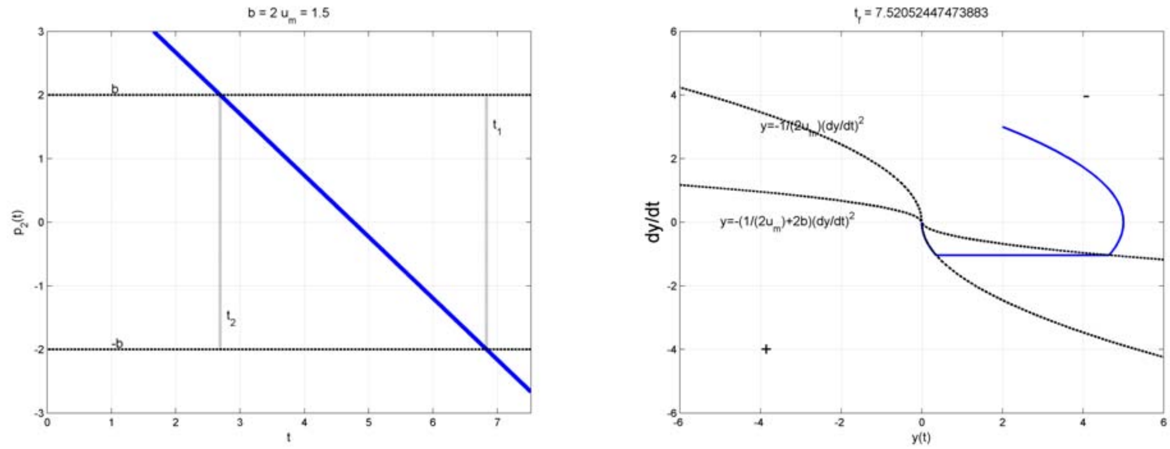


Figure 9.8: $y_0 = 2\ \dot{y}_0 = 3\ b = 2\ u_m = 1.5$

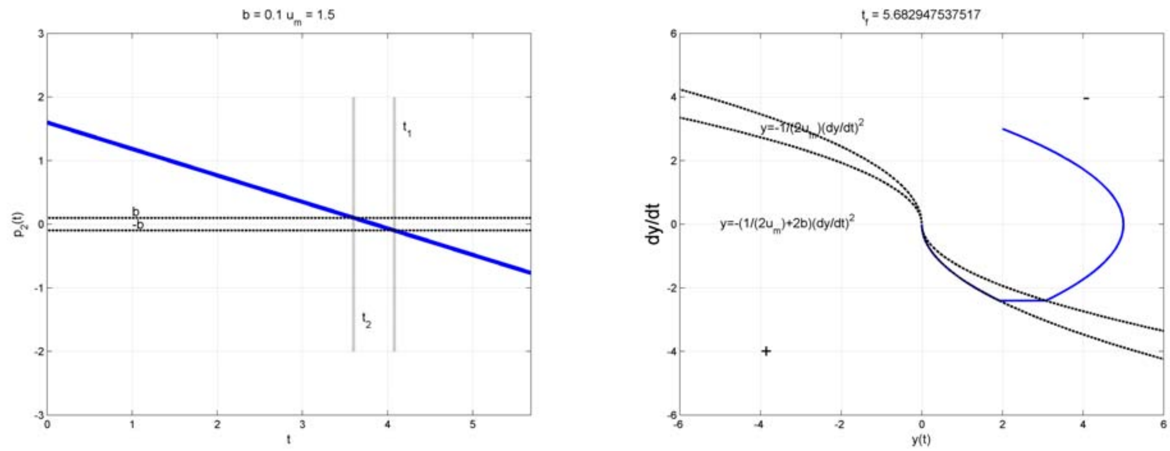


Figure 9.9: $y_0 = 2\ \dot{y}_0 = 3\ b = 0.1\ u_m = 1.5$

- Clearly get a special result as $b \longrightarrow 0$, which is the solution to the **minimum time problem**
  - Control inputs are now just **Bang–Bang**
  - One parabolic curve defines switching from $+u_m$ to $-u_m$

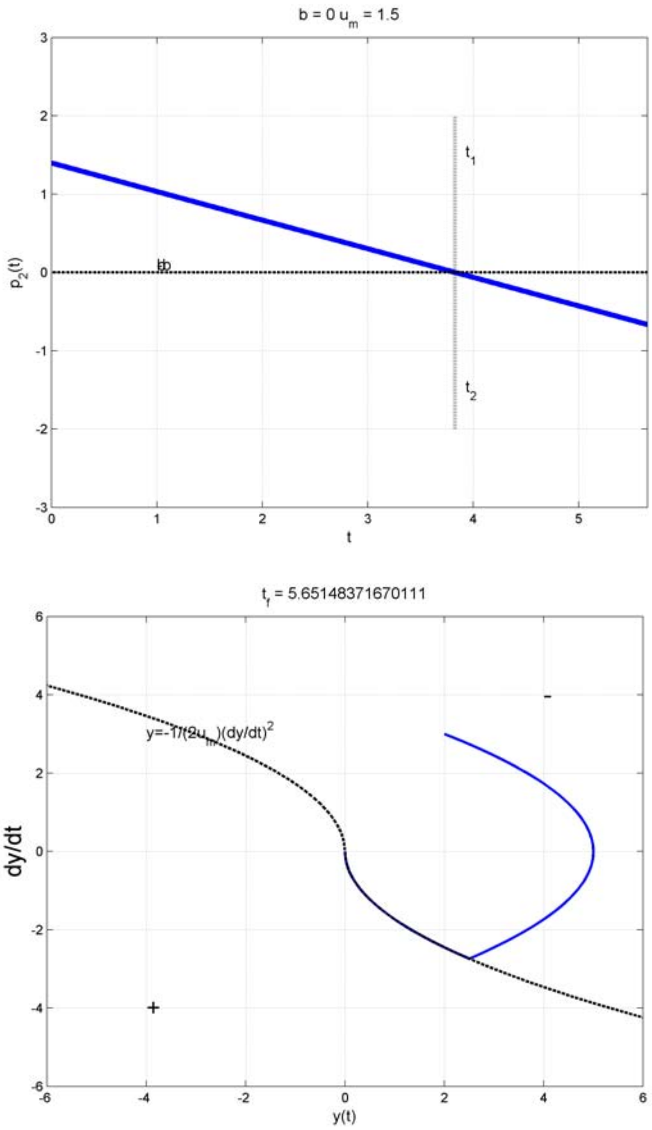


Figure 9.10: Min time: $y_0 = 2$ $\dot{y}_0 = 3$ $b = 0$ $u_m = 1.5$

- Can show that the switching and final times are given by

$$t_1 = \dot{y}(0) + \sqrt{y(0) + 0.5\dot{y}^2(0)} \qquad t_f = \dot{y}(0) + 2\sqrt{y(0) + 0.5\dot{y}^2(0)}$$

- **Trade-off:** coasting is fuel efficient, but it takes a long time.

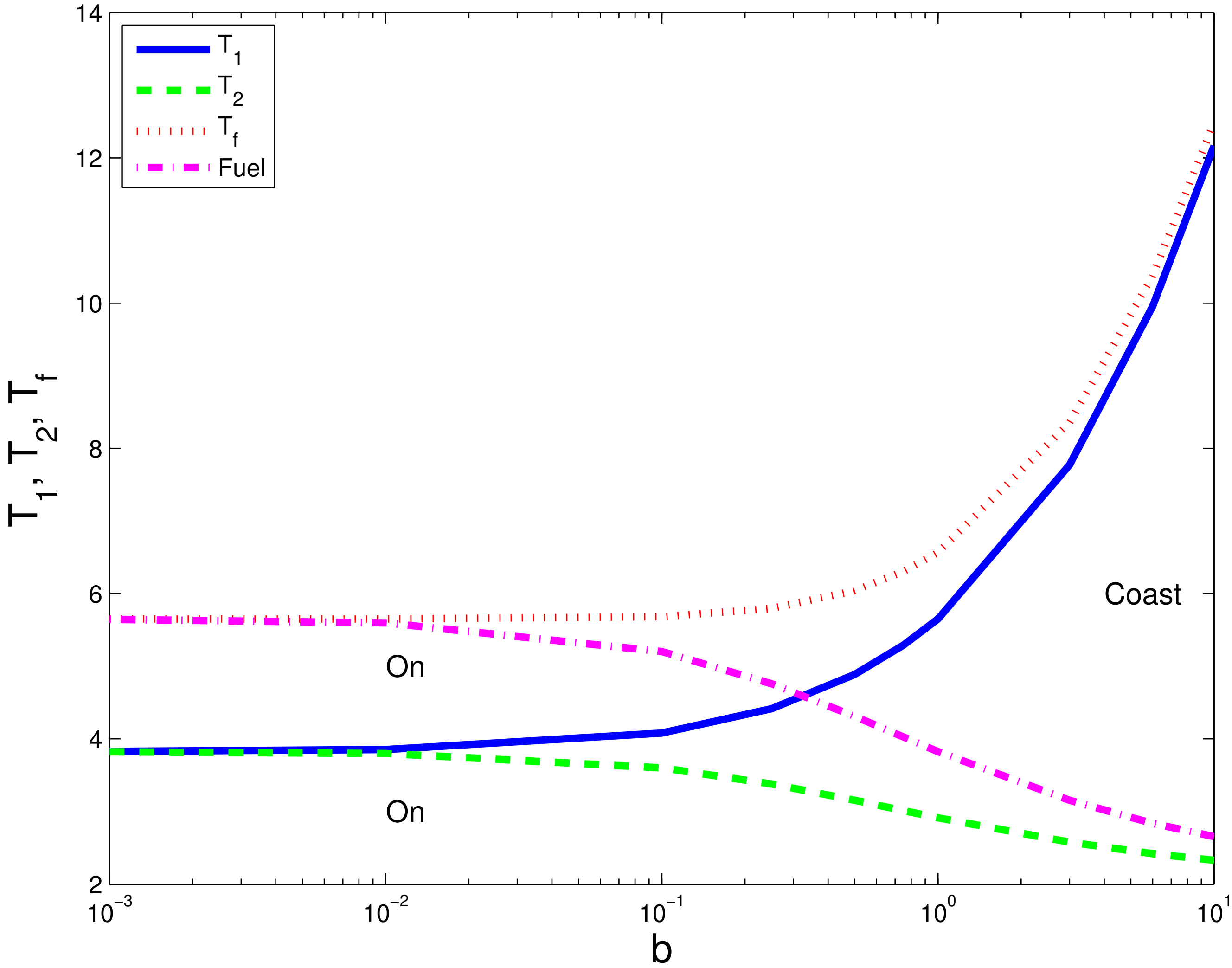


Figure 9.11: Summary of switching times for various fuel weights

Min time fuel

```
% Min time fuel for double integrator
% 16.323 Spring 2008
% Jonathan How
figure(1);clf;%
if jcase==1;y0=2;yd0=3; b=.75;u_m=1.5;% baseline
elseif jcase==2;y0=2;yd0=3; b=2;u_m=1.5;% fuel exp
elseif jcase==3;y0=2;yd0=3; b=.1;u_m=1.5;% fuel cheap
elseif jcase==4;y0=2;yd0=3; b=0;u_m=1.5;% min time
elseif jcase==5;y0=-4;yd0=4; b=1;u_m=1.5;% min time
end

% Tf is unknown - put together the equations to solve for it
alp=(1/2/u_m+2*b) % switching line
% middle of 8--6: t_2 as a ftn of t_f
T2=[1/u_m (2*b+1/u_m)*yd0/u_m]/(2*b+2/u_m);%
% bottom of 8--7: quadratic for y(t_2) in terms of t_2
% converted into quad in t_f
T_f=roots(-u_m/2*conv(T2,T2)+yd0*[0 T2]+[0 0 y0]  - ...
alp*conv(-u_m*T2+[0 yd0],-u_m*T2+[0 yd0]));%
t_f=max(T_f);t=[0:.01:t_f]'; %

c_1=(2*b+2/u_m)/(t_f-yd0/u_m);% key parameters for p(t)
c_2=c_1*t_f-(b+1/u_m);% key parameters for p(t)
t_1=t_f-1/(u_m*c_1); t_2=t_f-(2*b+1/u_m)/c_1;%switching times

G=ss([0 1;0 0],[0 1]',eye(2),zeros(2,1));
arc1=[0:.001:t_2]'; arc2=[t_2:.001:t_1]';arc3=[t_1:.001:t_f]'; %
if jcase==4;arc2=[t_2 t_2+1e-6]';end
[Y1,T1,X1]=lsim(G,-u_m*ones(length(arc1),1),arc1,[y0 yd0]'); %
[Y2,T2,X2]=lsim(G,0*ones(length(arc2),1),arc2,Y1(end,:)'); %
[Y3,T3,X3]=lsim(G,u_m*ones(length(arc3),1),arc3,Y2(end,:)'); %
plot(Y1(:,1),Y1(:,2),'Linewidth',2); hold on%
plot(Y2(:,1),Y2(:,2),'Linewidth',2); plot(Y3(:,1),Y3(:,2),'Linewidth',2);%
ylabel('dy/dt','Fontsize',18); xlabel('y(t)','Fontsize',12);%
text(-4,3,'y=-1/(2u_m)(dy/dt)^2','Fontsize',12)%
if jcase ~= 4; text(-5,0,'y=-(1/(2u_m)+2b)(dy/dt)^2','Fontsize',12);end
text(4,4,'-','Fontsize',18);text(-4,-4,'+','Fontsize',18);grid;hold off
title(['t_f = ',mat2str(t_f)],'Fontsize',12)%

hold on;% plot the switching curves
if jcase ~= 4;kk=[0:.1:5]'; plot(-alp*kk.^2,kk,'k--','Linewidth',2);plot(alp*kk.^2,-kk,'k--','Linewidth',2);end
kk=[0:.1:5]';plot(-(1/(2*u_m))*kk.^2,kk,'k--','Linewidth',2);plot((1/(2*u_m))*kk.^2,-kk,'k--','Linewidth',2);%
hold off;axis([-4 4 -4 4]/4*6);

figure(2);p2=c_2-c_1*t;%
plot(t,p2,'Linewidth',4);%
hold on; plot([0 max(t)],[b b],'k--','Linewidth',2);hold off; %
hold on; plot([0 max(t)],-[b b],'k--','Linewidth',2);hold off; %
hold on; plot([t_1 t_1],[-2 2],'k:','Linewidth',3);hold off; %
text(t_1+.1,1.5,'t_1','Fontsize',12)%
hold on; plot([t_2 t_2],[-2 2],'k:','Linewidth',3);hold off; %
text(t_2+.1,-1.5,'t_2','Fontsize',12)%
title(['b = ',mat2str(b),' u_m = ',mat2str(u_m)],'Fontsize',12);%
ylabel('p_2(t)','Fontsize',12); xlabel('t','Fontsize',12);%
text(1,b+.1,'b','Fontsize',12);text(1,-b+.1,'-b','Fontsize',12)%
axis([0 t_f -3 3]);grid on; %
%
if jcase==1
print -f1 -dpng -r300 fopt5a.png;;print -f2 -dpng -r300 fopt5b.png;
elseif jcase==2
print -f1 -dpng -r300 fopt6a.png;print -f2 -dpng -r300 fopt6b.png;
elseif jcase==3
print -f1 -dpng -r300 fopt7a.png;print -f2 -dpng -r300 fopt7b.png;
elseif jcase==4
print -f1 -dpng -r300 fopt8a.png;print -f2 -dpng -r300 fopt8b.png;
end
```

Minimum Time Problems

- Can repeat this analysis for minimum time and energy problems using the PMP
  - Issue is that the process of a developing a solution by analytic construction is laborious and very hard to extend to anything nonlinear and/or linear with more than 2 states
- Need to revisit the problem statement and develop a new approach.
- **Goal:** develop the control input sequence

$$M_i^- \leq u_i(t) \leq M_i^+$$

that drives the system (nonlinear, but linear control inputs)

$$\dot{\mathbf{x}} = A(\mathbf{x}, t) + B(\mathbf{x}, t)\mathbf{u}$$

from an arbitrary state $\mathbf{x}_0$ to the origin to minimize maneuver time

$$\min J = \int_{t_0}^{t_f} dt$$

- **Solution:** form the Hamiltonian

$$\begin{aligned} H &= 1 + \mathbf{p}^T(t)\{A(\mathbf{x}, t) + B(\mathbf{x}, t)\mathbf{u}\} \\ &= 1 + \mathbf{p}^T(t)\{A(\mathbf{x}, t) + \begin{bmatrix} \mathbf{b}_1(\mathbf{x}, t) & \mathbf{b}_2(\mathbf{x}, t) & \cdots & \mathbf{b}_m(\mathbf{x}, t) \end{bmatrix} \mathbf{u}\} \\ &= 1 + \mathbf{p}^T(t)A(\mathbf{x}, t) + \sum_{i=1}^{m} \mathbf{p}^T(t)\mathbf{b}_i(\mathbf{x}, t)u_i(t) \end{aligned}$$

- Now use the PMP: select $u_i(t)$ to minimize H, which gives

$$u_i(t) = \begin{cases} M_i^+ & \texttt{if} \;\; \mathbf{p}^T(t)\mathbf{b}_i(\mathbf{x}, t) < 0 \\ M_i^- & \texttt{if} \;\; \mathbf{p}^T(t)\mathbf{b}_i(\mathbf{x}, t) > 0 \end{cases}$$

which gives us the expected **Bang-Bang** control

- Then solve for the costate

$$\dot{\mathbf{p}} = -H_{\mathbf{x}}^T = -\left(\frac{\partial A}{\partial \mathbf{x}} + \frac{\partial B}{\partial \mathbf{x}}u\right)^T \mathbf{p}$$

  - Could be very complicated for a nonlinear system.

- Note: shown how to pick $u(t)$ given that $\mathbf{p}^T(t)\mathbf{b}_i(\mathbf{x},t) \neq 0$
  - Not obvious what to do if $\mathbf{p}^T(t)\mathbf{b}_i(\mathbf{x},t) = 0$ for some finite time interval.
  - In this case the coefficient of $u_i(t)$ is zero, and PMP provides no information on how to pick the control inputs.
  - Will analyze this **singular condition** in more detail later.

- To develop further insights, restrict the system model further to LTI, so that

$$A(\mathbf{x},t) \rightarrow A\mathbf{x} \qquad B(\mathbf{x},t) \rightarrow B$$

  - Assume that $[A, B]$ controllable
  - Set $M_i^+ = -M_i^- = u_{m_i}$

- Just showed that if a solution exists, it is Bang-Bang
  - **Existence:** if $\mathbb{R}(\lambda_i(A)) \leq 0$, then an optimal control exists that transfers any initial state $\mathbf{x}_0$ to the origin.
    - ◇ Must eliminate unstable plants from this statement because the control is bounded.
  - **Uniqueness:** If an extremal control exists (i.e. solves the necessary condition and satisfies the boundary conditions), then it is unique.
    - ◇ Satisfaction of the PMP is both necessary and sufficient for time-optimal control of a LTI system.

- If the eigenvalues of A are all real, and a unique optimal control exists, **then each control input can switch at most $n-1$ times.**
  - Still need to find the costates to determine the switching times – but much easier in the linear case.

Min Fuel Problems

- **Goal:** develop the control input sequence

$$M_i^- \leq u_i(t) \leq M_i^+$$

that drives the system

$$\dot{\mathbf{x}} = A(\mathbf{x}, t) + B(\mathbf{x}, t)\mathbf{u}$$

from an arbitrary state $\mathbf{x}_0$ to the origin in a fixed time t_f and optimizes the cost

$$\min J = \int_{t_0}^{t_f} \sum_{i=1}^{m} c_i |u_i(t)| dt$$

- **Solution:** form the Hamiltonian

$$\begin{aligned} H &= \sum_{i=1}^{m} c_i |u_i(t)| + \mathbf{p}^T(t)\{A(\mathbf{x}, t) + B(\mathbf{x}, t)\mathbf{u}\} \\ &= \sum_{i=1}^{m} c_i |u_i(t)| + \mathbf{p}^T(t) A(\mathbf{x}, t) + \sum_{i=1}^{m} \mathbf{p}^T(t)\mathbf{b}_i(\mathbf{x}, t) u_i(t) \\ &= \sum_{i=1}^{m} \left[c_i |u_i(t)| + \mathbf{p}^T(t)\mathbf{b}_i(\mathbf{x}, t) u_i(t) \right] + \mathbf{p}^T(t) A(\mathbf{x}, t) \end{aligned}$$

- Use the PMP, which requires that we select $u_i(t)$ to ensure that for all admissible $u_i(t)$

$$\sum_{i=1}^{m} \left[c_i |u_i^\star(t)| + \mathbf{p}^T(t)\mathbf{b}_i(\mathbf{x}, t) u_i^\star(t) \right] \leq \sum_{i=1}^{m} \left[c_i |u_i(t)| + \mathbf{p}^T(t)\mathbf{b}_i(\mathbf{x}, t) u_i(t) \right]$$

- If the components of $\mathbf{u}$ are independent, then can just look at

$$c_i |u_i^\star(t)| + \mathbf{p}^T(t)\mathbf{b}_i(\mathbf{x}, t) u_i^\star(t) \leq c_i |u_i(t)| + \mathbf{p}^T(t)\mathbf{b}_i(\mathbf{x}, t) u_i(t)$$

  – As before, this boils down to a comparison of c_i and $\mathbf{p}^T(t)\mathbf{b}_i$

  – Resulting control law is:

$$u_i^\star(t) = \begin{cases} M_i^- & \texttt{if} \quad c_i < \mathbf{p}^T(t)\mathbf{b}_i \\ 0 & \texttt{if} \quad -c_i < \mathbf{p}^T(t)\mathbf{b}_i < c_i \\ M_i^+ & \texttt{if} \quad \mathbf{p}^T(t)\mathbf{b}_i < -c_i \end{cases}$$

Example: 9–1

- Consider $G(s) = 1/s^2 \Rightarrow A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

$$\min J = \int_{t_0}^{t_f} |u(t)| dt$$

  – Drive state to the origin with t_f fixed.

- Gives $H = |u| + p_1 x_2 + p_2 u$
  – Final control $u(t_f) = u_m \Rightarrow p_2(t_f) < -1$ $p_2(t) = c_2 - c_1 t$

- As before, integrate EOM forward from 0 to t_2 using $-u_m$, then from t_2 to t_1 using $u = 0$, and from t_1 to t_f using u_m
  – Apply terminal conditions and solve for c_1 and c_2

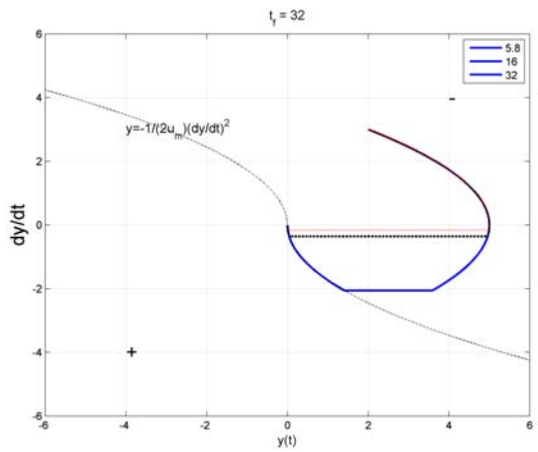


Figure 9.12: Min Fuel for varying final times

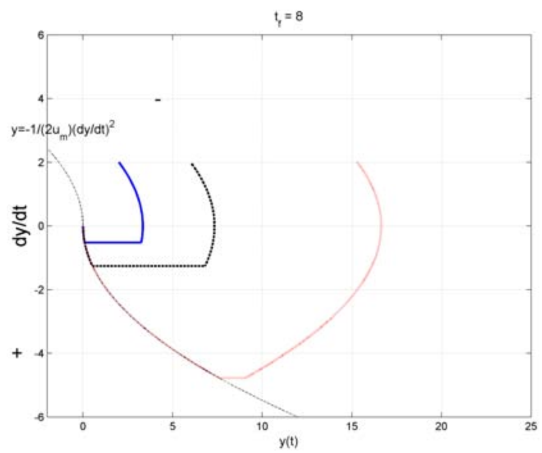


Figure 9.13: Min fuel for fixed final time, varying IC's

- First switch depends on IC and $t_f \Rightarrow$ no clean closed-form solution for switching curve
  – Larger t_f leads to longer coast.
  – For given t_f, there is a limit to the IC from which we can reach the origin.

- If specified completion time $t_f > T_{\min} = \dot{y}(0) + 2\sqrt{y(0) + 0.5\dot{y}^2(0)}$, then

$$\begin{aligned}
t_2 &= 0.5\left\{(\dot{y}(0) + t_f) - \sqrt{(\dot{y}(0) - t_f)^2 - (4y(0) + 2\dot{y}^2(0))}\right\} \\
t_1 &= 0.5\left\{(\dot{y}(0) + t_f) + \sqrt{(\dot{y}(0) - t_f)^2 - (4y(0) + 2\dot{y}^2(0))}\right\}
\end{aligned}$$

Minimum Energy Problem

- Goal: for a fixed final time and terminal constraints

$$\min J = \frac{1}{2}\int_0^{t_f} \mathbf{u}^T R \mathbf{u}\, dt \qquad R > 0$$

- Again use special dynamics:

$$\begin{aligned} \dot{\mathbf{x}} &= A(\mathbf{x},t) + B(\mathbf{x},t)\mathbf{u} \\ H &= \frac{1}{2}\mathbf{u}^T R\mathbf{u} + \mathbf{p}^T\{\ A(\mathbf{x},t) + B(\mathbf{x},t)\mathbf{u}\} \end{aligned}$$

- Obviously with no constraints on $\mathbf{u}$, solve $H_{\mathbf{u}} = 0$, to get

$$\mathbf{u} = -R^{-1}B^T\mathbf{p}(t)$$

- But with bounded controls, must solve:

$$\mathbf{u}^\star(t) = \arg \min_{\mathbf{u}(t)\in\mathcal{U}} \left[\frac{1}{2}\mathbf{u}^T R\mathbf{u} + \mathbf{p}^T B(\mathbf{x},t)\mathbf{u}\right]$$

which is a constrained quadratic program in general

– However, for diagonal R, the effects of the controls are independent

$$\mathbf{u}^\star(t) = \arg \min_{\mathbf{u}(t)\in\mathcal{U}} \left[\sum_{i=1}^{m} \frac{1}{2}R_{ii}u_i^2 + \mathbf{p}^T\mathbf{b}_i u_i\right]$$

– In the unconstrained case, each $u_i(t)$ can easily be determined by minimizing

$$\frac{1}{2}R_{ii}u_i^2 + \mathbf{p}^T\mathbf{b}_i u_i \qquad \rightarrow \qquad \tilde{u}_i = -R_{ii}^{-1}\mathbf{p}^T\mathbf{b}_i$$

- The resulting controller inputs are $u_i(t) = \texttt{sat}(\tilde{u}_i(t))$

$$u_i(t) = \begin{cases} M_i^- & \texttt{if} \qquad \tilde{u}_i < M_i^- \\ \tilde{u}_i & \texttt{if}\ M_i^- < \tilde{u}_i < M_i^+ \\ M_i^+ & \texttt{if}\ M_i^+ < \tilde{u}_i \end{cases}$$

Min Fuel

```
% Min fuel for double integrator
% 16.323 Spring 2008
% Jonathan How
%
c=1;
t=[0:.01:t_f];
alp=(1/2/u_m) % switching line
T_2=roots([-u_m/2 yd0 y0] + conv([-u_m yd0],[-2 t_f+yd0/u_m])-alp*conv([-u_m yd0],[-u_m yd0]));%
t_2=min(T_2);
yd2=-u_m*t_2+yd0;yd1=yd2;
t_1=t_f+yd1/u_m;
c_1=2/(t_1-t_2);c_2=c_1*t_1-1;

G=ss([0 1;0 0],[0 1]',eye(2),zeros(2,1));
arc1=[0:.001:t_2]'; arc2=[t_2:.001:t_1]';arc3=[t_1:.001:t_f]'; %
[Y1,T1,X1]=lsim(G,-u_m*ones(length(arc1),1),arc1,[y0 yd0]'); %
[Y2,T2,X2]=lsim(G,0*ones(length(arc2),1),arc2,Y1(end,:)'); %
[Y3,T3,X3]=lsim(G,u_m*ones(length(arc3),1),arc3,Y2(end,:)'); %
plot(Y1(:,1),Y1(:,2),zzz,'Linewidth',2); hold on%
plot(Y2(:,1),Y2(:,2),zzz,'Linewidth',2); plot(Y3(:,1),Y3(:,2),zzz,'Linewidth',2);%
ylabel('dy/dt','Fontsize',18); xlabel('y(t)','Fontsize',12);%
text(-4,3,'y=-1/(2u_m)(dy/dt)^2','Fontsize',12)%
text(4,4,'-','Fontsize',18);text(-4,-4,'+','Fontsize',18);grid on;hold off
title(['t_f = ',mat2str(t_f)],'Fontsize',12)%

hold on;% plot the switching curves
kk=[0:.1:8]'; plot(-alp*kk.^2,kk,'k--');plot(alp*kk.^2,-kk,'k--');
hold off;axis([-4 4 -4 4]/4*6);

figure(2);%
p2=c_2-c_1*t;%
plot(t,p2,'Linewidth',4);%
hold on; plot([0 t_f],[c c],'k--','Linewidth',2);hold off; %
hold on; plot([0 t_f],-[c c],'k--','Linewidth',2);hold off; %
hold on; plot([t_1 t_1],[-2 2],'k:','Linewidth',3);hold off; %
text(t_1+.1,1.5,'t_1','Fontsize',12)%
hold on; plot([t_2 t_2],[-2 2],'k:','Linewidth',3);hold off; %
text(t_2+.1,-1.5,'t_2','Fontsize',12)%
title(['c = ',mat2str(c),' u_m = ',mat2str(u_m)],'Fontsize',12);%
ylabel('p_2(t)','Fontsize',12); xlabel('t','Fontsize',12);%
text(1,c+.1,'c','Fontsize',12);text(1,-c+.1,'-c','Fontsize',12)%
axis([0 t_f -3 3]);grid on; %

return

figure(1);clf
y0=2;yd0=3;t_f=5.8;u_m=1.5;zzz='-';minu;
figure(1);hold on
y0=2;yd0=3;t_f=16;u_m=1.5;zzz='k--';minu;
figure(1);hold on
y0=2;yd0=3;t_f=32;u_m=1.5;zzz='r:';minu;
figure(1);
axis([-6 6 -6 6])
legend('5.8','16','32')
print -f1 -dpng -r300 uopt1.png;


figure(1);clf
y0=2;yd0=2;t_f=8;u_m=1.5;zzz='-';minu
figure(1);hold on
y0=6;yd0=2;t_f=8;u_m=1.5;zzz='k--';minu
figure(1);hold on
y0=15.3;yd0=2;t_f=8;u_m=1.5;zzz='r:';minu
figure(1);axis([-2 25 -6 6])
print -f1 -dpng -r300 uopt2.png;
```