ECE 6553: Homework #3

Klaus Okkelberg

March 14, 2017

1. Let the Lagrange multiplier for this new problem be $\mu(t) = [\mu_x^{\mathrm{T}}(t)\ \mu_\lambda^{\mathrm{T}}(t)]^{\mathrm{T}}$. Then, the Hamiltonian is $H = \mu^{\mathrm{T}} F$.

   The optimal λ_0 that solves the Bolza problem results in $\mu_\lambda(0) = 0$, where $\mu(t)$ satisfies

$$
\begin{aligned}
\dot{\mu} &= -\frac{\partial H^{\mathrm{T}}}{\partial z} = -\frac{\partial}{\partial z}\begin{bmatrix}\dot{x} & \dot{\lambda}\end{bmatrix}\begin{bmatrix}\mu_x \\ \mu_\lambda\end{bmatrix} \\
&= -\frac{\partial}{\partial z}\left\{ f^{\mathrm{T}}(x, u^*(x,\lambda))\mu_x - \frac{\partial H^{\mathrm{T}}(x, u^*(x,\lambda), \lambda)}{\partial x}\mu_\lambda \right\} \\
&= \begin{bmatrix} -\dfrac{\partial f^{\mathrm{T}}}{\partial x}\mu_x + \dfrac{\partial}{\partial x}\dfrac{\partial H^{\mathrm{T}}}{\partial x}\mu_\lambda \\ -\dfrac{\partial f^{\mathrm{T}}}{\partial \lambda}\mu_x + \dfrac{\partial}{\partial \lambda}\dfrac{\partial H^{\mathrm{T}}}{\partial x}\mu_\lambda \end{bmatrix} \\
\mu(T) &= \frac{\partial C(z(T))}{\partial z} = \frac{\partial}{\partial z}\left\| \lambda(T) - \frac{\partial \Psi(x(T))}{\partial x} \right\|^2 \\
&= \begin{bmatrix} -2\left(\lambda(T) - \dfrac{\partial \Psi(x(T))}{\partial x}\right)^{\mathrm{T}} \dfrac{\partial^2 \Psi(x(T))}{\partial x^2} \\ 2\left(\lambda(T) - \dfrac{\partial \Psi(x(T))}{\partial x}\right) \end{bmatrix}
\end{aligned}
$$

   For given $\gamma, \varepsilon > 0$, the algorithm that finds the optimal λ_0 is given by

---

Pick λ_0
$z(0) \coloneqq \begin{bmatrix} x_0 \\ \lambda_0 \end{bmatrix}$
repeat
  Simulate $z(t)$ from $z(0)$ over $[0, T]$
  Simulate $\mu(t)$ backwards from $\mu(T)$ using $z(t)$
  Update $\lambda_0 \coloneqq \lambda_0 - \gamma\mu_\lambda(0)$
  Update $z(0) \coloneqq \begin{bmatrix} x_0 \\ \lambda_0 \end{bmatrix}$
until $\|\mu_\lambda(0)\| \le \varepsilon$

---

2. (a)

$$H = \frac{1}{2}(u_1^2 + u_2^2) + \lambda_1 u_1 + \lambda_2 u_2 + \lambda_3(x_1 u_2 - x_2 u_1)$$

$$\frac{\partial H}{\partial u_1} = u_1 + \lambda_1 - \lambda_3 x_2 = 0$$

$$\frac{\partial H}{\partial u_2} = u_2 + \lambda_2 + \lambda_3 x_1 = 0$$

$$\implies \boxed{u^* = \begin{bmatrix} -\lambda_1 + \lambda_3 x_2 \\ -\lambda_2 - \lambda_3 x_1 \end{bmatrix}}$$

$$\dot{\lambda} = -\frac{\partial H}{\partial x}^{\mathrm{T}}$$

$$\implies \boxed{\dot{\lambda} = \begin{bmatrix} -\lambda_3 u_2 \\ \lambda_3 u_1 \\ 0 \end{bmatrix}}$$

$$\lambda(T) = \frac{\partial \Psi}{\partial x}^{\mathrm{T}}$$

$$\implies \boxed{\lambda(T) = \begin{bmatrix} 0 \\ 0 \\ -\rho x_3(T) \end{bmatrix}}$$

(b) The dynamics of $z(t)$ are

$$F = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{\lambda}_1 \\ \dot{\lambda}_2 \\ \dot{\lambda}_3 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ x_1 u_2 - x_2 u_1 \\ -\lambda_3 u_2 \\ \lambda_3 u_1 \\ 0 \end{bmatrix} = \begin{bmatrix} -\lambda_1 + \lambda_3 x_2 \\ -\lambda_2 - \lambda_3 x_1 \\ x_1(-\lambda_2 - \lambda_3 x_1) - x_2(-\lambda_1 + \lambda_3 x_2) \\ -\lambda_3(-\lambda_2 - \lambda_3 x_1) \\ \lambda_3(-\lambda_1 + \lambda_3 x_2) \\ 0 \end{bmatrix}$$

Thus, the dynamics of the costate are

$$\dot{\mu} = -\frac{\partial}{\partial z} F^{\mathrm{T}} \mu = \begin{bmatrix} -\lambda_3 \mu_{x_2} - (\lambda_2 + 2\lambda_3 x_1)\mu_{x_3} + \lambda_3^2 \mu_{\lambda_1} \\ \lambda_3 \mu_{x_1} + (\lambda_1 - 2\lambda_3 x_2)\mu_{x_3} + \lambda_3^2 \mu_{\lambda_2} \\ 0 \\ -\mu_{x_1} + x_2 \mu_{x_3} - \lambda_3 \mu_{\lambda_2} \\ -\mu_{x_2} - x_1 \mu_{x_3} + \lambda_3 \mu_{\lambda_1} \\ x_2 \mu_{x_1} - x_1 \mu_{x_2} - (x_1^2 + x_2^2)\mu_{x_3} + (\lambda_2 + 2\lambda_3 x_1)\mu_{\lambda_1} + (-\lambda_1 + 2\lambda_3 x_2)\mu_{\lambda_2} \end{bmatrix}$$

Additionally, using

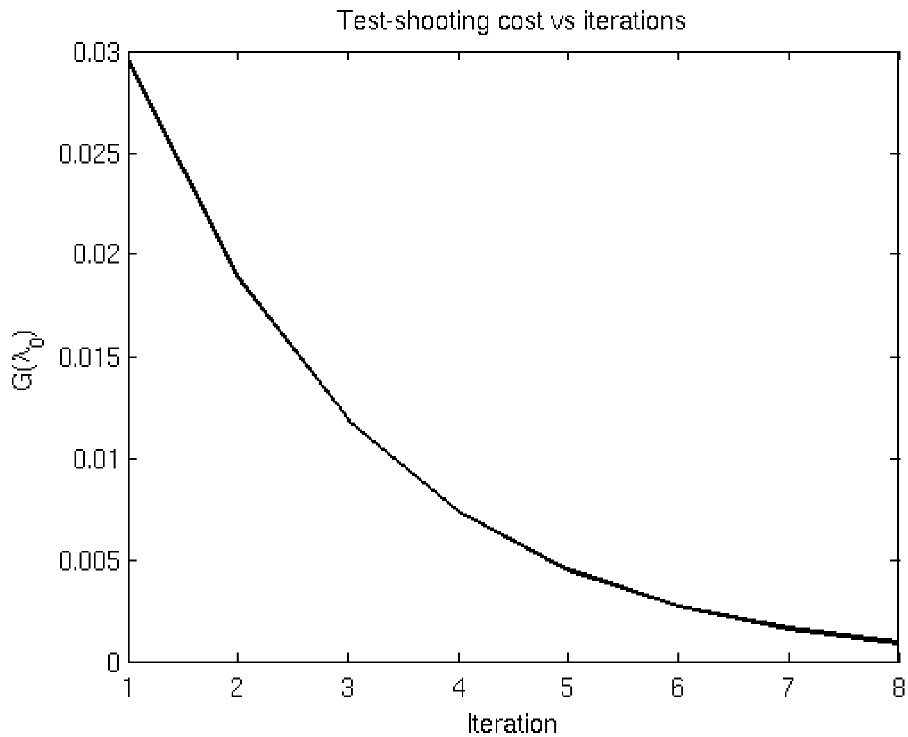
$$\frac{\partial \Psi}{\partial x} = \begin{bmatrix} 0 \\ 0 \\ -\rho x_3 \end{bmatrix}, \qquad \frac{\partial^2 \Psi}{\partial x^2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -\rho \end{bmatrix}$$

the terminal boundary conditions of the costate are

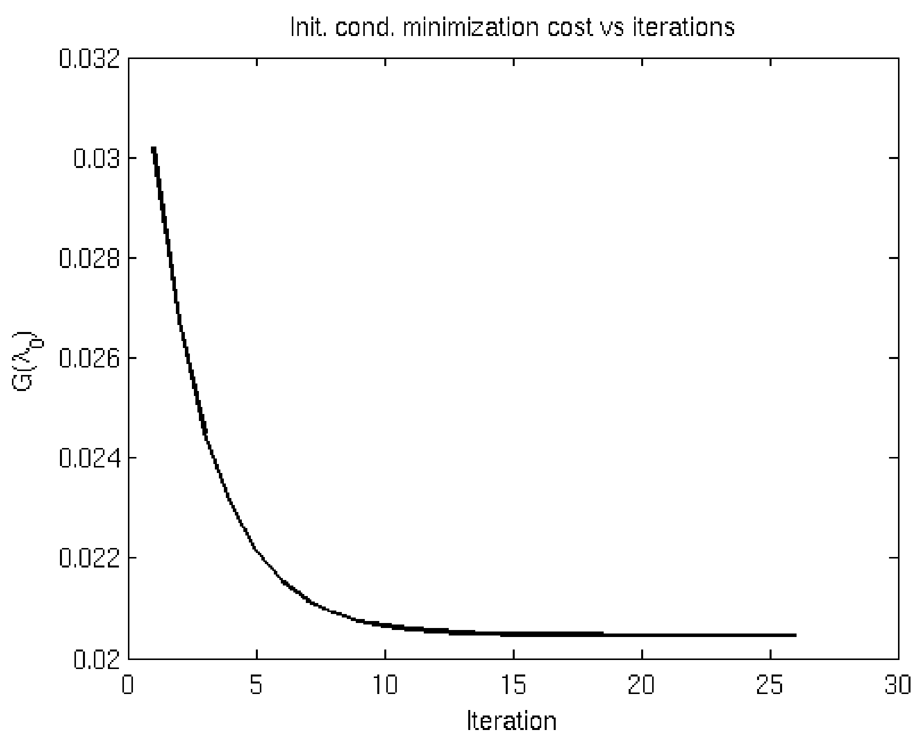
$$\mu(T) = \frac{\partial}{\partial z} \left\| \lambda(T) - \frac{\partial \Psi(x(T))}{\partial x} \right\|^2 = \begin{bmatrix} 0 \\ 0 \\ 2\rho(\lambda_3(T) + \rho x_3(T)) \\ 0 \\ 0 \\ 2(\lambda_3(T) + \rho x_3(T)) \end{bmatrix}$$

The optimal λ_0 results in $\mu_\lambda(0) = 0$.

3. (a) The optimization condition was $G(\lambda_0) \leq 10^{-3}$.



(b) The optimization condition was $\|\mu_\lambda(0)\| \leq 10^{-3}$.



(c) The test shooting method was both more efficient and more accurate. It achieved a lower final cost and needed less iterations to achieve that cost. I think this is because the test shooting method was created expressly for solving this two-point boundary value problem, while the other algorithm was adapted for this problem. This means the descent direction for the other algorithm is optimal for reducing $\|\mu_\lambda(0)\|$ and non-optimal for reducing $G(\lambda_0)$. This means that the test shooting method needs less iterations since each descent step results in a greater decrease of the cost.

Furthermore, the new problem has double the number of dimensions, which could have introduced local minima such that $G(\lambda_0) > 0$ when $\|\mu_\lambda(0)\| = 0$. Our choice of λ_0 could be near one of such local minima. This would explain why the cost for the second algorithm flattens out around $G(\lambda_0) \approx 0.0205$.

Moreover, from an implementation standpoint, the test shooting method was also easier to implement, because we only need to solve forward in time so we do not need to store the entire trajectories $x(t)$ and $\lambda(t)$. Rather, only the final values $x(T)$ and $\lambda(T)$ are needed. The second algorithm needs the entire $z(t)$ in order to solve backwards for $\mu(t)$.

```
% Klaus Okkelberg
% ECE 6553
% HW3 P3

% Numerical parameters
delta = 1e-3; % optimization termination
epsilon = 1e-2; % test shooting step
gamma = 1e-1; % descent step
dt = 1e-2; % discrete integration step size

% problem parameters
T = 1;
p = 2;
lambda0 = 0.1*ones(3,1);
x0 = zeros(3,1);

%% Part a
G = inf;
Ghist = [];
while G > delta
    t = 0;
    X = x0;
    lambda = lambda0;

    % Run the test shooting
    for idx = 1:3
        ei=zeros(3,1); ei(idx)=1;
        X = [X x0];
        lambda = [lambda lambda0+epsilon*ei];
    end

    % Solve for lambda0 as well as for the perturbed initial conditions
    while t <= T
        for idx = 1:4 % Position 1 is the nominal system
            x = X(:,idx);
            l = lambda(:,idx);
            u = [-l(1) + l(3)*x(2);
                -l(2) - l(3)*x(1)];
            dx = [u(1); u(2); x(1)*u(2)-x(2)*u(1)];
            dl = [-l(3)*u(2); l(3)*u(1); 0];
            x = x + dt*dx;
```

```
            l = l + dt*dl;
            X(:,idx) = x;
            lambda(:,idx) = l;
        end
        t = t + dt;
    end

    % Compute the gradient dg/dlambda0
    G = lambda(1)^2 + lambda(2)^2 + (lambda(3) + p*X(3))^2;
    dG = zeros(3,1);
    for idx = 2:4
        Gi = lambda(1,idx)^2 + lambda(2,idx)^2 + (lambda(3,idx) + p*X(3,idx))^2;
        dG(idx-1) = (Gi - G)/epsilon;
    end

    % gradient descent
    lambda0 = lambda0 - gamma*dG;
    Ghist = [Ghist G];
end

% plot cost
figure('DefaultAxesFontSize',12)
plot(Ghist,'k','LineWidth',2)
xlabel('Iteration')
ylabel('G(\lambda_0)')
title('Test-shooting cost vs iterations')

%% Part b
% reset lambda0
lambda0 = 0.1*ones(3,1);
% make descent step larger

G = inf;
Ghist = [];
mu = inf(6,1);
while norm(mu(4:6,1)) > delta
    % Solve for z(t)
    z = zeros(6,T/dt+1);
    z(:,1) = [x0; lambda0];
    for idx = 1:T/dt
        u = [-z(4) + z(6)*z(2);
            -z(5) - z(6)*z(1)];
        dz = [u(1);
            u(2);
            z(1)*u(2) - z(2)*u(1);
            -z(6)*u(2);
            z(6)*u(1);
            0];
        z(:,idx+1) = z(:,idx) + dt*dz;
    end

    % Solve backwards for mu(t)
    mu = zeros(6,T/dt+1);
    mu(:,end) = [0; 0; 2*p*(z(6,end) + p*z(3,end));
        0; 0; 2*(z(6,end) + p*z(3,end))];
    for idx = T/dt+1:-1:2
        dF = [0 z(6,idx) 0 -1 0 z(2,idx);
            -z(6,idx) 0 0 0 -1 -z(1,idx);
            -z(5,idx)-2*z(6,idx)*z(1,idx) z(4,idx)-2*z(6,idx)*z(2,idx) 0 ...
```

```
            z(2,idx) -z(1,idx) -z(1,idx)^2-z(2,idx)^2;
            z(6,idx)^2 0 0 0 z(6,idx) z(5,idx)+2*z(1,idx)*z(6,idx);
            0 z(6,idx)^2 0 -z(6,idx) 0 -z(4,idx)+2*z(2,idx)*z(6,idx);
            0 0 0 0 0 0];
        dmu = -dF'*mu(:,idx);
        mu(:,idx-1) = mu(:,idx) - dt*dmu;
    end

    % Calculate cost
    G = z(4,end)^2 + z(5,end)^2 + (z(6,end) + p*z(3,end))^2;

    % gradient descent
    lambda0 = lambda0 - gamma*mu(4:6,1);
    Ghist = [Ghist G];
end

% plot cost
figure('DefaultAxesFontSize',12)
plot(Ghist,'k','LineWidth',2)
xlabel('Iteration')
ylabel('G(\lambda_0)')
title('Init. cond. minimization cost vs iterations')
```

4. I assumed that $T > 1$. Notice that in addition to an initial condition, there is also an intermediate condition at $t = 1$. Therefore, the costate could have a discontinuity at $t = 1$. The optimality conditions are

$$
\begin{aligned}
H &= \frac{1}{2}(x^2 + u^2) + \lambda u \\
\frac{\partial H}{\partial u} &= u + \lambda = 0 \\
u &= -\lambda \\
\dot{\lambda} &= -\frac{\partial H}{\partial x} = -x \\
\lambda(T) &= \frac{\partial H}{\partial x} = 0
\end{aligned}
$$

Taking the second derivative of $x(t)$ results in

$$\ddot{x} = \dot{u} = -\dot{\lambda} = x.$$

The characteristic equation $z^2 - 1 = 0$ has roots of ± 1, so the optimal $x(t)$ has the form

$$x(t) = c_1 e^t + c_2 e^{-t}.$$

Since $\dot{x} = u$,

$$u(t) = c_1 e^T - c_2 e^{-t}.$$

For $t \in [0, 1]$, the boundary conditions are

$$
\begin{aligned}
x(0) &= c_1 + c_2 = 1 \\
x(1) &= c_1 e + c_2 e^{-1} = 1
\end{aligned}
$$

Substituting $c_2 = 1 - c_1$ from the first equation into the second,

$$
\begin{aligned}
c_1 e + (1 - c_1)e^{-1} &= 1 \\
c_1(e - e^{-1}) &= 1 - e^{-1} \\
c_1 &= \frac{1 - e^{-1}}{e - e^{-1}} \\
c_2 &= \frac{e - 1}{e - e^{-1}} \\
u(t) &= \frac{1 - e^{-1}}{e - e^{-1}} e^t - \frac{e - 1}{e - e^{-1}} e^{-t}, \quad 0 \le t \le 1
\end{aligned}
$$

For $t \in [1, T]$, the boundary conditions are

$$
\begin{aligned}
x(1) &= c_1 e + c_2 e^{-1} = 1 \\
\lambda(T) &= -u(T) = -\dot{x}(T) = -c_1 e^T + c_2 e^{-T} = 0
\end{aligned}
$$

Equating c_2 from the two equations results in

$$
\begin{aligned}
e - c_1 e^2 = c_2 &= c_1 e^{2T} \\
c_1 &= \frac{e}{e^{2T} + e^2} \\
c_2 = e - \frac{e}{e^{2T} + e^2} e^2 &= \frac{e^{2T+1}}{e^{2T} + e^2} \\
u(t) = \frac{e}{e^{2T} + e^2} e^t &- \frac{e^{2T+1}}{e^{2T} + e^2} e^{-t}, \quad t > 1
\end{aligned}
$$

Thus, the optimal control is

$$
\boxed{u^* = \begin{cases} \dfrac{1 - e^{-1}}{e - e^{-1}} e^t - \dfrac{e - 1}{e - e^{-1}} e^{-t}, & 0 \le t < 1 \\ \dfrac{e}{e^{2T} + e^2} e^t - \dfrac{e^{2T+1}}{e^{2T} + e^2} e^{-t}, & 1 \le t \le T \end{cases}}
$$

5. (a) The augmented cost is

$$
\tilde{J}(p) = \int_0^T [L + \lambda^{\mathrm{T}}(f - \dot{x})] \, \mathrm{d}t = \int_0^T [H - \lambda^{\mathrm{T}} \dot{x}] \, \mathrm{d}t
$$

A variation $p \mapsto p + \varepsilon$ results in $x \mapsto x + \varepsilon\eta$, where $\eta(0) = 0$ since $x(0)$ is fixed. Thus,

$$
\begin{aligned}
\tilde{J}(p + \varepsilon) &= \int_0^T [H(x + \varepsilon\eta, p + \varepsilon) - \lambda^{\mathrm{T}}(\dot{x} + \varepsilon\dot{\eta})] \, \mathrm{d}t \\
&= \int_0^T \left[H + \varepsilon \frac{\partial H}{\partial x} \eta + \varepsilon \frac{\partial H}{\partial p} - \lambda^{\mathrm{T}}(\dot{x} + \varepsilon\dot{\eta}) \right] \mathrm{d}t + o(\varepsilon) \\
\tilde{J}(p + \varepsilon) - \tilde{J}(p) &= \int_0^T \left[\varepsilon \frac{\partial H}{\partial x} \eta + \varepsilon \frac{\partial H}{\partial p} - \lambda^{\mathrm{T}} \dot{\eta} \right] \mathrm{d}t + o(\varepsilon)
\end{aligned}
$$

Taking the derivative with respect to p and using integration by parts,

$$\begin{aligned}
\frac{\partial \tilde{J}(p)}{\partial p} &= \lim_{\varepsilon \to 0} \frac{\tilde{J}(p+\varepsilon) - \tilde{J}(p)}{\varepsilon} \\
&= \int_0^T \left[\frac{\partial H}{\partial x}\eta + \frac{\partial H}{\partial p} - \lambda^{\mathrm{T}}\dot{\eta} \right] \mathrm{d}t \\
&= \int_0^T \left[\frac{\partial H}{\partial x}\eta + \frac{\partial H}{\partial p} + \dot{\lambda}^{\mathrm{T}}\eta \right] \mathrm{d}t + \lambda^{\mathrm{T}}(0)\underbrace{\eta(0)}_{=0} - \lambda^{\mathrm{T}}(T)\eta(T) \\
&= \int_0^T \left[\frac{\partial H}{\partial x} + \dot{\lambda}^{\mathrm{T}} \right] \eta \, \mathrm{d}t + \int_0^T \frac{\partial H}{\partial p} \, \mathrm{d}t - \lambda^{\mathrm{T}}(T)\eta(T) = 0
\end{aligned}$$

Thus, the optimality conditions are

$$\begin{aligned}
\frac{\partial H}{\partial p} &= \frac{\partial f}{\partial p}\lambda = 0 \\
\dot{\lambda} &= -\frac{\partial H^{\mathrm{T}}}{\partial x} = -\frac{\partial L^{\mathrm{T}}}{\partial x} - \frac{\partial f}{\partial x}\lambda \\
\lambda(T) &= 0
\end{aligned}$$

(b) We can write the second term of $\partial \tilde{J}(p)/\partial p$ as

$$\mu(T) = \int_0^T \frac{\partial H}{\partial p} \, \mathrm{d}t = 0.$$

since we want the term to be zero. Additionally, taking its derivative

$$\dot{\mu} = \frac{\partial H}{\partial p} = \frac{\partial f}{\partial p}\lambda.$$

Thus, the optimality conditions can be rewritten as

$$\begin{aligned}
\dot{\lambda} &= -\frac{\partial L^{\mathrm{T}}}{\partial x} - \frac{\partial f}{\partial x}\lambda \\
\dot{\mu} &= \frac{\partial f}{\partial p}\lambda \\
\lambda(T) &= 0 \\
\mu(T) &= 0
\end{aligned}$$

6. This is important observation for two reasons:

   1) The costate is dependent on the current state so we do not have to seperately solve the costate equation backwards from time T. In fact, using $u^* = -R^{-1}B^{\mathrm{T}}\lambda$ (shown on midterm), the optimal control is only a function of the state and we don't need to find the costate. This is very advantageous if T is large or infinite. This also avoids having to resort to numerics to find the optimal solution.
   2) Since the optimal control only depends on the current state, this control is optimal for any initial boundary condition on $x(0)$. Additionally, the control still functions if the state changes for some reason, e.g. from noise, due to this feedback.