Linear Systems and Optimal Control
Condensed Notes

J. A. McMahan Jr.

November 12, 2010

Contents

1 Definitions **3**

1.1 Linear Systems 3

1.2 Optimization 4

2 Theorems and Formulas **6**

2.1 Static Optimization 6

2.1.1 Optimization Without Constraints 6

2.1.2 Optimization With Equality Constraints 7

2.2 Optimal Control Basics 10

2.2.1 Euler Equations 10

2.2.2 General Continuous-time and Discrete-time Optimization 11

2.3 Linear Quadratic Regulator and Related Problems 12

2.3.1 Linear Quadratic Regulator Optimal Feedback 12

2.3.2 LQR Suboptimal Feedback and the Algebraic Ricatti Equation 13

2.3.3 Algebraic Ricatti Equation Theorems 14

2.3.4 Ricatti Equation Analytic Solution 16

2.3.5 Linear Quadratic Tracker 16

2.4 Final-Time-Free Problems 18

2.4.1 Notes and Examples 18

2.4.2 Linear Quadratic Minimum Time Problem 18

2.5 Constrained Input Control 19

2.5.1 Bang-Bang Control 19

2.5.2 Bang-Off-Bang Control 19

2.5.3 Constrained Input Control LQR Problems 21

2.6 Numerical Techniques For Control Problems 21

Chapter 1

Definitions

1.1 Linear Systems

ordinary differential equation: An equation of the form $\dot{x} = f(x,t)$, where $x = x(t)$. Here $\dot{x}$ can be a vector and f can be vector valued so that this may be a system of ODE's.

domain: An open, nonempty, connected subset of $\mathbb{R}^{n+1}$. This is associated with a problem where $x \in \mathbb{R}^n$.

linear system: If for any two solutions ϕ_1 and ϕ_2 of an ODE, $c\phi_1 + \phi_2$ is also a solution, where c is a constant, the ODE is called linear. These can be written as $\dot{x} = A(t)x$, where $A(t)$ is a matrix-valued function.

autonomous system: An ODE of the form $\dot{x} = f(x)$ (i.e., no explicit time dependence) is called autonomous or time-invariant.

equivalent matrices: Let $A, B \in \mathbb{R}^{m\times n}$, $Q \in \mathbb{R}^m$, $P \in \mathbb{R}^n$, and Q and P both nonsingular. A and B are equivalent if $A = QBP$.

similar matrices: If A and B are square matrices that are equivalent in the form $A = P^{-1}BP$, then they are called similar.

characteristic polynomial: The characteristic polynomial of a matrix A, is the function defined by $p(x) = \det(A - xI)$.

minimal polynomial: The minimal polynomial of A is the polynomial $m(x)$ of least degree such that $m(A) = 0$.

companion form: The companion form of a matrix A is a matrix A_c which is similar to A and of the form

$$\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{bmatrix}.$$

Note that the a_i elements are also the coefficients of the characteristic polynomial of A. The companion form does not always exist.

nilpotent: A matrix is nilpotent is $N^q = 0$ for some integer q.

fundamental matrix: Let $\{\phi_1, \ldots, \phi_n\}$ be a set of n linearly independent solutions of an n^{th} order linear homogeneous ODE. A fundamental matrix of the system is

$$\Phi = \begin{bmatrix}\phi_1 & \phi_2 & \cdots & \phi_n\end{bmatrix}$$

state transition matrix: For any fundamental matrix Ψ of a linear homogeneous ODE, the state transition matrix is defined by

$$\Phi(t, t_0) = \Psi(t)\Psi^{-1}(t_0).$$

mode of a system:

zero-input response: The response of an ODE system with the input set to 0.

zero-state response: The response of an ODE system with the initial condition set to 0.

transfer function: The transfer function of the system

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du,\end{aligned}$$

is the function

$$H(s) = C\left(sI - A\right)^{-1}B + D.$$

1.2 Optimization

static optimization: Optimization of functions where time is not a parameter.

performance index: A scalar function, $L(x, u) \in \mathbb{R}$, which gives a cost or performance value for a given x (the state or auxiliary vector) and u (the control or decision vector).

critical point: A critical point (also called stationary point) of a function L is a point where the increment $dL = 0$ to first order for the increments of the variables which L depends on.

saddle point: A critical point which is not an extremum.

curvature matrix: The second derivative of a scalar function $L(u)$, L_{uu}. This is also called the Hessian matrix.

positive definite / semidefinite: A scalar function is positive definite if its range is positive and positive semidefinite if its range is nonnegative.

negative definite / semidefinite: A scalar function is negative definite if its range is negative and negative semidefinite if its range is nonpositive.

indefinite: A scalar function is indefinite if its range contains both positive and negative values.

level curve: The level curve of a scalar function corresponding to the value c is the set of points $\{u : L(u) = c\}$.

constraint equation: An equation where a vector-valued function of x and u, $f(x, u) \in \mathbb{R}^n$, is set to 0, i.e., $f(x, u) = 0$. In optimization with constraints, the problem to be solved is the minimization of a scalar performance index, $L(x, u)$, while simultaneously satisfying the constraint equation $f(x, u) = 0$.

Lagrange multiplier: A vector, $\lambda \in \mathbb{R}^n$, which arises in the derivation of necessary conditions for the optimization of a scalar performance index with equality constraints. It is an expression of the requirement that the derivatives of the scalar performance index and the constraint equation function form a singular linear system.

Hamiltonian: A function of the state, control, and Lagrange multipliers which gives a concise representation of the necessary conditions for an extremum with equality constraints. Given a scalar performance index, $L(x, u)$, and a constraint equation function, $f(x, u)$, the Hamiltonian is defined as

$$H(x, u, \lambda) = L(x, u) + \lambda^T (f(x, u).$$

This is for the general static optimization case. The discrete-time Hamiltonian for the general optimization problem is given later.

admissible region: For constrained input problems, this is the region which $u(t)$ is allowed to lie in.

sign function: The sign function, denoted $\mathrm{sgn}(x)$, is defined as

$$\mathrm{sgn}(x) = \begin{cases} -1, & x < 0, \\ \text{undetermined}, & x = 0, \\ 1, & x > 0. \end{cases}$$

If x is a vector, this definition applies component-wise.

dead-zone function: The dead-zone function, denoted $\mathrm{dez}(x)$, is defined as

$$\mathrm{dez}(x) = \begin{cases} -1, & x < -1, \\ \text{between } -1 \text{ and } 0, & x = -1, \\ 0, & -1 \leq x \leq 1, \\ \text{between 0 and 1}, & x = 1, \\ 1, & x > 0. \end{cases}$$

If x is a vector, this definition applies component-wise.

saturation function: The saturation function, denoted $\mathrm{sat}(x)$, is defined as

$$\mathrm{sat}(x) = \begin{cases} -1, & x < -1, \\ x, & |x| \leq 1, \\ 1, & x > 0. \end{cases}$$

If x is a vector, this definition applies component-wise.

singular condition: This refers to linear minimum-time problems with constrained control inputs, not to matrices, linear functions, etc., although the idea is basically the same. A singular condition is when the optimal control is not well-defined over a finite interval by the sgn function (i.e., when the switching function is 0 on a finite interval).

normal problem: A linear minimum-time problem which involves no singular conditions.

Chapter 2

Theorems and Formulas

2.1 Static Optimization

2.1.1 Optimization Without Constraints

We want to find an extremum of an unconstrained performance function. Usually we are looking for a minimum, although there's no conceptual difference in finding a maximum.

Assumptions

- Local extrema exist.
- If L is continuous on a compact set Ω, then L attains a maximum and minimum on this set, so this may often be assumed.
- $L(u) \in \mathbb{R}$ where $u \in \mathbb{R}^m$ is a scalar performance index.

Conclusions

A necessary condition for an extremum at u is for u to be a critical point of L, i.e.

$$L_u = 0.$$

The following list characterizes a critical point:

- If $L_{uu} > 0$ (i.e., the Hessian matrix of L is positive definite), then the critical point is a local minimum.
- If $L_{uu} < 0$ (i.e., the Hessian matrix of L is negative definite), then the critical point is a local maximum.
- If L_{uu} is indefinite, then the critical point is a saddle point.
- If L_{uu} is semidefinite, then higher order terms of the Taylor expansion for an increment in L must be examined to determine what type of critical point we have.

Facts

- A matrix A is positive definite if all of its eigenvalues are positive. It is positive semidefinite if all of its eigenvalues are nonnegative.
- A matrix A is negative definite if all of its eigenvalues are negative. It is negative semidefinite if all of its eigenvalues are nonpositive.
- A matrix A is indefinite if some of its eigenvalues are positive, some of its eigenvalues are negative, and none of its eigenvalues are zero.
- The gradient L_u is always perpendicular to the level curves of L and points in the direction of increasing $L(u)$.

Proof Notes

Write out the Taylor series for an increment in L,

$$dL = L_u^T \, du + \frac{1}{2} du^T \, L_{uu} \, du + O(\|u\|^3).$$

The above follows fairly easily from analyzing this.

2.1.2 Optimization With Equality Constraints

We want to select a control u to optimize a scalar performance index $L(x, u)$ while simultaneously satisfying a constraint equation $f(x, u) = 0$. We will generally look for the minimum value, although finding the maximum value should be similar.

Assumptions

- $x, u \in \mathbb{R}^n$.
- $L(x,u) \in \mathbb{R}$ is a scalar performance index.
- $f(x,u) = 0$ where $f \in \mathbb{R}^n$ is a constraint equation.

Conclusions

- **Necessary conditions for an extremum in terms of L and f**

$$L_u - f_u^T f_x^{-T} L_x = 0.$$

- **Necessary conditions for an extremum in terms of the Hamiltonian**

$$\begin{aligned} H(x,u,\lambda) &= L(x,u) + \lambda^T f(x,u), \\ H_\lambda &= f = 0, \\ H_x &= L_x + f_x^T \lambda = 0, \\ H_u &= L_u + f_u^T \lambda = 0. \end{aligned}$$

- **Sufficient conditions for a minimum**

  The curvature matrix with constant f equal to zero must be positive definite. That is

$$L_{uu}^f = \begin{bmatrix} -f_u^T f_x^{-T} & I \end{bmatrix} \begin{bmatrix} H_{xx} & H_{xu} \\ H_{ux} & H_{uu} \end{bmatrix} \begin{bmatrix} -f_x^{-1} f_u \\ I \end{bmatrix}$$

  must be positive definite. If this is negative definite, it is a sufficient condition for a maximum. If it is indefinite, it is a sufficient condition for a saddle point.

- **Changes in constraints**

  Remaining at an optimal solution we want the dx, du, and dL as functions of df. The expressions for this are

$$\begin{aligned} C &= (L_{uu}^f)^{-1}(H_{ux} - f_u^T f_x^{-T} H_{xx}) f_x^{-1} \\ du &= -C\, df, \\ dx &= f_x^{-1}(I + f_u C)\, df, \\ dL &= -\lambda^T\, df + \frac{1}{2}\, df^T\, (f_x^{-T} H_{xx} f_x^{-1} - C^T L_{uu}^f C)\, df + O(\|df\|^3) \end{aligned}$$

  where L_{uu}^f is the curvature matrix of L with constant f equal to zero.

Facts

- Set $du = 0$. Then $L_f = -\lambda$. That is, the Lagrange multiplier is the partial derivative of L with respect to the constraint while holding u constant.
- The Lagrange multiplier indicates the rate of change of the optimal value of the performance index with respect to the constraint.

Proof Notes

- **Necessary Conditions for an Extremum**

  (First Way)

  To first order, dL is equal to zero for all increments du when df is zero. Write out first order Taylor expansions for dL and df. Note that f_x is nonsingular (from implicit function theorem?). Use to eliminate dx and get dL in terms of du. The condition follows.

  (Second Way)

  Write the first order expansions as a matrix equation

  $$\begin{bmatrix} dL \\ df \end{bmatrix} = \begin{bmatrix} L_x^T & L_u^T \\ f_x & f_u \end{bmatrix} \begin{bmatrix} dx \\ du \end{bmatrix} = 0.$$

  This implies the coefficient matrix is singular. So write

  $$\begin{bmatrix} 1 & \lambda^T \end{bmatrix} \begin{bmatrix} L_x^T & L_u^T \\ f_x & f_u \end{bmatrix}$$

  where λ is a Lagrange multiplier. Eliminate λ to get the expression

  (Third Way)

  Write out the increment of the Hamiltonian function to first order in terms of increments of x, u, and λ. Choose u and require $H_\lambda = 0$. Choose λ so that $H_x = 0$. This leaves $dL = dH = H_u^T\, du$ which requires $H_u = 0$.

- **Sufficient Conditions for a Minimum**

  Examine the second order term of the Taylor expansion of the increment of the Hamiltonian. Manipulate the results and reason similar to the unconstrained case that the resulting matrix is positive definite.

- **Changes in Constraints**

  Perform algebraic manipulations on the changes in the partial derivatives of H in terms of the increments of dx, du, and $d\lambda$.

2.2 Optimal Control Basics

2.2.1 Euler Equations

Assumptions

- Performance index in the form $J(x) = \int_a^b L(x(t), \dot{x}(t), t)\, dt$
- L is smooth enough to satisfy the equations below.

Conclusions

- For a fixed end-point, a necessary condition for an extremum x^* in the cost J, is that x^* satisfies the boundary value problem

$$L_x - \frac{dL_{\dot{x}}}{dt} = 0, \quad x(a) = x_a, \quad x(b) = x_b$$

- For a free end-point, a necessary condition for an extremum x^* in the cost J, is that x^* satisfies the boundary value problem

$$L_x - \frac{dL_{\dot{x}}}{dt} = 0, \quad x(a) = x_a, \quad L_{\dot{x}}(x^*, \dot{x}^*, t)\big|_{t=b} = 0$$

Facts and Proof Notes

- When applying the equation above, remember that taking the derivative with respect to $\dot{x}$ works the same as if we relabel with $v = \dot{x}$ and take the derivative with respect to v. Make sure the time derivative is applied to x and $\dot{x}$ implicitly.
- This can be proved with some techniques from the calculus of variations. Assume x^* is the minimum and compute the Taylor expansion of J at x^* evaluated at some perturbation of x^*, like $x^* + \delta x$. $J(x^* + \delta x) - J(x^*) \geq 0$ for a minimum. Discard any nonlinear parts of the Taylor series. Use integration by parts and simplify to get the above equations.

2.2.2 General Continuous-time and Discrete-time Optimization

Necessary Conditions for Performance Index Minimum

	Discrete	**Continuous**
System Model:	$x_{k+1} = f^k(x_k, u_k), \quad k \geq k_0$	$\dot{x} = f(x, u, t), \quad t \geq t_0$
Performance Index:	$J_i = \phi(N, x_N) + \sum_{k=i}^{N-1} L^k(x_k, u_k)$	$J(t_0) = \phi(x(T), T) + \int_{t_0}^{T} L(x, u, t)\, dt$
Final-State Constraint:	(continuous only)	$\psi(x(T), T) = 0$
Hamiltonian:	$H^k = L^k + \lambda_{k+1}^T f^k$	$H(x, u, t) = L(x, u, t) + \lambda^T f(x, u, t)$
State Equation:	$H^k_{\lambda_{k+1}} = x_{k+1} = f^k(x_k, u_k)$	$H_\lambda = \dot{x} = f(x, u, t)$
Costate Equation:	$H^k_{x_k} = \lambda_k = (f^k_{x_k})^T \lambda_{k+1} + L^k_{x_k}$	$H_x = -\dot{\lambda} = (f_x)^T \lambda + L_x$
Stationarity Equation:	$H^k_{u_k} = 0 = (f^k_{u_k})^T \lambda_{k+1} + L^k_{u_k}$	$H_u = 0 = (f_u)^T \lambda + L_u$

Boundary Conditions

Discrete:

$$\left(L^i_{x_i} + \left(f^i_{x_i}\right)^T \lambda_{i+1}\right)^T dx_i = 0,$$

$$(\phi_{x_N} - \lambda_N)^T dx_N = 0$$

Continuous:

$$\lambda^T\big|_{t_0} dx(t_0) - H\big|_{t_0} dt_0 = 0$$

$$\left(\phi_x + \psi_x^T \nu - \lambda\right)^T\Big|_T dx(T) + \left(\phi_t + \psi_t^T \nu + H\right)\big|_T dT = 0$$

Facts and Proof Notes

- For the discrete case, construct an augmented performance index by appending the state constraint with the Lagrange multiplier to the given performance index. Simplify the notation by inserting the Hamiltonian function defined above. Compute the increment of this augmented performance index and set it to zero, as the theory from static optimization says this is a necessary condition for an extremum. This allows you to reason the various expressions making up the increment must be zero, which gives the above expressions.

- For the continuous case, you need to use some ideas from the calculus of variations which I'll fill in later.

- The boundary condition involving the initial condition for the continuous case may be off. Usually the initial condition is given so don't worry too much about it.

2.3 Linear Quadratic Regulator and Related Problems

The linear quadratic regulator (LQR) defines an important class of control problems which involve linear dynamics and a quadratic cost. They are analytical results available for these problems which make them fairly well-understood.

2.3.1 Linear Quadratic Regulator Optimal Feedback

LQR Problem Statement

	Discrete	**Continuous**
State Equation:	$x_{k+1} = A_k x_k + B_k u_k$	$x(t) = A(t)x + B(t)y$
Performance Index:	$J_i = \frac{1}{2} x_N^T P x_N$	$J(t_0) = \frac{1}{2} x^T(T) P x(T)$
	$+ \frac{1}{2} \sum_{k=1}^{N-1} x_k^T Q_k x_k + u_k^T R_k u_k$	$+ \frac{1}{2} \int_{t_0}^{T} \left(x^T Q(t) x + u^T R(t) u \right) \, dt$

Ricatti Equations

Discrete:

$$S_k = A_k^T \left(S_{k+1} - S_{k+1} B_k \left(B_k^T S_{k+1} B_k + R_k \right)^{-1} B_k^T S_{k+1} \right) A_k + Q_k, \quad S_N = P$$

Continuous:

$$-\dot{S} = A^T S + SA - SBR^{-1}B^T S + Q, \quad S(T) = P$$

Optimal Feedback

	Discrete	**Continuous**
Kalman Gain:	$K_k =$	$K(t) = R^{-1}(t) B^T(t) S(t)$
	$(B_k^T S_{k+1} B_k + R_k)^{-1} B_k^T S_{k+1} A_k$	
Optimal Feedback:	$u_k = -K_k x_k$	$u(t) = -K(t)x(t)$
Optimal Cost To Go:	$J_i = \frac{1}{2} x_i^T S_i x_i$	$J(t_0) = \frac{1}{2} x^T(t_0) S(t_0) x(t_0)$

Restrictions

A, B, Q, R, P, S linear
$R > 0$, symmetric
$Q \geq 0$, $P \geq 0$, $S \geq 0$, all symmetric

Proof and Usage Notes

- For open loop control of an LQR system, just plug the equations into the general optimization formulas and solve.
- For closed loop control, we will have P given. Solve the Ricatti equation backwards in time using $S(T) = P$ as the end condition. Plug the result into the formula for the Kalman gain and then into the formula for the optimal feedback control.
- A technique that often works for deriving the Ricatti equations and related expressions is to assume a form for λ which matches what the boundary conditions give you. Plugging this into the necessary conditions often allows you to eliminate λ and find an expression in terms of given parameters in the problem.
- The cost-to-go is the cost starting at time t_i or t_0 to reach the desired state when using the optimal feedback control. So this is the optimal cost for a given initial time and initial state. This can be derived by plugging the optimal control expression into the cost and manipulating algebraically.

2.3.2 LQR Suboptimal Feedback and the Algebraic Ricatti Equation

Algebraic Ricatti Equation (ARE)

Discrete: $$S_\infty = A^T \left(S_\infty - S_\infty B \left(B^T S_\infty B + R \right)^{-1} B^T S_\infty \right) A + Q,$$

Continuous: $$0 = A^T S_\infty + S_\infty A - S_\infty B R^{-1} B^T S_\infty + Q,$$

Suboptimal Feedback

	Discrete	**Continuous**
Kalman Gain:	$K_\infty = \left(B^T S_\infty B + R \right)^{-1} B^T S_\infty A$	$K_\infty = R^{-1} B S_\infty$
Suboptimal Feedback:	$u_k = -K_\infty x_k$	$u(t) = -K_\infty x(t)$

Restrictions

A, B, Q, R time invariant
S_∞ must exist (check later theorems for some existence criteria)

Facts and Usage Notes

- The feedback works in the same way as the optimal feedback, only you solve the algebraic Ricatti equations rather than the differential Ricatti equations.
- The solutions to the algebraic Ricatti equations do not need to be unique, symmetric, positive definite, real, or finite, so this is not always an option. Some theorems are given later on to identify situations where they do exist.
- The solution to the algebraic Ricatti equation is an equilibrium point for the Ricatti difference / differential equaion. The particular solution we want to use in the suboptimal feedback is a limit of the Ricatti equation as time goes to infinity, so the suboptimal feedback can be thought of as the optimal feedback on infinite intervals. As such, it works better when it is used over longer time intervals.

2.3.3 Algebraic Ricatti Equation Theorems

These theorems give some criteria for the existence of useful solutions to the algebraic Ricatti equations. They are identical for the discrete and continuous cases.

Assumptions

- The system is an LQR system.
- (A, B) is stabilizable.

Conclusions

- There exists $S_\infty = \lim_{t\to-\infty} S(t) \quad \left(\lim_{k\to-\infty} S_k\right)$, a bounded solution to the Ricatti equaion, for every $S(T)$ (S_N).
- S_∞ is a positive definite solution to the algebraic Ricatti equation.

Proof Notes

Stabilizability implies the existence of a feedback which keeps the state bounded and drives the state to zero as time goes to ∞ and implies a finite cost as $t \to \infty$ $(k \to \infty)$. The optimal cost is bounded above by the cost of this feedback system. This can be used to show the solution to the Ricatti equation associated with the optimal cost system is bounded for all time. Symmetry at all points in time implies symmetry of the solution, and similar conclusions are drawn for the positive definiteness of the solution.

Assumptions

- The system is an LQR system.
- $Q = C^T C \geq 0, \quad R > 0.$
- $(A, C) = (A, \sqrt{Q})$ is observable.

Equivalent Statements

- (A, B) is stabilizable.
- 1. There exists a unique positive definite limiting solution, S_∞ to the Ricatti equation which is also the unique positive definite solution to the algebraic Ricatti equation.
  2. The closed-loop system

$$\text{(Continuous)} \quad \dot{x} = (A - BK_\infty)x$$

$$\text{(Discrete)} \quad x_{k+1} = (A - BK_\infty)x_k$$

is asymptotically stable.

Facts and Proof Notes

- To be clear, items 1. and 2. above are equivalent. These are each equivalent to the stabilizability of $(A, \sqrt{Q})$.
- One direction of the proof is easy, as the asymptotic stability of the closed loop plant implies (A, B) is stabilizable. The other direction involves using the optimal cost equation to show the observability of $(A, \sqrt{Q})$ requires the asymptotic stability of the state.
- This result is the same as the previous theorem with the observability condition added in.

2.3.4 Ricatti Equation Analytic Solution

	Discrete	**Continuous**
Hamiltonian Matrix:	$H = \begin{bmatrix} A^{-1} & A^{-1}BR^{-1}B^T \\ QA^{-1} & A^T + QA^{-1}BR^{-1}B^T \end{bmatrix}$	$H = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix}$
System:	$\begin{bmatrix} x_k \\ \lambda_k \end{bmatrix} = H \begin{bmatrix} x_{k+1} \\ \lambda_{k+1} \end{bmatrix}$	$\begin{bmatrix} \dot{x} \\ \dot{\lambda} \end{bmatrix} = H \begin{bmatrix} x \\ \lambda \end{bmatrix}$
H **Eigenvalue Property:**	$\mu \in \sigma(H) \Rightarrow \frac{1}{\mu} \in \sigma(H)$	$\mu \in \sigma(H) \Rightarrow -\mu \in \sigma(H)$
Eigenvalue Matrix:	$D = \begin{bmatrix} M & 0 \\ 0 & M^{-1} \end{bmatrix}$	$D = \begin{bmatrix} -M & 0 \\ 0 & M \end{bmatrix}$

Take $W = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}$ such that $W^{-1}HW = D$

Solution:	$S_\infty = W_{21}W_{11}^{-1}$	
Restrictions:	A nonsingular	None

Facts and Proof Notes

- The W given above is the matrix of eigenvectors corresponding to the eigvenvalues in matrix D, the diagonal matrix of eigenvalues of H.
- The continuous case constructs the ARE solution from the stable eigenvectors of H, while the discrete case uses the unstable eigenvectors of H (or stable eigenvectors of H^{-1}.
- The discrete matrix defined above seems to be backwards in time, so see if you can define it forwards in time. This may make it match with the continuous case better.
- Proving the eigenvalue property above is relatively straightforward by using the properties of H as a symplectic matrix.
- The rest is shown by writing out the above matrices, multiplying, manipulating algebraically, and using the asymptotic stability of the Ricatti equation to the algebraic Ricatti equation in certain cases to eliminate some terms.
- For the discrete case, there is a somewhat more complicated result for A singular, which I'll add later. It involves a generalized eigenvalue problem.

2.3.5 Linear Quadratic Tracker

The LQR tracking problem is a variation on the LQR problem which provides a way to track a reference trajectory for a linear system model.

Performance Index

Discrete:

$$J_i = \frac{1}{2}\left(Cx_N - r_N\right)^T P \left(Cx_N - r_N\right)$$

$$+ \frac{1}{2}\sum_{k=i}^{N-1}\Big(\left(Cx_k - r_k\right)^T Q \left(Cx_k - r_k\right) + u_k^T R u_k\Big)$$

Continuous:

$$J(t_0) = \frac{1}{2}\left(Cx(T) - r(T)\right)^T P \left(Cx(T) - r(T)\right)$$

$$+ \frac{1}{2}\int_{t_0}^{T}\Big(\left(Cx - r\right)^T Q \left(Cx - r\right) + u^T R u\Big)\, dt$$

Optimal Affine Control

Discrete:

$$K_k = \left(B^T S_{k+1} B + R\right)^{-1} B^T S_{k+1} A, \quad S_N = C^T P C$$

$$S_k = A^T S_{k+1}\left(A - BK_k\right) + C^T Q C$$

$$v_k = \left(A - BK_k\right)^T v_{k+1} + C^T Q r_k, \qquad v_N = C^T P r_N$$

$$K_k^v = \left(B^T S_{k+1} B + R\right)^{-1} B^T$$

$$u_k = -K_k x_k + K_k^v v_{k+1}$$

Continuous:

$$K(t) = R^{-1} B S(t), \qquad S(T) = C^T P C$$

$$-\dot{S} = A^T S + SA - SBR^{-1}B^T S + C^T Q C$$

$$-\dot{v} = \left(A - BK\right)^T v + C^T Q r, \qquad v(T) = C^T P r(T)$$

$$K^v(t) = R^{-1} B^T v(t)$$

$$u(t) = -K(t)x(t) + K^v(t)v(t)$$

Restrictions

$R > 0$, symmetric
$P \geq 0$, $Q \geq 0$, symmetric

- As in other situations involving the Ricatti equations, the form of the final solution can be derived by assuming the form for λ which matches the boundary conditions and eliminating λ by plugging this into the necessary conditions and manipulating. So you would set $\lambda(t) = S(t)x(t) - v(t)$ and plug in.
- The optimal cost-to-go can be computed in a similar way as in the LQR problem, with some extra terms added.

2.4 Final-Time-Free Problems

When the final time, T for the target state, $x(T)$, is not given in a problem, then dT is not necessarily 0. This must be taken into account when setting up the necessary boundary conditions for a minimum.

2.4.1 Notes and Examples

- Find a control which moves the state from x_0 to x_T in the minimum time. This will have $J(t_0) = \int_{t_0}^{T} 1\, dt$, or something similar.
- Problems with $x(T)$ and T both free and independent of each other. This will require the corresponding terms in the boundary conditions to each be set to 0.
- Problems with $x(T)$ and T both free, but dependent. For example, $x(t)$ may need to intersect with some given function $p(t)$. This allows you to write $dx(T) = \frac{dp(T)}{dT} dT$, and combine terms in the boundary conditions.
- Problems where T is free and $x(T)$ is required to be on a given fixed set, which does not change with time. This is a generalization of the second example.
- The conditions on the final costate in the last three examples are called the **transversality condition**.

2.4.2 Linear Quadratic Minimum Time Problem

The LQR problem can be extended to incorporate the time involved in reaching the final state by adding a constant term, ρ, to the cost functional. This gives

$$J = \frac{1}{2} x^T(T) P x(T) + \frac{1}{2} \int_{t_0}^{T} \left(\rho + x^T Q x + u^T R u\right)\, dt.$$

Note that no final state is specified in this example. By working through the necessary conditions for an optimal solution, we get the same conditions for the LQR case, except that we have $H(t) = 0$ for all t. Using $\lambda = Sx$ and $u = R^{-1} B^T S x$ and plugging this into the expression for $H(t_0)$, we find

$$x^T(t_0) \dot{S} x(t_0) = \rho.$$

The optimal control and Kalman gain will be the same as in the standard LQR case, so to solve this problem, we can integrate the Ricatti equation backwards from some final time, T_f, until we find a t such that $x^T(t_0) \dot{S} x(t_0) = \rho$ holds. The minimum time interval will then be $T_f - t$.

2.5 Constrained Input Control

Constrained input control problems provide some admissible region which the control, $u(t)$, must lie in. For these problems, we must replace the stationarity condition, $H_u = 0$, with the more general condition called **Pontryagin's Minimum Principle**.

Assumptions

- The system is given using the same notation as in the general optimization section.
- An admissible region for $u(t)$ is given.

Conclusions

Necessary conditions for a minimal-cost control are the same as in the general case, with the stationarity condition replaced by

$$H(x^*, u^*, \lambda^*, t) \leq H(x^*, u^* + \delta u, \lambda^*, t)$$

for all admissible δu. The asterisks denote optimal values.

This is sort of common sense, so there should be no need to talk about the proof.

2.5.1 Bang-Bang Control

Bang-bang control arises in linear minimum-time problems with constrained input magnitude. The resulting optimal control for these problems needs only take on two values, which are the extreme values of the control.

Optimal Control of Linear Minimum-Time Problem

System Model:	$\dot{x} = Ax + Bu$
Performance Index:	$J(t_0) = \int_{t_0}^{T} dt$
Constraints:	$\lvert u(t) \rvert \leq 1$
Optimal Control:	$u^*(t) = -\text{sgn}\left(B^T \lambda(t)\right)$

Proof and Usage Notes

- The proof of this follows from applying Pontryagin's minimum principle to get $(\lambda^*)^T B u^* \leq (\lambda^*)^T B u$. The left side is minimized by choosing u to be as small as possible, which depends on the sign of $(\lambda^*)^T B$.
- Note that the above may not define a unique control if $B^T \lambda$ is 0 over a finite interval.

2.5.2 Bang-Off-Bang Control

Bang-off-bang control arises in linear minimum-fuel problems with constrained input magnitude.

Optimal Control of Linear Minimum-Fuel Problems

System Model: $\dot{x} = Ax + Bu$

Performance Index: $J(t_0) = \int_{t_0}^{T} C^T \, |u(t)| \; dt$

Constraints: $|u(t)| \leq 1$

Optimal Control: $u_i^*(t) = -\text{dez}\left(\frac{b_i^T \lambda(t)}{c_i}\right)$

Proof and Usage Notes

- The proof of this follows from applying Pontryagin's minimum principle in a similar way as in the previous section. You will get a vector expression that allows you to consider each component separately, giving $c_i|u_i^*| + (\lambda^*)^T b_i u_i^* \leq c_i|u_i| + (\lambda^*)^T b_i u_i$. You can split this into cases to eliminate the absolute value symbol. From these expressions, you arrive at the necessary conditions for the minimum by making the terms on the left-side as small as possible, resulting in the given control
- Note that the above may not define a unique control if $B^T\lambda$ is 1 or -1 over a finite interval.

Equivalent Statements

- The system is normal.
- $U_i = \begin{bmatrix} b_i & Ab_i & \cdots & A^{n-1}b_i \end{bmatrix}$ has full rank for each $i = 1, 2, \ldots, m$.
- The system is reachable by each separate component of u.

Assumptions

- $x(T) = 0$
- The system has no eigenvalues with positive real parts.

Conclusions

- A minimum-time control exists.

Assumptions

- $x(T)$ is fixed.
- A minimum-time control exists.

Conclusions

- The minimum-time control is unique.

Assumptions

- The n eigenvalues of the system are all real.
- The minimum-time control exists.

Conclusions

- Each component $u_i(t)$ of the minimum-time control can switch no more than $n-1$ times.

2.5.3 Constrained Input Control LQR Problems

These are minimum-energy problems which have constraints on the control. The problem statement is the same as for the LQR case, with the addition of bounds on the control input. If the R in the cost functional is diagonal, the optimal control is given by

$$u_i^*(t) = -\mathrm{sat}\left([R^{-1}B^T\lambda]_i\right).$$

It may be possible to use this when R is merely diagonalizable, but as the diagonalization process skews the space, this must be done carefully. The proof of this follows from Pontryagin's principle, as in the previous sections.

2.6 Numerical Techniques For Control Problems

It is important to know the properties of the basic numerical analysis techniques when dealing with control problems, as most practical problems will require numerical techniques to solve. Particularly important are the properties of differential equation solvers. Some things that might be good to know about are

- Splines.
- One-step methods (simpler, easier to get started, may be more work).
- Multi-step methods (more complicated, can be faster, can have more problems with stability).
- Stiff problems.

Here are a couple of techniques for solving optimal control problems numerically.

Control Parameterization

Given a system $\dot{x} = f(x, u, t)$, parameterize your control input by a finite number of parameters. For example, sample points, or some sort of parameters to make up a spline. Write a function that gives the cost in terms of these parameters. Use a nonlinear programming optimization software with this function to find the minimum in terms of the parameterization you provided.

Direct Transcription

Here you discretize everything, not just the control. The constraints are the discretization equations (for example, whatever numerical method you are using). You run this through nonlinear programming optimization software and see what it gives you.