

## Optimal Control

Control problem stated in terms of finding the control input for a given system which minimizes some **objective function**  $J$ .  
Function minimization

- Unconstrained
- Constrained
  - Equality constraints
  - Inequality constraints

Expand function using Taylor series

$$f(x + \Delta x) = f(x) + g^\top(x)\Delta x + \frac{1}{2}\Delta x^\top H(x)\Delta x + \dots$$

Want to find a minimum.

### Optimal Control Problem

The problem statement is

$$\begin{aligned} \text{Minimize } J &= h + \int_t^{t_f} g dt \\ \text{subject to } \dot{x} &= f \end{aligned}$$

Ways to solve

- Direct transcription
- Variational approach
- Dynamic programming
  - DP problems usually have to be solved numerically
  - Basic recursion of DP basically says to start at the end of time and work backwards, at each step minimizing the cost just to get you to the prior step
  - Considering infinite horizon LQR (discrete first) the result is the Algebraic Riccati Equation and a constant static gain feedback control law
  - DP in continuous time

Principle of optimality—every sub-trajectory of an optimal trajectory is optimal.

## Dynamic Programming

### Continuous Time

Start with same problem statement, break limits of integration into two intervals, from  $t$  to  $t + \Delta t$  and then from  $t + \Delta t$  to  $t_f$ . Use principle of optimality and express optimal cost as the cost to get from  $t$  to  $t + \Delta t$  and the cost-to-go to get from  $t + \Delta t$  to  $t_f$ . Expand the cost-to-go using a Taylor series expansion, and approximate the over the integration interval  $\Delta t$  by assuming  $\Delta t$  is small. After some algebra, the result is a PDE known as the Hamilton-Jacobi-Bellman (HJB) equation. We simplify the notation in the HJB equation by defining the Hamiltonian.

$$-J_t^* = \min_u \mathcal{H}$$

$$\mathcal{H} = g + J_x^* f$$

### LQR Using Dynamic Programming

Use the HJB equation. Write out the Hamiltonian and minimize with respect to  $u$  to find  $u^*$ . Propose a form for  $J^*$  plugging in  $u^*$  to derive differential Riccati equation.

## Variational Approach

Augment cost function with a Lagrange multiplier to enforce system dynamics. This Lagrange multiplier must be time varying, and we call the Lagrange multiplier the costate  $p$ . When we calculate the first variation and set to zero, a condition comes out specifying  $\dot{p}$ . Combining this differential equation with the one for the system dynamics we get a system of differential equations in  $x$  and  $p$ , expressed with the Hamiltonian matrix  $\bar{H}$ . To solve, we guess a form of the solution  $p$  and check that it works. The result is the Riccati

differential equation. For infinite time, this differential equation reaches steady-state and becomes an algebraic equation. We can solve it to obtain the optimal control law, and thus the closed-loop system poles. We see that these are the poles (positive and negative) of  $\bar{H}$  by doing a similarity transform on  $\bar{H}$ .

### Variational Approach to Optimal Control

$$\begin{aligned} \text{Minimize } J &= h + \int_t^{t_f} g dt \\ \text{subject to } \dot{x} &= f \end{aligned}$$

<b>Hamiltonian:</b>	$\mathcal{H} = g + p^\top f$
<b>Necessary Conditions:</b>	$\dot{x} = \mathcal{H}_p^\top = f$
	$\dot{p} = -\mathcal{H}_x^\top$
	$\mathcal{H}_u = 0$

1. Write out Hamiltonian  $\mathcal{H}$
2. Evaluate  $\mathcal{H}_u$  and  $\mathcal{H}_x$  to get costate equation and  $u^*$
3. Can put state and costate equations into matrix form and solve eigenvalue problem using the Hamiltonian matrix  $\bar{H}$ .  
Solution is:

$$\begin{bmatrix} x(t) \\ p(t) \end{bmatrix} = v_1 \alpha_1 e^{\lambda_1 t} + v_2 \alpha_2 e^{\lambda_2 t}$$

4. Or can propose a solution  $p = Px$  to get Riccati equation
5. If  $t_f = \infty$  then  $\dot{P} = 0$  in the Riccati equation

For LQR problem the Hamiltonian is

$$\mathcal{H} = \frac{1}{2}(x^\top R_{xx}x + u^\top R_{uu}u) + p^\top (Ax + Bu)$$

evaluate  $\mathcal{H}_x$  and  $\mathcal{H}_u$  we get

$$\mathcal{H}_x = x^\top R_{xx} + p^\top A$$

$$\mathcal{H}_u = u^\top R_{uu} + p^\top B$$

Plugging in

$$\dot{p} = -R_{xx}x - A^\top p$$

$$0 = R_{uu}u + B^\top p$$

Solving the second condition for  $u^*$

$$u^* = -R_{uu}^{-1}B^\top p$$

Plugging in control we get following differential equations in matrix form

$$\begin{bmatrix} \dot{x} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} A & -BR_{uu}^{-1}B^\top \\ -R_{xx} & -A^\top \end{bmatrix} \begin{bmatrix} x \\ p \end{bmatrix}$$

To solve the state and costate equations, guess a solution

$$p = Px$$

Differentiate and simplify to get Riccati equation.

$$\dot{P} = PA + A^\top P + R_{xx} - PBR_{uu}^{-1}B^\top P$$

When we have infinite time horizon,  $\dot{P} = 0$ . We need to solve this equation to find  $P$ , and then use  $p = Px$  with the  $P$  we found to determine the control law  $u$ . For scalar systems we can easily solve the Riccati equation by hand. We can also compute the solution  $P$  using the eigenvalues and eigenvectors of  $\bar{H}$ , although this approach doesn't always work.

**Constrained Optimization with Pontryagin**

$$\begin{aligned} \text{Minimize} \quad & J = h + \int_t^{t_f} g dt \\ \text{subject to} \quad & \dot{x} = f \\ & c(x, u, t) \leq 0 \end{aligned}$$

<b>Hamiltonian:</b>	$\mathcal{H} = g + p^\top f$
<b>Necessary Conditions:</b>	$\dot{x} = \mathcal{H}_p^\top = f$
	$\dot{p} = -\mathcal{H}_x^\top$
	$u^* = \arg \min_u \mathcal{H}$

1. Form Hamiltonian
2. Find  $u^*$  as it depends on sign of other stuff
  - Usually this stuff has  $p$  in it, so have to find costate equation
3. Plug each option for  $u^*$  into the state equations and integrate back to find expression for  $x(t)$ 
  - Will use initial conditions for constants of integration
  - Can see switching curve for each option by plugging in final condition and eliminating  $t_f$
4. More notes:  $h_x = p(t_f)$

**Solving the HJB Equation**

$$\begin{aligned} \text{Minimize} \quad & J = h + \int_t^{t_f} g dt \\ \text{subject to} \quad & \dot{x} = f \end{aligned}$$

Using dynamic programming in continuous time leads to the HJB equation.

<b>Hamiltonian:</b>	$\mathcal{H} = g + J_x^* f$
<b>HJB Equation:</b>	$-J_t^* = \min_u \mathcal{H}$
<b>Boundary condition:</b>	$J^*(t_f) = h$

1. Write out Hamiltonian  $\mathcal{H}$
2. Evaluate  $\mathcal{H}_u = 0$  and solve for  $u^*$  in terms of  $J_x^*$ 
  - Can set  $\mathcal{H}_u = 0$  since control is not constrained
3. Substitute  $u^*$  into HJB equation and simplify
4. Propose (guess) a solution form for  $J^* = px^2$
5. Plug solution into HJB
6. Solve for  $p(t)$ , which should give  $J_x^*$  and thus  $u^*$ 
  - Use boundary condition  $J^*(t_f) = h$

For singular arc must have  $\mathcal{H}_u = 0$  and  $\dot{\mathcal{H}}_u = 0$ . Remember in Goddard problem we dot  $\ddot{\mathcal{H}} = 0$  to constrain  $u$ !

lec 18: For min time problems if system is completely controllable by  $B_i$   $u_i$  can have no singular arcs. If not completely controllable  $u_i$  must have a singular interval.

For min time-fuel, if completely controllable by  $B_i$  and  $A$  nonsingular, there can be no singular intervals.

LQR vs KE  $P, Q; A, A^\top; B, C_y^\top; Q, R_{ww}; R, R_{vv}$

<b>Plant:</b>	$\dot{x} = Ax + B_u u + B_w w$ $y = Cx + v$
<b>Estimator:</b>	$\dot{\hat{x}} = A\hat{x} + B_u u + L(y - C\hat{x})$
<b>Error:</b>	$\dot{\tilde{x}} = (A - LC)\tilde{x} + B_w w - Lv$

$Q$  is the covariance matrix. Consider the following

$$\dot{x} = Ax + B_w w$$

$$E[\tilde{x}(t)\tilde{x}^\top(t)] = Q(t)$$

$$\begin{aligned}\dot{Q} &= E[\dot{\tilde{x}}\tilde{x}^\top] + E[\tilde{x}\dot{\tilde{x}}^\top] \\ &= E[(Ax + B_w w)x^\top] + E[x(Ax + B_w w)^\top] \\ &= E[Axx^\top + B_w wx^\top] + E[xx^\top A^\top + w^\top B_w^\top] \\ &= AQ + QA^\top + B_w R_{ww} B_w^\top\end{aligned}$$

This is the differential Lyapunov equation, which we substitute in for our terms in the estimation error equation and get:

$$\dot{Q} = (A - LC)Q + Q(A - LC)^\top + B_w R_{ww} B_w^\top + L R_{vv} L^\top$$

Now the goal is:

$$\text{minimize } J = E[\tilde{x}^\top(t_f) P_f \tilde{x}(t_f)]$$

subject to Differential Lyapunov equation

We can express  $J$  in terms of  $Q$  as

$$\begin{aligned}J &= \text{tr}E[P_f \tilde{x}(t_f) \tilde{x}^\top(t_f)] \\ &= \text{tr}P_f Q(t_f)\end{aligned}$$

So make  $Q(t)$  the state  $L$  the control and its like an optimal control problem. Define Hamiltonian

$$\mathcal{H} = \text{tr}[P\{(A - LC)Q + Q(A - LC)^\top + B_w R_{ww} B_w^\top + L R_{vv} L^\top\}]$$

Optimal control satisfies

$$\mathcal{H}_L = 0$$

$$\mathcal{H}_L = -PQC^\top + PLR_{vv}$$

$$L = QC^\top R_{vv}^{-1}$$

Plugging  $L$  back into differential Lyapunov equation, we get estimation Riccati equation.

$$\dot{Q} = AQ + QA^\top + B_w R_{ww} B_w^\top - QC^\top R_{vv}^{-1} CQ$$

## More Notes

when  $\rho \rightarrow \infty$  (expensive) closed-loop poles are stable OL poles and reflection of unstable OL poles.  $\rho \rightarrow 0$  control is cheap, so CL poles go to OL zeros (stable ones plus reflection of unstable) and remaining go to infinity. When control is cheap CL poles go to zeros and we minimize  $J$  by making things unobservable. Non-minimum phase zeros place fundamental limitation on cost.

If  $g$  and  $h$  not functions of time,  $\mathcal{H}$  is constant along optimal trajectory.

Singular arcs  $\mathcal{H}_u = 0$  and  $\dot{\mathcal{H}}_u = 0$

No cost along a singular arc.

To find optimal control for a continuous time system, can use Hamiltonian like always, write state and costate equations and put in matrix form. If it is infinite time, can solve for the the costate equation by  $P = V_{21} V_{11}^{-1} x$ . (see lec 11-3).

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} / \det$$

$$\frac{d}{dX} \text{tr}[AXB] = A^\top B^\top$$

If  $A$  is invertible, then

$$\det \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \det(A) \det(D - CA^{-1}B)$$

$A$  and  $B$  square  $\det(AB) = \det(A)\det(B)$ . For any matrices  $C$  and  $D$  such that  $CD$  is square,  $\det(I + CD) = \det(I + DC)$ .  
Symmetric root locus

$$\bar{\mathcal{H}} = \begin{bmatrix} A & -BR_{uu}^{-1}B^\top \\ -R_{xx} & -A^\top \end{bmatrix}$$

$$\det(sI - \bar{\mathcal{H}}) = (-1)^n \{D(s)D(-s) + \frac{R_{zz}}{R_{uu}} N(s)N(-s)\} = 0$$

where  $\frac{R_{zz}}{R_{uu}} = \frac{1}{\rho}$ ,  $G_{zu}(s) = C_z(sI - A)^{-1}B_u$  and  $D(s) = \det(sI - A)$ . Use the 180 degree root locus for  $n - m$  even, 0 degree root locus for  $n - m$  odd. 180 degree is the "normal" root locus.

To plot SRL given  $(A, B, C_z)$ . Write TF from  $u$  to  $z$ . Plot all poles and zeros. Reflect all poles and zeros across imaginary axis. Check  $n - m$  and draw asymptotes.

## LQR Margins

Loop transfer function from  $u$  to  $-u$

$$L(s) = K(sI_a)^{-1}$$

## Kalman frequency domain equality

$$[I + L(-s)]^\top [I + L(s)] = I + \frac{1}{\rho} G_{zu}^\top(-s) G_{zu}(s)$$

lec 14: For free final time problems  $h_t + H(t_f) = 0$

lec 15: Augmenting Hamiltonian with point constraint.  $J_a = h + \int \mathcal{H} - p^\top f dt + \pi^\top N$  we find  $p^\top(t_1^+) = p^\top(t_1^-) - \pi^\top N_x(t_1)$  and  $\mathcal{H}(t_1^+) = \mathcal{H}(t_1^-) + \pi^\top N_t(t_1)$ .

lec 17: WHEN  $t_f$  AND  $x(t_f)$  ARE FREE TRANSVERSALITY IS:

$$h_{t_f}(t_f) + \mathcal{H}(t_f) = 0$$

(this might be true even when  $x(t_f)$  is fixed)

## Constrained Optimization with Augmented Hamiltonian

$$\begin{aligned}\text{Minimize } & J = h + \int_{t_0}^{t_f} g dt \\ \text{subject to } & \dot{x} = f \\ & c(x, u, t) \leq 0\end{aligned}$$

$$\text{Hamiltonian: } \mathcal{H} = g + p^\top f$$

$$\text{Aug. Hamiltonian: } \mathcal{H}_a = \mathcal{H} + \nu^\top c$$

$$\text{Necessary Conditions: } \dot{x} = f$$

$$\dot{p} = -(\mathcal{H}_a)_x^\top$$

$$(\mathcal{H}_a)_u = 0$$

$$\nu_i \geq 0 \quad \text{if } c_i = 0 \quad (\text{active})$$

$$\nu_i = 0 \quad \text{if } c_i < 0 \quad (\text{inactive})$$

**General Stuff**

Taylor series expansion

$$f(x + \Delta x) = f(x) + g^\top(x)\Delta x + \frac{1}{2}\Delta x^\top H(x)\Delta x + \dots$$

where  $g(x) = \nabla f(x)$  is the gradient and  $H(x)$  is the Hessian

$$g(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \dots & \frac{\partial f}{\partial x_n} \end{bmatrix}^\top$$

$$H(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \dots & \frac{\partial^2 f}{\partial x_1 x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n x_1} & \dots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

**Stationary point**  $g(x^*) = 0$ . If  $H(x^*) > (\geq) 0$  then  $x^*$  is a **strong (weak) minimum**.

**Newton's Method**

Guess a stationary point  $x_k$ . Then can write actual stationary point as  $x = x_k + \Delta x_k$  and we have  $f(x) = f(x_k + \Delta x_k) \approx f(x_k) + g(x_k)^\top \Delta x_k + \frac{1}{2}\Delta x_k^\top H(x_k)\Delta x_k$ . and we have  $\nabla f(x) \approx g(x_k) + H(x_k)\Delta x_k$ . And we want this to be zero so set it to zero and get  $x_{k+1} = x_k - H_k^{-1}g_k$ .

**Convergence Rates****Constrained Minimization****Equality Constraints**

Given a problem in the form

$$\begin{aligned} \text{optimize} \quad & f(x, y) \\ \text{subject to} \quad & c(x, y) = 0 \end{aligned}$$

the following approach will find stationary points which satisfy the constraint.

**1. Form Lagrangian**

$$\text{Lagrangian:} \quad L(x, y, \lambda) = f(x, y) + \lambda^\top c(x, y)$$

**2. Form necessary conditions**

$$\frac{\partial L}{\partial x} = 0 \quad \frac{\partial L}{\partial y} = 0 \quad \frac{\partial L}{\partial \lambda} = 0$$

**3. Solve** and find the solution (if there are more than one) that corresponds to either min or max**Inequality Constraints**

Given a problem in the form

$$\begin{aligned} \text{optimize} \quad & f(x, y) \\ \text{subject to} \quad & c_1(x, y) \leq 0 \\ & c_2(x, y) \geq 0 \end{aligned}$$

Approach is similar to equality constraints. Only this time, we have different cases,  $\lambda_i = 0$  meaning the constraint is inactive, or  $\lambda_i \neq 0$  means the constraint is active.

**1. Rewrite the inequality constraints** so they are all in the form *less than zero*

$$c_i(x, y) \leq 0$$

**2. Form Lagrangian** as:  $L(x, y, \lambda) = f(x, y) + \lambda^\top c(x, y)$ **3. Form necessary conditions** (note change from before)

$$\frac{\partial L}{\partial x} = 0 \quad \frac{\partial L}{\partial y} = 0 \quad \lambda_i \frac{\partial L}{\partial \lambda_i} = 0$$

**4. Consider different cases** whether each constraint is active/inactive

- (a) **Both constraints inactive**  $\lambda_1 = \lambda_2 = 0$ . Plug this into equations, solve, verify the constraints are satisfied.  
 (b) **Inactive:**  $\lambda_1 = 0$ , **active:**  $\lambda_2 \neq 0$ . Same as the above.

- (c) **Active:**  $\lambda_1 \neq 0$ , **inactive:**  $\lambda_2 = 0$ . Same as the above.  
 (d) **Both active:**  $\lambda_1 \neq 0$ , and  $\lambda_2 \neq 0$ . No reduction of the equations that we have to solve, still have in this case solve four equations four unknowns.

**5. Look at each case above and evaluate**  $f(x, y)$  at each stationary point which satisfies the constraints to find the min/max.

*Note: according to KKT conditions, all active inequality constraints must have  $\lambda_i \geq 0$ , if not it is not a valid solution*

**Constraint Sensitivity**

To find the sensitivity of a cost to a constraint, write the constraint, say  $c_1(x, y) \leq 0$  as  $\bar{c}_1(x, y) \leq C_1$ . Then we calculate the change in cost as follows, where say the constraint  $C_1$  is changed to  $C_1 + \Delta C_1$ .

$$\Delta f = -\lambda_1 \Delta C_1$$

**Barrier Functions**

Augmented Lagrangians.

**Adjoint Methods**

Computer program  $y = f(x)$  where  $y$  is a scalar cost and  $x$  is a vector. We want to evaluate the gradient,  $g(x) = \frac{\partial y}{\partial x}$ . If in the computer program we calculate  $y$  by operating on  $x$  throughout the code as  $z_1 = F_1(x_1)$  then  $x_2 = F_2(z_1)$  and so on we have  $y = F_n(F_{n-1}(F_{n-2}(\dots x \dots)))$  and so to get the gradient it is lots of chain rules.  $\frac{\partial y}{\partial x} = \underbrace{\frac{\partial F_n}{\partial z_{n-1}}}_{1 \times m} \underbrace{\frac{\partial F_{n-1}}{\partial z_{n-2}}}_{m \times m} \dots \underbrace{\frac{\partial F_1}{\partial x}}_{m \times n}$ . Depending on

how we group these multiplications is either forward or reverse accumulation. Reverse is better because we the number of variables to do each operation is less ( $1 \times m$ ) instead of ( $m \times n$ ) with forward.

**Example 4-13**

$y = f(x_1, x_2) = \sin x_1 + x_1 x_2$ . Forward code: (1)  $t_1 = \sin x_1$ , (2)  $t_2 = x_1 x_2$ , (3)  $y = t_1 t_2$ . Reverse code: form  $z_{n-1} = F_n(z_{n-1})$ . (3)  $z_2 = [t_1 \ t_2 \ y]^\top$ ,  $F = [t_1 \ t_2 \ t_1 t_2]^\top$ . Evaluate  $T_3 = \frac{\partial F_3}{\partial z_2}$  where  $F$  is column  $z$  is row in making this derivative matrix. Then  $[t_1 \ t_2 \ y]_b = [t_1 \ t_2 \ y]_b T_3$ . Repeat this process for rest of code.

**Dynamic Programming****Discrete Time**

The problem statement is

$$\text{Minimize} \quad J = h(x_N, N) + \sum_{k=0}^{N-1} g_k(x_k, u_k)$$

$$\text{subject to} \quad x_{k+1} = f_k(x_k, u_k)$$

with  $x_0$  given and  $u \in \Omega$  feasible controls. We break this up by defining the **cost-to-go**, which is the cost to get to any point  $k$  to the end as

$$V_k(x) = \min_{u_i} \left\{ h(x_N, N) + \sum_{i=k}^{N-1} g_i(x_i, u_i) \right\}$$

We can break the cost-to-go up and get the **basic recursion of dynamic programming**

$$V_k(x) = \min_{u_k} \left\{ g_k(x_k, k) + V_{k+1}(f(x, u_k)) \right\}$$

**Discrete LQR**

Analytical solutions don't exist for many DP problems, but this is not the case for LQR.

$$\text{Minimize } J = \underbrace{\frac{1}{2} x_N^\top H_N x_N}_{h(x_N)} + \sum_{k=0}^{N-1} \underbrace{\frac{1}{2} x_k^\top Q_k x_k + u_k^\top R_k u_k}_{g_k(x_k, u_k)}$$

$$\text{subject to } x_{k+1} = A_k x_k + B_k u_k$$

where  $H > 0, Q \geq 0, R > 0$ . To solve use the basic recursion of dynamic programming.

1. Evaluate  $V_{N-1}$  using given  $h(x_N)$  and  $g_k(x_k, u_k)$ .
2. Plug in dynamics where  $x_N = A_{N-1} x_{N-1} + B_{N-1} u_{N-1}$ .
3. We want to find the control to minimize  $V_{N-1}$ , so differentiate the right hand side to get the following, which we will transpose and then set to zero

$$\frac{\partial V_{N-1}}{\partial u_{N-1}} = u_{N-1}^\top R_{N-1} + \{Ax + Bu\}_{N-1}^\top H B_{N-1} = 0$$

so doing algebra we get

$$u_{N-1}^* = - \underbrace{(R_{N-1} + B_{N-1}^\top H B_{N-1})^{-1} B_{N-1}^\top H A_{N-1}}_{F_{N-1}} x_{N-1}$$

4. Now plug optimal control  $u_{N-1}^*$  into  $V_{N-1}$  to get

$$V_{N-1} = \frac{1}{2} x_{N-1}^\top \underbrace{\{Q + F^\top R F + (A - B F)^\top H (A - B F)\}_{N-1}}_{P_{N-1}} x_{N-1}$$

5. Looking at the above expression, we now assume a solution  $V_k(x_k) = \frac{1}{2} x_k^\top P_k x_k$  and then find  $V_{k-1}(x_{k-1}) = \frac{1}{2} x_{k-1}^\top P_{k-1} x_{k-1}$ . Note that  $P_N = H$ .

$$P_{k-1} = Q_{k-1} + F_{k-1}^\top R_{k-1} F_{k-1} + \{A_{k-1} - B_{k-1} F_{k-1}\}^\top P_k \{A_{k-1} - B_{k-1} F_{k-1}\}$$

$$F_{k-1} = (R_{k-1} + B_{k-1}^\top P_k B_{k-1})^{-1} B_{k-1}^\top P_k A_{k-1}$$

6. Control is  $u_k = -F_k x_k$ .  $P_k$  and  $F_k$  are independent of the state and can be computed off-line. If we plug the expression for  $F_{k-1}$  into  $P_{k-1}$  we get discrete time Riccati equation.
7. If the system is LTI,  $A, B, Q, R$  are constants and so replace  $P$  with  $P_{ss}$  and solve Algebraic Riccati Equation.

$$P_{ss} = Q + A^\top \{P_{ss} - P_{ss} B (R + B^\top P_{ss} B)^{-1} B^\top P_{ss}\} A$$

**Continuous Time**

The problem statement is

$$\text{Minimize } J = h(x(t_f), t_f) + \int_t^{t_f} g(x(t), u(t), t) dt$$

$$\text{subject to } \dot{x} = f(x, u, t)$$

Instead of breaking the sum up in the cost to go, we break the integral up from  $t$  to  $t + \Delta t$  and  $t + \Delta t$  to  $t_f$ . We get

$$J^*(x(t), t) = \min_{t \leq \tau \leq t + \Delta t} \left\{ \int_t^{t+\Delta t} g(x, u, \tau) d\tau + J^*(x(t + \Delta t), t + \Delta t) \right\}$$

1. Represent the  $J^*(x(t + \Delta t), t + \Delta t)$  term using Taylor series.

$$J^*(x(t + \Delta t), t + \Delta t) \approx J^*(x(t), t) + J_x^*(x(t), t) \Delta t + J_x^*(x(t), t) \frac{dx}{dt} \Delta t$$

2. Must ensure first order terms match.

$$-J_t^*(x, t) = \min_u \mathcal{H}$$

where

$$\mathcal{H} = g(x, u, t) + J_x^*(x, t) f(x, u, t)$$

To solve CT LQR guess  $J^* = \frac{1}{2} x^\top(t) P(t) x(t)$ , form Hamiltonian, then minimize  $\mathcal{H}$  wrt to  $u$ . So take derivative and get  $u^*(t) = -R_{uu}^{-1}(t) B^\top(t) P(t) x(t)$ . Plug this control back into  $H$  and then to satisfy HJB we must have  $-J_t^* = -\frac{1}{2} x^\top(t) \dot{P}(t) x(t)$ , giving differential Riccati equation.

**Calculus of Variations****Derivation of Euler-Lagrange Equation**

We find increment defined as  $\Delta J = J(x + \delta x) - J(x)$ . Then simplify this increment to first order to get the variation, which by the FTocov says  $\delta J(x^*(t), \delta x(t)) = 0$ . Start with cost

$$J(x) = \int_{t_0}^{t_f} g(x, \dot{x}, t) dt$$

Evaluate perturbed cost

$$J(x + \delta x) = \int_{t_0 + \delta t_0}^{t_f + \delta t_f} g(x + \delta x, \dot{x} + \delta \dot{x}, t) dt$$

1. Split  $J(x + \delta x)$  integral into three pieces
2. Approximate the integrals using  $\int_t^{t+\delta t} f(\tau) d\tau = f(t) \delta t$
3. Approximate integrand using Taylor series

$$g(x + \delta x, \dot{x} + \delta \dot{x}, t) \approx g(x, \dot{x}, t) + \frac{\partial g}{\partial x} \delta x + \frac{\partial g}{\partial \dot{x}} \delta \dot{x}$$

4. Neglect terms higher than first order, and then subtract  $\delta J = J(x + \delta x) - J(x)$  and simplify
5. Integrate by parts

$$u = \frac{\partial g_a}{\partial \dot{x}} \quad du = \frac{d}{dt} \frac{\partial g_a}{\partial \dot{x}} \quad dv = \delta \dot{x} dt \quad v = \delta x$$

6. Simplify using following at  $t_0$  and  $t_f$

$$\delta x(t_0) \approx \delta x_0 - \dot{x}(t_0) \delta t_0$$

$$\delta J(x, \delta x) = \int_{t_0}^{t_f} \left\{ \frac{\partial g}{\partial x} - \frac{d}{dt} \frac{\partial g}{\partial \dot{x}} \right\} \delta x dt + \left( g(t_f) - \frac{\partial g}{\partial \dot{x}} \Big|_{t_f} \dot{x}(t_f) \right) \delta t_f - \left( g(t_0) - \frac{\partial g}{\partial \dot{x}} \Big|_{t_0} \dot{x}(t_0) \right) \delta t_0 + \frac{\partial g}{\partial \dot{x}} \Big|_{t_f} \delta x_f - \frac{\partial g}{\partial \dot{x}} \Big|_{t_0} \delta x_0$$

By setting this to zero for arbitrary  $\delta x$ , and by applying constraints e.g.  $t_0$  fixed,  $x(t_f)$  free, we get the necessary conditions.

**General Terminal Conditions**

If there are general terminal conditions  $m(x(t_f), t_f) = 0$  augment cost with them using Lagrange multiplier  $\nu$ . If cost is

$$J(x) = h(x(t_f), t_f) + \int_{t_0}^{t_f} g(x, \dot{x}, t) dt$$

augment as

$$J_a(x) = \underbrace{h(x(t_f), t_f) + \nu^\top m(x(t_f), t_f)}_{w(x(t_f), t_f)} + \int_{t_0}^{t_f} g(x, \dot{x}, t) dt$$

Do same thing as before to get necessary conditions:  $g_x - \frac{d}{dt} g_{\dot{x}} = 0$ ,  $(w_x + g_{\dot{x}})_{t_f} = 0$ ,  $(w_{t_f} + g - g_{\dot{x}} \dot{x})_{t_f} = 0$ , and  $m(x(t_f), t_f) = 0$ .

**Variational Approach to Optimal Control**

Augment cost using Lagrange multiplier  $p(t)$  as:

$$J_a = h(x(t_f), t_f) + \int_{t_0}^{t_f} \{g(x, u, t) + p^\top (f(x, u, t) - \dot{x})\} dt$$

$$\mathcal{H}(x, u, p, t) = g(x, u, t) + p^\top f(x, u, t)$$

$$w(x(t_f), \nu, t_f) = h(x(t_f), t_f) + \nu^\top m(x(t_f), t_f)$$

**Necessary Conditions:**

$$\dot{x} = \mathcal{H}_p^\top = f(x, u, t)$$

$$\dot{p} = -\mathcal{H}_x^\top$$

$$\mathcal{H}_u = 0$$

Boundary conditions: if  $t_f$  is not fixed use  $H(t_f) + w_{t_f}(t_f) = 0$ , otherwise don't use it. But use these regardless:  $m(x(t_f), t_f) = 0$ ,  $x(t_0), t_0$  given,  $p(t_f) = \left( \frac{\partial w}{\partial x} \Big|_{t_f} \right)^\top$