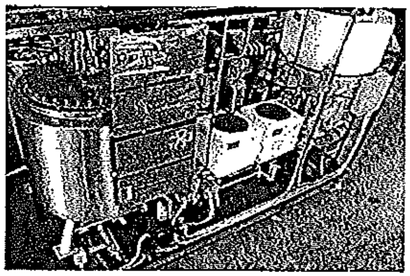
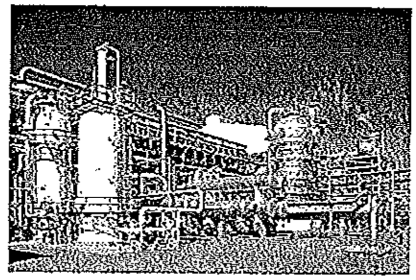
CHE 656 / IE 690
Advanced Process Control and Optimization

Spring 2011
(in collaboration with Honeywell Process Solutions)





Short course description: Topics in linear and nonlinear system theory applied to automatic control of processes. Subjects include stability analyses, phase plane methods, statistical disturbances, sampled systems, theoretical and experimental determination of process dynamics, numerical methods for optimization, and computer control. Examples drawn from the chemical process control industry, biological processes, and atomic/molecular dynamics.

Background: Economic, environmental and fundamental scientific incentives are increasing the demand for optimal control of chemical and biological processes, and a collaboration between Honeywell and Purdue aims to train the next generation of chemical and industrial engineers with fundamental skills in this area relevant to industry practice. More detailed treatment of several topics covered in CHE 456, with emphasis on application to realistic nonlinear dynamical processes. Suitable for undergraduates who have taken CHE 456.

Course Policy/Grading: No exams; 5 problem sets (60% of grade) + software term project (40% of grade)

Textbooks: Seborg, Process Dynamics and Control, 3rd ed. (required); Stengel, Optimal Control and Estimation (recommended)

3.000 Credit hours

Levels: Graduate, Professional, Undergraduate

Schedule Types: Lecture

Offered By : School of Chemical Engineering
Departments: Chemical Engineering, Industrial Engineering

Professor Raj Chakrabarti

Email: rchakra@purdue.edu

Room #: FRNY G124

Time: 12:30-1:20 MWF

CHE 656 / IE XXX
Advanced Process Control and Optimization
Spring 2011
(in collaboration with Honeywell Process Solutions)

Topics in linear and nonlinear system theory applied to automatic control of processes. Subjects include stability analyses, phase plane methods, statistical disturbances, sampled systems, theoretical and experimental determination of process dynamics, optimization, and computer control.

3.000 Credit hours

Levels: Graduate, Professional, Undergraduate
Schedule Types: Lecture

Offered By: School of Chemical Engineering
Departments: Chemical Engineering, Industrial Engineering

Course Policy/Grading: No exams; 5 problem sets (60 % of grade) + software term project (40% of grade)

Textbooks: Seborg, Process Dynamics and Control, 3rd ed. (required); Stengel, Optimal Control and Estimation (recommended)

Topics covered:

<u>Matrix methods</u>
- SVD, Gramians

<u>Optimal control theory</u>
- Nonlinear control methods and applications

<u>Linear multivariable control</u>
- State space analysis
- Setpts and local linearization
- Stability analysis: Lyapunov eqns
- Linear quadratic regulator (LQR)
  - MIMO control systems
  - Kalman gain; closed loop stability
- PID control within the LQR framework

Choosing cost function weights/pole placement

Controllability, observability, stabilizability - on stabilizability and pole placement note difference in interpretation for transfer fn vs closed loop fundamental matrix

Multivariable frequency domain analysis + transfer fns; modes of motion connection to lyapunov theory

PID pairing: input-output pairs in MIMO PID systems; choose simplest pairings that allow appl of SISO methods to MIMO

Controller tuning:

PID optimal tuning
Gain scheduling (state space model-based; extended linearization of nonlinear sys provides parameteric linear system / then apply pole placement - provide desired closed loop poles)
Initial gain diagonal
Applications of LP to tuning (generally are nonlinear input/output constraints)

Numerical methods for optimization:

Steepest Descent, Conjugate Gradient, Quasi-Newton
Monte Carlo algs (MCSA)
add: constrained opt (LP)
Two-pt boundary value probs: shooting, etc
Numerical methods for odes

Methods for state estimation

Properties of estimators

Nonlinear(/partial) least squares theory - connect to inferential dev, profit sensor/profit toolkits

Kalman filtering
  state, parameter estimation
  stability of the Kalman filter

Maximum likelihood estimation
MLE examples
Algorithms for ML estimation

Stochastic process control

Prob review
Autocorrelation fns

Background on stochastic processes
Stochastic control without filtering

Linear stochastic control without filtering

Stochastic control with filtering
linear

Linear Quadratic Gaussian regulator

Dynamic Programming and RTO (HJB eqn)
Numerical methods for DP

Model predictive control

Model identification and parameter estimation
- step tests
- freq domain vs state space reps
- model compactness
- Likelihood-based vs least-squares techniques in model id

Model uncertainty; id of true ind variables, experimental design, model quality analysis, validating data independence, data slicing

Applications:

Linearization - fan pitch controls, azeotropes, high-purity distillation

cascades: multiple input control variables (feedback loops) for single output; more responsive to disturbances; combine feedback and feedforward; when to break cascades

plant testing, exptl design vs model identification
open/closed loop testing

prebuild sims of distillation column

controller build, commissioning

(5)

Control, estimation and optimization topics

- Optimal control theory - learn to redirect dynamics to desired ends
- Analytic solutions to OCT problems
- Algorithms for numerical optimization: stochastic and deterministic
- Controllability
- Observability
- Estimation methods - likelihood-based, Bayesian; estimation algorithms: assess statistical error and incorporate
- Optimal feedback control: Hamilton-Jacobi-Bellman equations and dynamic programming
- Time permitting: model uncertainty

Course policy

- Research- and methods-oriented
- Homework assignments include code development for use in domain research
- 40% homework, coding; 20% midterm project; 20% final project; 20% final report
- Codes developed will be available on blackboard for registered students
- 500 lecture slides on optimization/control techniques available on blackboard for registered students
- Pass/fail option permitted

Numerical methods covered in HW exercises

Learn how to computationally optimize chemical, mechanical, electrical or molecular objective functions

- Genetic and evolutionary optimization
- Multiobjective optimization
- Constrained optimization (Newton-Raphson)
- Runge-Kutta ODE integration
- Markov Chain Monte Carlo numerical integration (MCMC)
- Self-consistent iterative algorithms
- Controllability and observability assessment

Some of the codes you write may be run in high performance parallel format to accelerate your research

Controllability and observability

CHE 597

Purdue University

March 22, 2010

Outline

Outline

Introduction: energy eigenstate controllability of molecular systems

- We saw that the matrix elements of the dipole moment operator, $\langle i|\mu|j\rangle$ determine the selection rules for light-induced transitions in atoms and molecules
- However, we also saw that direct (one photon) transitions between energy levels are not the only route for "state-to-state" transitions
- What are the analog of "selection rules" for multiphoton transitions?
- More generally, what determines if a initial wavefunction $|\psi(0)\rangle$ can be driven to any arbitrary final state $|\psi\rangle$ (at time T)?
- Subject is called *controllability*

Outline

Reachable set

Reachable set

The *reachable set* $R(x_0, T)$ at time T is the set of states $x(T)$ that can be reached from x_0 (the initial state) by an admissible control

- The *complete reachable set* $R(x_0) = \bigcup_{T>0} R(x_0, T)$

Controllability

Full controllability

A control system is (fully) *controllable* (at time T) if the reachable set $R(x_0, T)$ is equal to the state manifold.

- For unitary propagator (operator) quantum control, full controllability means $R(\psi_0, T) = \mathcal{U}(N)$
- For pure state control, full controllability means $R(U_0, T) = S_{\mathcal{H}_N}$
- Controllability theory does not rest on use of any particular cost functional, but sufficient conditions for controllability are sometimes conveniently derived using Lagrange functionals (with final state specified)

Formal solution to general linear systems

- Obtain the formal solution to the linear vector differential equation $\frac{dx}{dt} = Ax(t) + Bu(t)$ in two steps:
  1. Solve the homogeneous differential equation $\frac{dx}{dt} = Ax(t)$; this provides a reference frame moving with $x(t)$ in absence of control
  2. Rotate $x(t)$ to this reference frame; this produces a differential equation for $y(t) = U^{-1}(t)x(t)$; solve it by direct integration and then rotate back to the original reference frame
- We know the solution to the homogeneous equation is $x(t) = \exp(At)x(0)$ (matrix exponential) and the time evolution propagator $U(t) = \exp(At)$ satisfies $\frac{dU}{dt} = AU(t)$
- For the second step, use $\dot{U}^{-1}(t) = -U^{-1}(t)A$; hence $\dot{U}^{-1}(t)x(t) = -U^{-1}(t)Ax(t)$

$$\frac{d}{dt}\left(U^{-1}(t)x(t)\right) = U^{-1}(t)\frac{dx}{dt} + \dot{U}^{-1}(t)x(t) = U^{-1}(t)[Ax(t) + Bu(t)] - U^{-1}(t)Ax(t)$$

- So $\frac{d}{dt}\left(U^{-1}(t)x(t)\right) = U^{-1}(t)Bu(t)$ or

$$x(T) = U(T)x(0) + U(T)\int_0^T U^{-1}(t)Bu(t)\,dt$$

Formal solution to general linear systems: Laplace transform

- Consider solution of the general first-order scalar ode with constant coefficients: $\frac{dx(t)}{dt} = ax(t) + bu(t)$ with general, unknown control function $u(t)$ (not necc optimal for quadratic cost), via Laplace transforms
- $\mathcal{L}[\frac{dx(t)}{dt}] = \mathcal{L}[ax(t) + bu(t)]$
- Generalize to system of first-order linear odes

$$(sI - A)x(s) = x(0) + Bu(s)$$
$$x(s) = (sI - A)^{-1}[x(0) + Bu(s)]$$

- Inverse LT gives

$$x(T) = U(T)x(0) + U(T)\int_0^T U^{-1}(t)Bu(t)\ dt$$

(compare $\mathcal{L}^{-1}[x(s)] = \mathcal{L}^{-1}\left[\frac{x(0)+bu(s)}{s-a}\right]$ for scalar x)

Full controllability of time-invariant linear systems

- For linear control systems, it is a simple matter to assess full (state) controllability; find conditions that guarantee that $x(T)$ can be driven to 0 (since transferring system from any initial state to any final state may be put in this form by placing origin of state vector at desired target state)

$$x(T) = 0 = \exp(AT)x_0 + \int_0^T \exp(A(T-t'))Bu(t')\,dt'$$
$$= \exp(AT)\left[x(0) + \int_0^T \exp(-At')Bu(t')\,dt'\right]$$

(recall $u(t)$ for linear systems is m-component vector of controls, B is $N \times m$ matrix)

- According to the *Cayley-Hamilton theorem*, instead of Taylor expanding the matrix exponentials, we may represent them as matrix polynomials with at most $N-1$ terms:

$$\exp(-At) = a_0(t)I_N + a_1(t)A + a_2(t)A^2 + \cdots + a_{N-1}(t)A^{N-1}$$

The (time-invariant) controllability matrix

- So we have (left multiplying by $\exp(-AT)$)

$$-x(0) = B\int_0^T a_0(t')u(t')\ dt' + AB\int_0^T a_1(t')u(t')\ dt' + \cdots +$$
$$+ A^{N-1}B\int_0^T\int_0^T a_{N-1}u(t')\ dt'$$

- Can write as $[B, AB, \cdots, A^{N-1}B][\int_0^T a_0(t')u(t')\ dt', \cdots, \int_0^T a_{N-1}(t')u(t')\ dt']^T$ (note latter is Nm-dim vector since u is m-dim)
- The $N \times Nm$ *controllability matrix* is $[B, AB, \cdots, A^{N-1}B]$. If it is nonsingular (has N linearly independent rows/columns; or N nonzero singular values; or rank is N), the system is fully controllable since we can solve for $u(t)$ from this system of equations and independently drive all N elements of $x(T)$ to 0
- Check rank condition by singular value decomposition of controllability matrix (matrix is square only for one control)

Rank condition: numerical methods

Singular value decomposition

Recall the definition of *singular value decomposition*: for an $N \times m$ matrix A, the singular value decomposition is

$$A = USV^T,$$

where U is an $N \times m$ orthogonal matrix, S is a $m \times m$ diagonal matrix, and V is a $m \times m$ orthogonal matrix. The *singular values* of A are the diagonal elements $s_1, \cdots, s_m$; $s_i = +\sqrt{\lambda_i}$, where λ_i are the eigenvalues of $B = A^T A$.

- Columns of U (left singular vectors of A) corresponding to $s_i \neq 0$ are orthonormal basis vectors for the vector space spanned by the columns of A (range of A)
- This method for constructing an orthonormal basis is much more numerically stable than standard Gram-Schmidt orthogonalization

Controllability of time-invariant linear systems

- Use a quadratic Lagrange cost $L(u(t)) = u^T(t)u(t)$ with a terminal state constraint x_f
- Recall the form of the optimal control for the temperature control problem; generalize to vector linear system: $\bar{u}(t) = -B^T\phi(t)$
- Similarly generalize the costate differential equations: $\frac{d\phi(t)}{dt} = -A^T\phi(t)$
- Generalize $\phi(t)$ solution as $\phi(t) = \exp(A^T(T-t))\phi(T)$
- Then the state system of odes becomes: $\frac{dx}{dt} = Ax(t) - BB^T\exp(A^T(T-t))\phi(T)$ or $\frac{dx}{dt} = Ax(t) - BB^TU^T(T,t)\phi(T)$

Controllability of time-invariant linear systems

- Use the explicit solution for the linear system of odes above:

$$x(t) = U(t)x(0) - U(t)\left(\int_0^t U^{-1}(t')BB^T U^T(T,t')\,dt'\right)\phi(T)$$
$$= U(t)x(0) - \left(\int_0^t U(t,t')BB^T U^T(t,t')\,dt'\right)\phi(T);$$

  solve for $\phi(T)$ given known $x(T)$:

$$\phi(T) = \left(\int_0^T U(T,t')BB^T U^T(T,t')\,dt'\right)^{-1}(U(T)x(0) - x_f)$$

- Then substituting $\phi(T)$, obtain $\bar{u}(t) = B^T U^T(t,t)G^{-1}(T)[x_f - U(T)x(0)]$; condition for full controllability at time T is that the $N \times N$ *controllability Gramian*

$$G(T) = \int_0^T U(T,t')BB^T U^T(T,t')\,dt'$$

  is nonsingular

Local controllability

- For (time-varying) nonlinear systems (i.e., $\frac{dx}{dt} = F(x, u)$), there are no general rules for assessing full (state) controllability
- Must generally limit to *local controllability*, i.e., whether there exists a control perturbation $\delta u(t)$ that can achieve any arbitrary small perturbation from a nominal (reference) trajectory
- Denoting the reference trajectory by $x_r(t)$ and the perturbed trajectory by $x(t)$, we have

$$x(T) = x_r(T) + U(T)\delta x(0) + \int_0^T U(T, t')B(t')\delta u(t')\ dt'$$

  where $B(t')$ denotes the $N \times m$ Jacobian matrix $\frac{\partial F}{\partial u(t)}$ and $U(T) = \mathrm{T}\exp[\int_0^T \frac{\partial F}{\partial x(t)}\ dt]$ is $N \times N$ (both partial Jacobians evaluated at $x = 0, u = 0$)
- Local controllability is equivalent to the ability to drive all components of $x(T)$ to 0 by appropriate choice of $\delta u(t)$ over the interval 0, T

Local controllability (cont'd)

- A sufficient condition for local controllability is that the $N \times N$ *controllability Gramian matrix*

$$G(T) = \int_0^T U(T,t')B(t')B^T(t')U^T(T,t')\,dt'$$

  is nonsingular
- This follows because the control perturbation $\delta u(t)$ necessary to drive $x(T)$ to zero is $\delta u(t) = B^T(t)U^T(T,t)G^{-1}(T)[-x_r(T) - U(T)\delta x(0)]$ (note can set $\delta x(0) = 0$ if interested in control perturbations alone)
- Note that for linear time-variant systems, the controllability condition is derived as above but setting $x_r(t) = 0$
- However, for bilinear systems (a particular class of nonlinear systems), full controllability criteria exist

Controllability versus optimality of controls

- Optimal control theory seeks to maximize a cost function that may contain a contribution from the state as well as the control
- For Bolza and Mayer cost functionals, optimality of the control does not imply that a desired state is reachable.
- For Lagrange functionals, generally check controllability/reachability before imposing a terminal state constraint.
- If the system is uncontrollable, numerical algorithms may never achieve perfect objective function fidelity!

Outline

Observability of time-variant linear systems

- Consider the time-variant linear system $\frac{dx}{dt} = A(t)x(t)$ in the absence of control, with formal solution $x(t) = U(t)x_0$
- Consider a *linear observer* $y(t) = C(t)x(t) = C(t)U(t)x_0$, where $C(t)$ is $m \times N$
- The aim is to solve for x_0 by making m observations $y(t)$ at each time t
- To obtain a sufficient condition for this solution to exist, left-multiply the observation equation by $U^T(t)C^T(t)$ and integrate over all time:

$$\int_0^T U^T(t)C^T(t)y(t)\ dt = \int_0^T U^T(t)C^T(t)C(t)U(t)\ dt\ x_0$$

- Let $H(T) = \int_0^T U^T(t)C^T(t)C(t)U(t)\ dt$; note it is an $N \times N$ Gramian matrix. Now solve for x_0:

$$x_0 = H^{-1}(T) \int_0^T U^T(t)C^T(t)y(t)\ dt$$

- H is called the *observability Gramian matrix*.

Observability of time-invariant linear systems: rank condition

- Observability: Does there exist an observation sequence $y(t)$, $0 \leq t \leq T$, such that we can identify any $x(0)$? (note duality between controls (inputs) and observations (outputs))
- Consider the time-variant linear system $\frac{dx}{dt} = Ax(t)$ in the absence of control, with formal solution $x(t) = U(t)x_0$, with Bolza cost $J = \int_0^T x^T(t)Qx(t)\ dt + \frac{1}{2}x^T(T)S(T)x(T)$, $Q > 0$ and $Q^T = Q$ (the reason for notation $S(T)$ for endpoint weighting matrix will become clear below)

$$\frac{dx}{dt} = Ax$$
$$\frac{d\phi}{dt} = -Qx - A^T\phi$$

- Solve formally for $\phi(t)$:

$$\phi(t) = \exp[A^T(T-t)]\phi(T) + \int_t^T \exp[A^T(T-t')]Q\exp[A^Tt']x(t)\ dt'$$

$$\phi(0) = \exp[A^TT]\phi(T) + \int_0^T \exp[A^Tt']Q\exp[At']x(0)\ dt'$$

Observability of time-invariant linear systems: rank condition

$$\frac{dx}{dt} = Ax$$
$$y = \sqrt{Q}x$$
$$= \sqrt{Q}\exp(At)x_0$$

$$\phi(0) = \exp[A^T T]\phi(T) + \int_0^T \exp[A^T t']Q\exp[At']x(0)\ dt'$$

$$x(0) = \left[\int_0^T \exp[ATt']Q\exp[At']\ dt'\right]^{-1}\left[\phi(0) - \exp[A^T T]\phi(T)\right]$$

Observability of time-invariant linear systems: rank condition (cont)

- Compare $BR^{-1}B^T\lambda$ for Bu: now $\sqrt{Q}^T\sqrt{Q}x$; express in terms of $y = \sqrt{Q}x$: $\sqrt{Q}^T(\sqrt{Q}x)$
- m-component vector y is nothing but analog (dual) of control vector u

$$\frac{d\phi}{dt} = \sqrt{Q}^T y + A^T\phi$$

- Formally,

$$\phi(0) - \exp[A^T T]\phi(T) = \int_0^T \exp[A^T t]\sqrt{Q}^T y(t)\ dt$$
$$= \int_0^T \exp[A^T t]\sqrt{Q}^T\sqrt{Q}x(t)\ dt,$$

although actual reconstruction of $x(0)$ requires measurement outcomes $y(t)$; ϕ formulation useful only for observability assessment

Observability of time-invariant linear systems: rank condition (cont)

- Setting $\phi(T) = 0$,

$$\phi(0) = C^T \int_0^T a_0(t')y(t')\ dt' + A^T C^T \int_0^T a_1(t')y(t')\ dt' + \cdots +$$
$$+ (A^T)^{N-1} C \int_0^T \int_0^T a_{N-1} y(t')\ dt'$$

- Since for linear systems there is a one-to-one correspondence between $\phi(0)$ and $x(0)$ (see above), if this equation can be solved for $y(t)$ the system is observable
- Can write as $[C^T, A^T C^T, \cdots, (A^T)^{N-1} C^T][\int_0^T a_0(t')y(t')\ dt', \cdots, \int_0^T a_{N-1}(t')y(t')\ dt']$ (note latter is Nm-dim vector since y is m-dim)
- The $N \times Nm$ *observability matrix* is $[C^T, A^T C^T, \cdots, (A^T)^{N-1} C^T]$. If it is nonsingular (has N linearly independent rows/columns; or N nonzero singular values; or rank is N), the system is fully controllable since we can solve for $y(t)$ from this system of equations and independently identify all elements of $x(0)$
- Check rank condition by singular value decomposition of observability matrix (matrix is square only for one component

Outline

Controllability of bilinear systems

- Consider the general bilinear control system:

$$\frac{dx(t)}{dt} = \left[A + \sum_i B_i u_i(t) \right] x(t)$$

Controllability of bilinear systems

- Consider the general bilinear control system:

$$\frac{dx(t)}{dt} = \left[A + \sum_i B_i u_i(t) \right] x(t)$$

- Note we assume the possibility of multiple controls (e.g., components of the electric or magnetic field) with associated Hamiltonians $B_i,\ 1 \leq i \leq m$

Controllability of bilinear systems

- Consider the general bilinear control system:

$$\frac{dx(t)}{dt} = \left[A + \sum_i B_i u_i(t)\right] x(t)$$

- Note we assume the possibility of multiple controls (e.g., components of the electric or magnetic field) with associated Hamiltonians $B_i,\ 1 \leq i \leq m$
- Control consists of applying each control Hamiltonian B_i with amplitude $u_i(t)$, generally in unison, at each time interval dt

Controllability of bilinear systems

- Consider the general bilinear control system:

$$\frac{dx(t)}{dt} = \left[A + \sum_i B_i u_i(t) \right] x(t)$$

- Note we assume the possibility of multiple controls (e.g., components of the electric or magnetic field) with associated Hamiltonians $B_i, \ 1 \leq i \leq m$
- Control consists of applying each control Hamiltonian B_i with amplitude $u_i(t)$, generally in unison, at each time interval dt
- The important feature of bilinear control systems that makes their controllability easier to assess than general nonlinear control systems is the fact that the solution to the ode can be formally expressed as a matrix exponential

Controllability of bilinear systems

Definition

A *Lie algebra* $\mathcal{L}$ is a vector space over a field $\mathcal{F}$ (here, real or complex numbers) together with a bilinear operation $[\cdot,\cdot] : \mathcal{L} \times \mathcal{L} \to \mathcal{L}$ called a *Lie bracket* that satisfies the following conditions:

1. Bilinearity: $[x+z,y] = [x,y]+[z,y]$, $[x,y+z] = [x,y]+[x,z]$; $\alpha[x,y] = [\alpha x,y] = [x,\alpha y]$
2. Skew-symmetry: $[x,y] = -[y,x]$
3. Jacobi identity: $[x,[y,z]] = -([z,[x,y]]+[y,[z,x]])$

- We will be concerned with Lie algebras where x, y are $N \times N$ matrices A, B and the Lie bracket is the commutator $[A,B] = AB - BA$, with the field $\mathcal{F} = \mathbb{R}$. The matrices we are concerned with are skew-Hermitian, i.e., $A^\dagger = -A$. The Lie algebra $u(N)$ is the set of skew-Hermitian matrices together with the commutator.
- In this case, the matrix exponential $\exp(A)$ is an element of the associated *Lie group* (see hw for further definitions).
- Dynamical propagators in quantum mechanics are members of the unitary Lie group $\mathcal{U}(N)$

Application of BCH theorem

- The application of a single control Hamiltonian B_i (or $A + \sum_i u_i B_i$) with amplitude u_i for time Δt produces time evolution $\exp\left(-\frac{i}{\hbar} u_i(t) B_i \Delta t\right)$ (for qc systems)
- This corresponds to (we call this) "motion in direction iB_i; use notation $iB_i \mapsto B_i$
- Can we only move system along directions corresponding to sums $A + \sum_i u_i B_i$?
- No - non-commuting Hamiltonians produce new directions:

$$\exp(B_j \Delta t) \exp(B_i \Delta t) = \exp\Big\{ B_i \Delta t + B_j \Delta t + [B_i \Delta t, B_j \Delta t] + \frac{1}{2!}[B_i \Delta t, [B_i \Delta t, B_j \Delta t]] + \frac{1}{3!}[B_i \Delta t, [B_i \Delta t, [B_i \Delta t, B_j \Delta t]] + \cdots \Big\}$$

- Each commutator $[B_{i_1}, [B_{i_{n-1}}, B_{i_n}]] \cdots$ is a new direction
- For arbitrarily shaped controls, the system may be driven in any of these directions by appropriate choice of $u_i(t)$ (we will prove this as a homework problem)

Repeated Lie brackets

Definition

The *Lie algebra generated* by $\{A_1, \cdots, A_n\}$, where $A_i \in g$, a Lie algebra, is the subalgebra of g spanned by $\{A_1, \cdots, A_n\}$ and all their repeated commutators. We denote this Lie algebra by $\{A_1, \cdots, A_n\}_{LA}$.

- The *linear span* of the (possibly complex) matrices $\{A_1, \cdots, A_n\}$ is the set of all matrices $\sum_i c_i A_i$ with coefficients $c_i \in \mathbb{R}$.

If $A_1, \cdots, A_n$ are control Hamiltonians (i.e., for finite-dimensional quantum control systems, $g = u(N)$ or $su(N)$), the generated Lie algebra is called the *dynamical Lie algebra* $\mathcal{L}$ of the control system.

Definition

A *repeated Lie bracket* is a Lie bracket of the form $[A_n, \cdots, [A_2, A_1]]$.

Lie algebra rank condition

Dynamical Lie algebra

The dynamical Lie algebra $\mathcal{L} = su(N)$ ($\mathcal{L} = u(N)$) (i.e., the system is *fully operator controllable*; if the rank of the Lie algebra spanned by $\{A_1, \cdots, A_n\}$ and all their repeated commutators is $N^2 - 1$ (N^2).

The proof follows from application of the BCH theorem, since sequential application of the control Hamiltonians generates new directions in the Lie algebra

- Note this implies that there exists a T and controls $u_i(t)$ such that $U(T) = U$ for any $U \in \mathcal{U}(N)$; however, T can be very large and unknown.

Lie algebra rank condition: numerical methods

- To numerically check the Lie algebra rank condition:
  1. Construct elements in the dynamical Lie algebra by taking commutators $[H_0, X_i]$, $[\mu, X_i]$ for each X_i, with the initial set $\{X_i\} = \{H_0, \mu\}$
  2. For each element (matrix) X_i in the current set, construct a column vector whose elements are the linearly independent elements of the matrix
  3. Concatenate these column vectors to obtain an $N^2 \times M$ matrix A
  4. Do an SVD on A and obtain the rank of the range of A; if this is unchanged from the last iteration, this is the rank of the dynamical Lie algebra

Density matrix controllability

Unitarily equivalent states

Two density matrices (states) ρ_1, ρ_2 are said to be *unitarily equivalent* if we can write $\rho_2 = U\rho_1 U^\dagger$ for some unitary matrix U. Of course, this is the same as saying that ρ_1, ρ_2 share the same eigenvalue spectrum.

A quantum control system is said to be *density matrix controllable* if any density matrix ρ_2 is reachable from the all unitarily equivalent density matrices ρ_1.

Conditions for density matrix controllability

- Controllability of two unitarily equivalent states (states with a given eigenvalue spectrum) requires that the orbit $\{U\rho_1 U^\dagger | U \in \exp(\mathcal{L})\}$ is equal to the largest possible such set, $\{U\rho_1 U^\dagger | U \in \exp(\sqcap(\mathcal{N}))\}$.
- To test for density matrix controllability, we need a simple (numerically testable) condition for this
- Since all possible evolutions of ρ_0 under the action of arise from the commutators (recall the von Neumann equation), a quantum system is pure state controllable if

$$\dim[i\rho_0, \mathcal{L}] = \dim[i\rho_0, u(N)]$$

- The rhs of this equation it the dimension of the state manifold

Pure state controllability

- Recall that $\dim S_{\mathcal{H}_N} = 2N - 1$
- A quantum system is *pure state controllable* if

$$\dim[i\rho_0, \mathcal{L}] = 2N - 1$$

- Note for molecular control problems, the required condition is even weaker because only observable expectation values must be controlled
- Because pure state controllability is generally satisfied and due to the dependence of observable control on the nature of the observable, we will not consider the latter here

- Because $\mathcal{U}(N)$ is compact, quantum system controllability has additional favorable features beyond that of general bilinear systems
- Specifically: for a controllable system any propagator can be written $U(T) = \exp(-\frac{i}{\hbar}H_{i_n}t_n)\cdots\exp(-\frac{i}{\hbar}H_{i_1}t_1)$ with finite n, for some set of H_{i_j} in the dynamical Lie algebra
- This means that sequential *independent* application of control Hamiltonians can achieve any propagator or state (previously we considered arbitrary superpositions of Hamiltonians)
- There are important implications for quantum computing

Deterministic algorithms for optimization and control

CHE 597 - Quantum Control Engineering - Spring 2010

Purdue University

Feb 19-24, 2010

Outline

Outline

Optimization strategies for Bolza and Mayer costs

- For numerical solutions based on the gradient of the objective function J with respect to the control, need to integrate state, costate or both sets of differential equations with an implicit expression for the field $\bar{\varepsilon}(\psi(t), \phi(t))$ at each step.
- In the absence of additional symmetries, need to integrate both state and costate equations simultaneously.
- With the additional symmetries of Hermiticity of the matrices A, B and bilinearity of the control system, we can reduce the numerical problem to just integration of the state equations in terms of $\phi(T)$.
- Recall the form of the gradient of the PMP-Hamiltonian with respect to the control:

$$\frac{\partial \mathbf{H}}{\partial \varepsilon(t)} = -\frac{\mathrm{i}}{\hbar} \mathrm{Tr} \left(U_k^\dagger(T) \nabla_{U_k(T)} F(U_k(T)) U_k(^\dagger t) \mu U_k(t) \right)$$

for Mayer functionals and

$$\frac{\partial \mathbf{H}}{\partial \varepsilon(t)} = -\varepsilon(t) - \frac{\mathrm{i}}{\hbar} \mathrm{Tr} \left(U_k^\dagger(T) \nabla_{U_k(T)} F(U_k(T)) U_k(^\dagger t) \mu U_k(t) \right)$$

for Bolza functionals with quadratic fluence cost.

Computational considerations concerning the calculation of the gradient

- The above *analytical expression for the gradient* is equivalent to $\frac{\partial J(\varepsilon(\cdot))}{\delta\varepsilon(\cdot)}$ at each time t
- For numerical optimization, discretize the control: $\varepsilon(t) = (\varepsilon(t_1), \cdots, \varepsilon(t_n))$
- For gradient-based optimization of quantum systems, integrate just the Schrödinger equation using, e.g., Runge-Kutta algorithms and compute the gradient as above at each step; note there is no additional computational cost in applying gradient algorithms compared to algorithms that only use the value of $J(\varepsilon(t))$.
- Application of gradient-based optimization to general control systems requires the integration of the costate equations as well, to obtain the gradient; optimization algorithms based only on the value of $J(\varepsilon(\cdot))$ are less expensive per iteration (generally true)
- Algorithms based on the objective function value alone are typically *stochastic* algorithms - i.e., starting two optimizations from the same initial guess will not reach the same point on the parameter space in n steps - whereas those based on the gradient (and/or Hessian matrix of second derivatives) are typically *deterministic*.

Gradient flow (steepest ascent) algorithms

The simplest first-order algorithm is the gradient flow of the objective function; the gradient flow trajectory is the solution $u_s(t)$ to the initial value problem

$$\frac{\partial u_s(t)}{\partial s} = \alpha(s)\,\frac{\delta J(u(t))}{\delta u(t)}$$

for a specified initial guess for the control u_0, where $\alpha(s)$ is an adaptive step size.

- The discretized form of the gradient will be written $\nabla_{\mathbf{x}} J(\mathbf{x}_s)$.
- $\alpha(s)$ is typically determined by *line maximization* algorithms, which search for the lowest function value along a given direction (here the gradient), e.g. by trying a large α to start with, then backtracking until the minimum along the direction is found.
- We will discuss line maximization methods in both one- and multidimensions in a later lecture

More advanced deterministic algorithms: improvements on steepest ascent

- Note that the gradients $\nabla_{\mathbf{x}} J(\mathbf{x}_{s+1})$, $\nabla_{\mathbf{x}} J(\mathbf{x}_s)$ in the steepest ascent method on successive line maximizations are orthogonal, i.e. $\nabla_{\mathbf{x}} J(\mathbf{x}_{s+1}) \cdot \nabla_{\mathbf{x}} J(\mathbf{x}_s) = 0$, which means that successive steps do not "interfere" with each other's maximizations.
- However, note that $\nabla_{\mathbf{x}} J(\mathbf{x}_{s+2}) \cdot \nabla_{\mathbf{x}} J(\mathbf{x}_s) \neq 0$, so that may counteract the work done in the s-th minimization during the $s+2$-th maximization
- The notion of conjugate directions rectifies the above circumstance, based on a second-order approximation to the objective function near the maximum.
- The most basic improvements on steepest ascent - the *conjugate gradient* (CG) and the *quasi-Newton* (QN) methods - are derived based on second-order approximations of J. We will cover both in turn. These use only first-order information to find the optimum of a function under the quadratic approximation.

Conjugate gradient optimization

- Consider the 2nd-order Taylor expansion of an arbitrary multivariable function around a point $\bar{x}$:

$$f(x) \approx c + \nabla f(\bar{x})(x - \bar{x}) + \frac{1}{2}(x - \bar{x})^T H(\bar{x})(x - \bar{x}).$$

  Let the (symmetric) Hessian matrix H be full rank so there is a unique solution (let us use the notation $x_i \equiv \mathbf{x}_i$ for convenience.
- A well-behaved function can always be approximated in this way near the optimum $\bar{x}$, but let us now assume that this approximation is valid for any x_i, and make the replacement $\bar{x} \to x_i$.
- At step 0, set the step direction $h_0 = g_0$, where g_0 denotes $\nabla f(x_0)$. At step i, move in direction h_i until the function stops decreasing. Let $g_i = \nabla f(x_i)$. Condition for maximum *along a line*: $h_i \cdot g_{i+1} = 0$
- To improve upon SD, we ensure that all previous step directions are perpendicular to the change of the gradient ("conjugacy condition") that occurs during the current step. According to the first-order Taylor expansion for the gradient

$$g(x_{i+1}) - g(x_i) \approx H(x_i)(x_{i+1} - x_i)$$

- The "conjugacy" condition is then

$$\begin{aligned} g(x_{i+1}) - g(x_i) &= H(x_i)(x_{i+1} - x_i) \\ h_j^T[g(x_{i+1}) - g(x_i)] &= h_j^T H(x_i)(x_{i+1} - x_i) \\ &= h_j^T H(x_i) h_i = 0, \quad \forall j < i. \end{aligned}$$

  A *conjugate set* with respect to a symmetric matrix H is a set of vectors such that all h_j, h_i in the set satisfy $h_j^T H h_i$ if $j \neq i$
- The first-order Taylor expansion for the gradient may be written

$$g_{i+1} = g_i + \lambda_i H(x_i) h_i,$$

  where now we have scaled the step h_i by a factor λ_i. λ_i is chosen to maximize f along h_i. We can solve for this step by applying the condition $h_i^T g_{i+1} = 0$ (line maximum condition).
- Henceforth, use the notation $H \equiv H(x_i)$ (assume a quadratic form with constant Hessian)

Obtaining the step size

- To solve for the step size λ_i under the quadratic approximation, multiply both sides of $g_{i+1} = g_i + \lambda_i H h_i$ by h_i^T, and apply $h_i^T g_{i+1} = 0$:

$$h_i^T g_{i+1} = h_i^T g_i + \lambda_i h_i^T H h_i$$
$$\lambda_i = -\frac{h_i^T g_i}{h_i^T H h_i}$$

- Computationally, λ_i is found using a line maximization algorithm, which does not require calculation of $H(x_i)$.
- Now assume that at each step the new step h_{i+1} can be written as a linear combination of old step and new gradient vector:

$$h_{i+1} = g_{i+1} + \gamma_i h_i;$$

we next solve for γ_i.

Obtaining the step update

- We solve for the γ_i that satisfies the conjugacy condition for h_{i+1}, h_i:

$$h_{i+1}^T H h_i = (g_{i+1} + \gamma_i h_i)^T H h_i = 0$$
$$= g_{i+1}^T H h_i + \gamma_i h_i^T H h_i$$

So $\gamma_i = -\frac{g_{i+1}^T H h_i}{h_i^T H h_i}$. Since $Hh_i = \frac{g_{i+1} - g_i}{\lambda_i}$,

$$\gamma_i = \frac{-g_{i+1}^T (g_{i+1} - g_i) \frac{1}{\lambda_i}}{h_i^T (g_{i+1} - g_i) \frac{1}{\lambda_i}}.$$

Because $g_{i+1}^T g_i = 0$ and $h_i^T g_{i+1} = 0$, we obtain

$$\gamma_i = \frac{g_{i+1}^T g_{i+1}}{h_i^T g_i}.$$

- Although we have an expression for the λ_i's, they are computed using line maximization approaches, since CG does not use the Hessian matrix H (due to the expense of calculating it).
- Note that for a quadratic form (e.g., $f(x) = c + b^T x + \frac{1}{2}x^T Hx$ or simply $f(x) = \frac{1}{2}x^T Hx$, with Hessian H), the optimal $\bar{x} = x_0 + \sum_{i=1}^{n} \lambda_i h_i$, i.e., the n h_i's comprise a (non-orthogonal) basis for $\mathbb{R}^n$ (they are said to be "H-orthogonal") with basis expansion coefficient λ_i. The CG algorithm then converges to the optimum of the function in exactly n steps, whereas steepest ascent may take an arbitrarily large number of steps to converge depending on the initial guess.
- The "conjugacy" of the directions in the above derivation holds rigorously only for a quadratic form, where H is constant. In general, H will be a function of $\mathbf{x}_i$, but we do not compute it in CG.

Conjugate gradient optimization (cont)

- It can be shown (try it) that h_i step directions constructed by this algorithm are *all* conjugate for a quadratic form, i.e.

$$h_j^T H h_i = 0$$

for all $j < i$ as well as

$$\begin{aligned} g_i \cdot g_j &= 0 \\ g_i \cdot h_j &= 0, \quad j < i. \end{aligned}$$

- The conjugate gradient method converges to the solution in N steps for a function f that is a quadratic form; a more sophisticated convergence analysis is required for other functions, which we may revisit later.

Outline

Quasi-Newton methods

- *Newton's method* in multidimensions uses the inverse Jacobian matrix to find the roots of a system of nonlinear equations.
- When these equations correspond to the components of the gradient vector, the method can be used to find minima/maxima
- *Quasi-Newton* methods are applied only to function optimization. They are based on the similar principles to conjugate gradient, but rather than searching for conjugate directions based on gradient information, they directly use the approximations to (inverse) Hessian to compute successive step directions.
- The approximations to the inverse Hessian in Quasi-Newton methods only require computation of the gradient!
- We start with Newton's method (also called the Newton-Raphson method) in multidimensions

Quasi-Newton methods (cont)

- Consider finding the solution the system of equations $F(x) = 0$
- Denoting the components of the *Jacobian matrix* of F by $\mathcal{J}_{ij} = \frac{\partial F_i}{\partial x_j}$, we have $F(x_{i+1}) - F(x_i) \approx \mathcal{J}(x_i)(x_{i+1} - x_i)$
- Setting $F(x_{i+1}) = 0$, we get $x_{i+1} - x_i \approx -\mathcal{J}^{-1}(x_i)F(x_i)$ as the Newton step
- To apply this to minimization of a function $f(x)$, we set $F(x) = \nabla f(x)$. Then,

$$\nabla f(x_{i+1}) - \nabla f(x_i) \approx H(x_i)(x_{i+1} - x_i)$$
$$x_{i+1} - x_i \approx -\lambda_i H^{-1}(x_i)\nabla f(x_i)$$

  where in the second line we have set $\nabla f(x_{i+1}) = 0$ as the condition for reaching the maximum in one step, unlike conjugate gradient where we aim to reach the maximum *along a line* in each step. The step length $\lambda = 1$ for a quadratic form.
- We will return to the general Newton-Raphson (NR) method when we discuss numerical methods for constrained optimization.

Quasi-Newton methods (cont)

The Newton step is "successful" if $\Delta f = f(x_{i+1}) - f(x_i) > 0$. Consider the second-order Taylor expansion for Δf:

$$\Delta f = \nabla f(x_i)(x_{i+1} - x_i) + \frac{1}{2}(x_{i+1} - x_i)^T H(x_i)(x_{i+1} - x_i) > 0$$

Inserting the Newton step, we get

$$\begin{aligned}\Delta f &= -(x_{i+1} - x_i)^T H(x_i)(x_{i+1} - x_i) + \frac{1}{2}(x_{i+1} - x_i)^T H(x_i)(x_{i+1} - x_i) \\ &= -\frac{1}{2}(x_{i+1} - x_i)^T H(x_i)(x_{i+1} - x_i) > 0\end{aligned}$$

which is satisfied if H is negative-definite and the step scale $\lambda > 1/2$. Because this is not always true, Quasi-Newton methods replace the inverse Hessian with an approximate inverse Hessian Q such that $\lim_{i\to\infty} Q(x_i) = H^{-1}(x_i)$.

Hessian updating schemes

- $x - x_i = -H^{-1}\nabla f(x_i)$
- This would take one to max if f is quadratic form; instead use line search to see where to stop
- Don't know H, H^{-1}; Start w e.g. $Q_0 = \pm I$ as H^{-1} guess (initial guess depends on whether we are maximizing or minimizing f)
- Subtract equations at iterations i and $i+1$ and let $\nabla f_i \equiv \nabla f(x_i)$:

$$x_{i+1} - x_i = Q_{i+1}(\nabla f_{i+1} - \nabla f_i)$$

- Note we have chosen to require the *new* approximate inverse Hessian Q_{i+1} satisfies this condition just like the real inverse Hessian would if f were a quadratic form

Hessian updating schemes (cont)

- $Q_{i+1} = Q_i +$ correction term
- Find possible correction terms consistent with above expression
- Since the approximate inverse Hessian must be symmetric, the inverse Hessian update must take the form $Q_{i+1} = Q_i + Q_i^u$, where the correction term Q_i^u is also a symmetric matrix
- A general symmetric matrix of order n can be written in the form $\sum_{i=1}^{n} a_i v_i v_i^T = \sum_{i=1}^{n} a_i v_i \otimes v_i$, i.e. as an expansion over the outer products of its eigenvectors v_i (with the expansion coefficients being the corresponding eigenvalues).
- The most common updating schemes are *rank-two updates*, i.e., $Q_i^u = a_1 v_1 \otimes v_1 + a_2 v_2 \otimes v_2$
- Rank two updates provide more flexibility in satisfying the QN condition on the inverse Hessian while generating efficient update scheme
- The standard rank-two update schemes are called DFP (Davidon-Fletcher-Powell), and BFGS (Broyden-Fletcher-Goldfarb-Shanno) updates; they are closely related, with the BFGS generally performing better.

DFP (Davidon-Fletcher-Powell) updating

- The DFP updating scheme for the inverse Hessian approximation uses $v_1 = x_{i+1} - x_i \equiv h_i$, and $v_2 = Q_i(\nabla f_{i+1} - \nabla f_i) := Q_i(g_{i+1} - g_i)$:

$$Q_{i+1} = Q_i + \frac{h_i \otimes h_i}{h_i^T(g_{i+1} - g_i)} - \frac{[Q_i(g_{i+1} - g_i)] \otimes (Q_i(g_{i+1} - g_i)]}{(g_{i+1} - g_i)^T Q_i(g_{i+1} - g_i)}$$

- Verify that this satisfies the QN required condition on the inverse Hessian by plugging into above expression $h_i = Q_{i+1}(g_{i+1} - g_i)$; this comes from 2nd term while third term cancels out contribution from Q_i
- We have

$$[h_i \otimes h_i](g_{i+1} - g_i) = h_i[h_i^T(g_{i+1} - g_i)]$$

and

$$[Q_i(g_{i+1} - g_i)]^T(g_{i+1} - g_i) = (g_{i+1} - g_i)^T Q_i(g_{i+1} - g_i)$$

- An advantage of QN methods over CG is that their formulation does not refer to precise maximization along each step direction (note we did not require $g_{i+1}^T g_i = 0$); we will return to this when we discuss line search methods below

BFGS (Browden-Fletcher-Goldfarb-Shanno) updating

- The BFGS update is analogous to the DFP update, but written for the Hessian instead of the inverse Hessian
- It follows from recognizing that if one has an update formula for $Q_i = H_i^{-1}$, one can obtain an update for H_i by replacing Q_i by H_i and interchanging the roles of $x_{i+1} - x_i \equiv h_i$ and $\nabla f_{i+1} - \nabla f_i := g_{i+1} - g_i$
- The BFGS update for H_i is then

$$H_{i+1} = H_i + \frac{(g_{i+1} - g_i) \otimes (g_{i+1} - g_i)}{(g_{i+1} - g_i)^T h_i} - \frac{(H_i h_i) \otimes (H_i h_i)}{h_i^T H_i h_i}$$

- The resulting formula for H_{i+1} can then be inverted to obtain the update for the inverse Hessian Q_{i+1}
- The reason the BFGS update can be applied with low computational expense, despite the fact that the update is defined in terms of the Hessian rather than inverse Hessian, is that there exists a analytic formula called the *Sherman-Morrison formula* for the inverse of a "matrix plus an update" when the update takes the form of an outer product of vectors.

Sherman-Morrison matrix inversion lemma

- Through a matrix Taylor expansion, we can simplify $(A + u \otimes v)^{-1}$:

$$\begin{aligned}
(A + u \otimes v)^{-1} &= ((I + A^{-1} u \otimes v)^{-1}) A^{-1} \\
&= (I - A^{-1} u \otimes v + A^{-1} u \otimes v \cdot A^{-1} u \otimes v) A^{-1} \\
&= A^{-1} - A^{-1} u \otimes A^{-1} v (1 - \lambda + \lambda^2 - \cdots) \\
&= A^{-1} - \frac{A^{-1} u \otimes A^{-1} v}{(1 - \lambda)}
\end{aligned}$$

  where we have used the associativity of matrix and tensor products and $\lambda = v^T A^{-1} u$.
- The Sherman-Morrison formula is

$$(A + u \otimes v)^{-1} = A^{-1} - \frac{(A^{-1} u) \otimes (A^{-1} v)}{1 - v^T A^{-1} u}$$

- You may apply it to the Hessian update above (possibly in a homework) to obtain the explicit expression for Q_{i+1} given Q_i (adds an additional correction term to DFP)
- S-M formula is very often used in numerical analysis to update inverse of a matrix given a perturbation with minimal computational expense

Outline

1. Introduction to numerical optimization algorithms
   - Gradient-based vs cost function-based optimization algorithms
   - Conjugate gradient
2. 2nd order algorithms
   - Newton's method
3. Line search algorithms using backtracking
4. Bracketing of minima (maxima) along a line
5. Algorithms for control optimization
   - The shooting method
   - Iterative control optimization algorithms based on PMP

Line search (adaptive step size) without bracketing

- Line search without bracketing is designed to increase the function "sufficiently" but not necessarily precisely to the line maximum
- These are commonly used in NR and QN methods, but not as much in CG (for which bracketing is used); the reason is that NR/QN do not require precise maximization along a line, as discussed
- Let $x_{new} = x_{old} + \lambda p, \quad 0 < \lambda \leq 1$ where p is the (Quasi-)Newton direction; for QN algorithm at step i, x_{old} is x_i, x_{new} is the current attempt at x_{i+1}
- Start with $\lambda = 1$; set acceptance criteria that must be satisfied, or otherwise reject and backtrack.
- Criteria not just $f(x_{new}) \geq f(x_{old})$. Require average rate of decrease of f to be at least fraction $\alpha < 1$ of initial rate of increase $(\nabla f \cdot p)$: i.e. check if $f(x_{new}) - f(x_{old}) \geq \alpha(\nabla f \cdot p)$

Polynomial line search (backtracking)

- Algorithm:
  1. Let $y(\lambda) = f(x_{old} + \lambda p)$; λ parametrizes a straight line through the parameter space in the direction p; then $\frac{dy}{d\lambda} = \nabla f \cdot p$, i.e., directional derivative of f along p Solve for second order coeff based on matching value at $y(1)$; then solve for zero of derivative. Use this maximum as next guess
  2. Do not compute the gradient at any point other than x_{old}; i.e., only $y'(0)$
  3. In next iteration use a cubic model (higher order Taylor approximation of $y(\lambda)$) based on same principle
- Step 1: $y(\lambda) = (y(1) - y(0) - y'(0))\lambda^2 + y'(0)\lambda + y(0)$; $y(1)$ is known
- Check: $y(1) = y'(0) + y(0) + y(1) - y'(0) - y(0)$
- Solve for $\lambda_2 = \lambda_{max}$ (i.e., λ such that $\frac{dy}{d\lambda} = 0$)

$$2\lambda(y(1) - y(0) - y'(0)) + y'(0) = 0$$

$$\lambda_2 = \frac{y'(0)}{2(y(1) - y(0) - y'(0))}$$

- The latter is the new λ guess; we have $\lambda_2 < 1$ since the curvature is negative

Polynomial line maximization (cont)

- Compute $y(\lambda_2)$ using λ_2 from quadratic model
- Now model $y(\lambda)$ as a cubic, using the four known values $y(0), y(1), y'(0), y(\lambda_2)$:

$$y(\lambda_2) = a\lambda_2^3 + b\lambda_2^2 + y'(0)\lambda_2 + y(0)$$
$$y(1) = a + b + y'(0) + y(0)$$

- Solve the above system of equations for a, b
- Find (local) maximum of the cubic:

$$\frac{dy}{d\lambda} = 3a\lambda_2^2 + 2b\lambda_2 + y'(0) = 0$$
$$\lambda_2 = \frac{-2b \pm \sqrt{(2b)^2 - 4(3a)y'(0)}}{6a}$$

Compare graphs for quadratic and cubic polynomials)

- Set $\lambda_{min} = \lambda_3$; note that $\lambda_3 < \lambda_2$
- Do same for $\lambda_4, \cdots, \lambda_n$, since higher order polynomials will have multiple local maxima

Outline

Golden section search

- Bracketing is a method for obtaining the minimum of an objective function J along a given direction (vector); it is typically used with conjugate gradient methods along the successive noninterfering directions
- A *bracket* of a minimum of an objective function J is a triplet of points $a < b < c$ where $f(a) > f(b)$ and $f(c) > f(b)$; we then have $a < x_{min} < c$; b is current guess for minimum
- *Golden section search*: updates bracketing until bracket is narrowed within a given tolerance
- Based on initial bracket, choose new pt x between a, b or b, c
- Given latter choice, if $f(b) < f(x)$, new bracket is a, b, x; otherwise b, x, c
- Given former choice, if $f(b) < f(x)$, new bracket is a, x, b; otherwise x, b, c

Golden section search (cont)

- Algorithms exist for choosing x given a, b, c: golden section search involves using larger of two intervals
- Let $\frac{b-a}{c-a} = w$, then $\frac{c-b}{c-a} = 1 - w$. Assume $w < 0.5$
- Call the first possible choice for the new bracket "Bracket I" (a, b, x) and the second "Bracket II" (b, x, c); assume (will validate shortly) that $b < x < c$.
- Impose condition that length of bracket I, $|x - a|$, equals length of bracket II, $|c - b|$.
- Then must have $|b - a| = |x - c|$
- Let $\frac{x-b}{c-a} = z$
- Since Bracket I is of length $(w + z)|c - a|$ and Bracket II is of length $(1 - w)|c - a|$, this implies $w + z = 1 - w$ or $z = 1 - 2w$ (**condition 1** on w)

Golden section search (cont)

- Secondly, require *scale similarity* between iterations - i.e., (b, x, c) is a smaller scaled version of (a, b, c): $\frac{z}{1-w} = w$ (**condition 2** on w)
- to be equal
- Solving for w given conditions 1 and 2 gives $w^2 - 3w + 1 = 0$; or $w = 0.38197$ (called golden mean)
- Continue until reaching tolerance in size of bracket (difference bet outer bounds)
- Convergence linear in sense of rate at which bracket size decreases (see above ratio)

Outline

Two-point boundary value problems

- Optimal control problems for Mayer functionals are often best solved using QN or CG methods
- For Mayer functionals, there is no way to express $\bar{u}(t)$ as implicit function of $x(t), \phi(t)$
- For Lagrange or Bolza functionals, we write $\bar{u}(t) = g(x(t), \phi(t))$ and then integrate x, ϕ odes in terms of known $x(0)$ and unknown $\phi(0)$
- Mathematically this is known as a system of differential equations with *split boundary conditions* or a *two-point boundary value problem*
- Even if we have analytic solutions for $x(t)$ and $\phi(t)$, if the state/costate odes are coupled, we cannot solve for the unknown integration constants in a single step

Two-point boundary value problem: example

- Numerically, we cannot just propagate the system of equations forward from a single $x(0), \phi(0)$ to obtain the solution
- This circumstance arises when, upon substitution of the implicit expression for the control in terms of $x(t), \phi(t)$, we obtain a coupled system of odes called a *PMP-Hamiltonian dynamical system*.
- Consider the following generic example of a scalar linear control system, whose PMP-Hamiltonian system is also linear:

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{d\phi}{dt} \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x(t) \\ \phi(t) \end{bmatrix} := A \begin{bmatrix} x(t) \\ \phi(t) \end{bmatrix}$$

  with $x(0)$ given, and $\phi(T) = \nabla_x F(x(T))$.
- This problem can be solved analytically but we will use it to illustrate the general numerical shooting approach.

Numerical methods for two-point boundary value problems: shooting method

- The *shooting method* (iteratively) converges upon the target $\phi(T)$ vector by making successive changes in the initial conditions $\phi(0)$; i.e., it shoots from $x(0), \phi(0)$, trying to hit the terminal boundary conditions $\phi(T)$
- Numerical algorithms for shooting are typically based on a combination of (i) the Newton-Raphson method; and (ii) the Runge-Kutta ODE integration method.
- RK is used to integrate the state/costate ODEs at each step, given $x(0)$ and guess for $\phi(0)$ vectors
- NR is used to solve for the roots of the boundary condition equations, i.e., $\phi(T) - \phi_f = 0$ Call these f_i and let $\phi_i(0) = c_i$; then NR step is, $\delta \mathbf{c} = \lambda J^{-1} F(\mathbf{c})$, where the elements of the Jacobian are $\mathcal{J}_{ij} \equiv \frac{\partial f_i}{\partial c_j}$

Shooting method (cont)

- Each iteration of NR (function evaluation) requires the integration of the $2n$ state, costate ODEs by RK
- QN is typically not used since would require taking additional derivatives in order to obtain gradient conditions rather than root conditions; do not have analytic derivatives. QN updates cannot be applied to Jacobian.
- For Lagrange-type costs, the n unknown terminal boundary conditions are on $x(T)$, not $\phi(T)$, but procedure otherwise same
- Stepsizes λ_i typically determined by polynomial line search
- Shooting can be applied to either Lagrange or Bolza functionals

Analytical methods for two-point boundary value problems

- For linear control systems, the elements of the Jacobian $\frac{\partial f_i}{\partial \phi_j(0)}$ (columns of the Jacobian matrix $\frac{\partial f}{\partial \phi_j(0)}$) can be identified analytically
- This provides further insight into the shooting method
- The method of *unit solutions* is used for this purpose
- *Method of unit solutions* for solving linear two-point boundary value problems relies on the principle of superposition: the notion that any solution to homogeneous system of linear differential equations can be represented as a linear combination of a complete set of basis functions (linearly independent solutions).

Method of unit solutions

- Integrate PMP-Hamiltonian system using initial conditions $x(0) = x_{init}$ and $\phi(0) = 0$; call resulting solution $x^0(t), \phi^0(t)$. In order to obtain n unknown initial conditions $\phi(0)$:
- Integrate with n initial conditions $\phi_i(0) = 1, \phi_j(0) = 0, \; j \neq i; \quad x_i(0) = 0, i = 1, \cdots, n$. Call the resulting solns $x^i(t), \phi^i(t)$.
- Write the general solution as linear combination

$$x(t) = x^0(t) + \sum_i c_i x^i(t)$$

$$\phi(t) = \phi^0(t) + \sum_i c_i \phi^i(t)$$

  note $x^0(t)$ will contain known initial conditions x_{init}
- By setting $\phi(T) = \nabla_x F(x(T))$, solve for the unknown coefficients $c_i = \phi_i(0)$
- For linear control systems, the $\phi^i(t)$ are the columns $\frac{\partial f}{\partial \phi_i(0)}$ and the Jacobian is constant; hence NR should converge in approximately 1 step
- Complete the solution by plugging the c_i into expressions for $x(t), \lambda(t), u(t)$

Method of unit solutions: scalar example

- Assume we have integrated the general scalar linear PMP-Hamiltonian system introduced above, without application of the initial conditions on $x(t)$ or (unknown) terminal conditions on $\phi(T)$.
- Method of unit solutions: (i) Write the solution with $x(0) = x_{init},\ \phi(0) = 0$, call it $[x^0(t), \phi^0(t)]^T$; then write solution with $x(0) = 0,\ \phi(0) = 1$, call it $[x^1(t), \phi^1(t)]^T$. Then we can express the true solution as

$$\begin{bmatrix} x(t) \\ \phi(t) \end{bmatrix} = \begin{bmatrix} x^0(t) \\ \phi^0(t) \end{bmatrix} + c \begin{bmatrix} x^1(t) \\ \phi^1(t) \end{bmatrix}$$

.
- Here, $f = \phi^0(T) + c\phi^1(T) - \nabla_x F(x(c, T))$, which is linear fn of c. The Jacobian is simply $\frac{\partial f}{\partial c} = \phi_1(T)$.
- Numerically, guess a value for c, solve for $x(c, T)$ from the 1st row of the vector equation above, solve for c_{opt} according to $c_{opt} = c - J^{-1} f(c)$.
- So, the linearity of the control system has enabled application of the principle of superposition, which in turn leads to the linearity of the optimization problem.

Self-consistent iterative algorithms: formulation

- A common optimization strategy for Bolza functionals is the use of so-called iterative algorithms that are based on the PMP.

Self-consistent iterative algorithms: formulation

- A common optimization strategy for Bolza functionals is the use of so-called iterative algorithms that are based on the PMP.
- An initial guess for $u(t)$ (denoted $\tilde{u}_0(t)$), is used to integrate the dynamical equation forward starting from initial condition x_0, and the costate equation backward from final condition $\nabla_{x(T)}F(x(T))$; these steps are iterated self-consistently.

Self-consistent iterative algorithms: formulation

- A common optimization strategy for Bolza functionals is the use of so-called iterative algorithms that are based on the PMP.
- An initial guess for $u(t)$ (denoted $\tilde{u}_0(t)$), is used to integrate the dynamical equation forward starting from initial condition x_0, and the costate equation backward from final condition $\nabla_{x(T)}F(x(T))$; these steps are iterated self-consistently.
- For a quadratic cost on the control (for other costs the implicit expression for $u(t)$ will change)

$$\frac{dx_k(t)}{dt} = f(x(t), \tilde{u}_k(t)), \;\; x(0) = x_0$$

$$\frac{d\phi_{k+1}(t)}{dt} = \nabla_{x(t)}\mathbf{H}(x_k(t), \phi_{k+1}(t), u_{k+1}(t)), \;\; \phi_{k+1}(T) = \nabla_{x(T)}F(x_k(T))$$

$$u_{k+1}(t) = \frac{\partial}{\partial u(t)}\langle \phi_{k+1}(t), f(x_k(t), u_{k+1}(t))\rangle$$

$$\tilde{u}_{k+1}(t) = \frac{\partial}{\partial u(t)}\langle \phi_{k+1}(t), f(x_{k+1}(t), u_{k+1}(t))\rangle$$

Optimal control theory

CHE 597

Purdue University

January 15, 2010

Outline

Outline

Course policy

- Research- and methods-oriented
- Homework assignments include code development for use in domain research
- 40% homework, coding; 20% midterm project; 20% final project; 20% final report
- Codes developed will be available on blackboard for registered students
- 500 lecture slides on optimization/control techniques available on blackboard for registered students
- Pass/fail option permitted

Control, estimation and optimization topics

- Optimal control theory - learn to redirect dynamics to desired ends
- Analytic solutions to OCT problems
- Algorithms for numerical optimization: stochastic and deterministic
- Controllability
- Observability
- Estimation methods - likelihood-based, Bayesian; estimation algorithms: assess statistical error and incorporate
- Optimal feedback control: Hamilton-Jacobi-Bellman equations and dynamic programming
- Time permitting: model uncertainty

Numerical methods covered in HW exercises

Learn how to computationally optimize chemical, mechanical, electrical or molecular objective functions

- Genetic and evolutionary optimization
- Multiobjective optimization
- Constrained optimization (Newton-Raphson)
- Runge-Kutta ODE integration
- Markov Chain Monte Carlo numerical integration (MCMC)
- Self-consistent iterative algorithms
- Controllability and observability assessment

Some of the codes you write may be run in high performance parallel format to accelerate your research

Extending control engineering to the micro, submicro and nanodomains

In addition to generic engineering applications of optimization and control methods,

- Introductory molecular quantum mechanics and quantum chemistry
- Atomic and molecular optimal control
- Laser control of reactive chemistry
- Optimal design of quantum computers (quantum dots, nuclear spins, etc)
- Optimal design and control for coherent quantum transport: exciton control for photovoltaics (nanosolar cells)
- Optimal control of semiconductor optical switching
- See distributed handouts for details
- This semester's course will be basis for molecular optimal control book by Chakrabarti and Rabitz, Taylor and Francis, 2011: be a part of the development
- New course: *register* for blackboard access to all course materials

Outline

Types of control systems

We will be concerned only with first-order systems, i.e., where the dynamics of the state evolution are specified by a system of first-order ordinary differential equations (ODEs). In optimal control, these are called the *dynamical equations of the variational system*.

Linear control system

A *linear* control system is one that is linear in the control and the state; it has the general form

$$\frac{dx}{dt} = Ax(t) + Bu(t)$$

where A is a $n \times n$ matrix, B is an $n \times m$ matrix, x is the n-component state vector and u is a m component vector of controls. A,B and x may be either real or complex; u must be real.

Bilinear control systems

Bilinear control system

A *bilinear* control system is one that is linear in both the control and the state, and where the control and state enter multiplicatively; it has the general form

$$\frac{dx}{dt} = \left[A + \sum_i B_i u_i(t)\right] x(t)$$

where each B_i is a $n \times n$ matrix and $u = (u_1, \cdots, u_m)$ is the m component vector of controls.

For linear and bilinear control systems, the term $Ax(t)$ is referred to as the *drift* of the control system, since it specifies how the system evolves when the control is turned off. (For bilinear systems in physics, A is sometimes referred to as the drift Hamiltonian, and B_i as the control Hamiltonians).

Nonlinear control systems

Nonlinear control system

A *nonlinear* control system is nonlinear in either the control, the state, or both; it cannot be expressed in either form above and has the general form

$$\frac{dx}{dt} = f(x(t), u(t)).$$

Outline

Bolza, Mayer and Lagrange type functionals

There are three primary types of optimal control cost functionals.

Bolza, Mayer and Lagrange type functionals

There are three primary types of optimal control cost functionals.

- The most "general" are functionals of the *Bolza type*:

$$J[x(\cdot),\ u(\cdot)] = F(x(T)) + \int_0^T L(x(t), u(t))\ dt,$$

Bolza, Mayer and Lagrange type functionals

There are three primary types of optimal control cost functionals.

- The most "general" are functionals of the *Bolza type*:

$$J[x(\cdot),\ u(\cdot)] = F(x(T)) + \int_0^T L(x(t), u(t))\ dt,$$

- If only the term $\int_0^T L(x(t), u(t))\ dt$ is present, the cost functional is said to be of the *Lagrange* type.

Bolza, Mayer and Lagrange type functionals

There are three primary types of optimal control cost functionals.

- The most "general" are functionals of the *Bolza type*:

$$J[x(\cdot),\ u(\cdot)] = F(x(T)) + \int_0^T L(x(t), u(t))\ dt,$$

- If only the term $\int_0^T L(x(t), u(t))\ dt$ is present, the cost functional is said to be of the *Lagrange* type.
- If only the term $F(x(T))$ is present, the functional is said to be of the *Mayer* type.

Outline

The Lagrangian functional

The optimal control problem may be stated as

$$\max_{u(\cdot)} J[x(\cdot),\ u(\cdot)] \tag{1}$$

subject to the constraint of the dynamical differential equation.

Define a Lagrangian functional $\bar{J}$ that directly imposes the constraint in the dynamical equation:

$$\bar{J}[\ x(\cdot), \phi(\cdot),\ x(\cdot)] = F(x(T)) + \int_0^T \left[\lambda L(x(t), u(t)) + \langle \phi(t), f(x(t), u(t), t) - \frac{dx(t)}{dt} \rangle \right] dt \tag{2}$$

First-order variation

- Define the *PMP-Hamiltonian* function

$$\mathbf{H}(x(t), \phi(t), u(t)) = \lambda L(x(t), u(t)) + \langle \phi(t), f(x(t), u(t), t) \rangle$$

.

First-order variation

- Define the *PMP-Hamiltonian* function

$$\mathbf{H}(x(t), \phi(t), u(t)) = \lambda L(x(t), u(t)) + \langle \phi(t), f(x(t), u(t), t) \rangle$$

.

- Expressing the Lagrangian in terms of **H** and integrating $\langle \phi(t), \frac{dx(t)}{dt} \rangle$ by parts, we get

$$\bar{J} = F(x(T)) - \langle \phi(T), x(T) \rangle + \langle \phi(0), x(0) \rangle + \int_0^T \mathbf{H}(x(t), \phi(t), u(t)) + \langle \frac{d\phi(t)}{dt}, x(t) \rangle \; dt.$$

First-order variation

- Define the *PMP-Hamiltonian* function

$$\mathbf{H}(x(t), \phi(t), u(t)) = \lambda L(x(t), u(t)) + \langle \phi(t), f(x(t), u(t), t) \rangle$$

.

- Expressing the Lagrangian in terms of **H** and integrating $\langle \phi(t), \frac{dx(t)}{dt} \rangle$ by parts, we get

$$\begin{aligned} \bar{J} = {} & F(x(T)) - \langle \phi(T), x(T) \rangle + \langle \phi(0), x(0) \rangle \\ & + \int_0^T \mathbf{H}(x(t), \phi(t), u(t)) + \langle \frac{d\phi(t)}{dt}, x(t) \rangle \; dt. \end{aligned}$$

- The first-order variation of this Lagrangian is

$$\begin{aligned} \delta \bar{J} = {} & \langle \nabla_{x(T)} F(x(T)) - \phi(T), \delta x(T) \rangle + \langle \phi(0), \delta x(0) \rangle + \\ & + \int_0^T \langle \nabla_{x(t)} \mathbf{H} + \frac{d\phi(t)}{dt}, \delta x(t) \rangle + \nabla_{u(t)} \mathbf{H} \cdot \delta u(t) \; dt. \end{aligned}$$

First-order variation

- Define the *PMP-Hamiltonian* function

$$\mathbf{H}(x(t), \phi(t), u(t)) = \lambda L(x(t), u(t)) + \langle \phi(t), f(x(t), u(t), t) \rangle$$

.

- Expressing the Lagrangian in terms of $\mathbf{H}$ and integrating $\langle \phi(t), \frac{dx(t)}{dt} \rangle$ by parts, we get

$$\bar{J} = F(x(T)) - \langle \phi(T), x(T) \rangle + \langle \phi(0), x(0) \rangle + \int_0^T \mathbf{H}(x(t), \phi(t), u(t)) + \langle \frac{d\phi(t)}{dt}, x(t) \rangle \; dt.$$

- The first-order variation of this Lagrangian is

$$\delta \bar{J} = \langle \nabla_{x(T)} F(x(T)) - \phi(T), \delta x(T) \rangle + \langle \phi(0), \delta x(0) \rangle + + \int_0^T \langle \nabla_{x(t)} \mathbf{H} + \frac{d\phi(t)}{dt}, \delta x(t) \rangle + \nabla_{u(t)} \mathbf{H} \cdot \delta u(t) \; dt.$$

- The corresponding first-order conditions (Euler-Lagrange equations) follow from the requirement that $\delta \bar{J} = 0$ for any δu, and hence for any $\delta x(t)$.

Euler-Lagrange equations 1,2

The first two E-L equations are

1. $\nabla_{x(t)}\mathbf{H} + \frac{d\phi(t)}{dt} = 0,$
2. $\nabla_{u(t)}\mathbf{H} = 0, \quad 0 \leq t \leq T.$

Dynamical equation of the adjoint system

The first Euler-Lagrange equation can be expanded as

$$\begin{aligned}\frac{d\phi(t)}{dt} &= -\nabla_{x(t)}\mathbf{H} \\ &= -\lambda\nabla_{x(t)}L(x(t), u(t)) - \nabla_{x(t)}\langle\phi(t), f(x(t), u(t))\rangle,\end{aligned}$$

which is referred to as the dynamical equation for the adjoint system. We will write explicit forms of the E-L equations for linear and bilinear systems, in turn.

Dynamical equation of the adjoint system: linear control

For linear control systems, we can make the identification

$$\nabla_{x(t)}(\mathbf{H} - \lambda L) = A^{\dagger}\phi(t).$$

So, we have

$$\frac{d\phi(t)}{dt} = -\lambda\left(\nabla_{x(t)} L\right) - A^{\dagger}\phi(t)$$

Dynamical equation of the adjoint system: bilinear control

- For bilinear control systems, we can make the identification

$$\nabla_{x(t)}(\mathbf{H} - \lambda L) = \left(A^\dagger + B^\dagger u(t)\right) \phi(t)$$

So we have

$$\frac{d\phi(t)}{dt} = -\lambda \nabla_{x(t)} L - \left(A^\dagger + B^\dagger u(t)\right) \phi(t).$$

Dynamical equation of the adjoint system: bilinear control

- For bilinear control systems, we can make the identification

$$\nabla_{x(t)}(\mathbf{H} - \lambda L) = \left(A^\dagger + B^\dagger u(t)\right) \phi(t)$$

  So we have

$$\frac{d\phi(t)}{dt} = -\lambda \nabla_{x(t)} L - \left(A^\dagger + B^\dagger u(t)\right) \phi(t).$$

- If $L = L(u(t))$ (i.e., L is not a function of $x(t)$, which is almost always the case since $-L$ typically represents a resource cost) we have

$$\frac{d\phi(t)}{dt} = -\left(A^\dagger + B^\dagger u(t)\right) \phi(t).$$

Boundary conditions for the 2nd E-L equation

- If the cost function is of Mayer or Bolza type (latter required for linear systems), the 2nd E-L eqn is associated with boundary condition
$$\phi(T) = \nabla_{x(T)} F(x(T)),$$

- Note that the boundary conditions for the optimal control problem with endpoint cost, specified in the variational and adjoint equations, are "split" between the initial and final times; the costate $\phi(t)$ is propagated backwards in time starting from $\phi(T)$, whereas the "state" $x(t)$ is propagated forward in time starting from $x(0)$.

For linear systems,

$$\frac{\partial \mathbf{H}}{\partial u(t)} = 0, \quad 0 \le t \le T$$
$$= \lambda \nabla_{u(t)} L(u(t)) + \langle \phi(t), \vec{b} \rangle = 0$$

where $\vec{b}$ is an n-component vector that is the first column of B.

The third Euler-Lagrange equation: bilinear systems

Whereas for bilinear systems,

$$\begin{aligned}\frac{\partial \mathbf{H}}{\partial u(t)} &= 0, \quad 0 \leq t \leq T \\ &= \lambda \nabla_{u(t)} L(u(t)) + \langle \phi(t), Bx(t) \rangle.\end{aligned}$$

Outline

Pontryagin Maximum Principle (PMP)

The Euler-Lagrange equations can be succinctly stated in terms of the Pontryagin Maximum Principle.

For the class of problems considered above with fixed terminal time T, the Pontryagin Maximum Principle is:

Theorem

(Pontryagin) An optimal control $\bar{u}(\cdot)$ that solves the control problem $\max \bar{J}$ satisfies $\frac{\partial \mathbf{H}}{\partial u(t)} = 0$ for a matrix $\phi(T) = \nabla_{x(T)} F(x(T))$ for Bolza or Mayer functionals (otherwise unspecified for Lagrange functionals) and scalar λ where at least one of $\phi(T), \lambda$ is nonzero.

PMP conditions for functionals: bilinear control

- For a bilinear control systems, the PMP thus demands that

$$\frac{\partial \mathbf{H}}{\partial u(t)} = \lambda \frac{\partial L(u(t))}{\partial u(t)} + \langle \phi(t), Bx(t) \rangle = 0, \quad 0 \leq t \leq T,$$

for cost functionals of the Lagrange type.

PMP conditions for functionals: bilinear control

- For a bilinear control systems, the PMP thus demands that

$$\frac{\partial \mathbf{H}}{\partial u(t)} = \lambda \frac{\partial L(u(t))}{\partial u(t)} + \langle \phi(t), Bx(t) \rangle = 0, \quad 0 \leq t \leq T,$$

  for cost functionals of the Lagrange type.

- For cost functionals of the Bolza type, we have

$$\frac{\partial \mathbf{H}_B}{\partial u(t)} = \lambda \frac{\partial L(u(t))}{\partial u(t)} + \langle \phi(t), Bx(t) \rangle = 0$$

  for bilinear control systems, with the boundary condition $\phi(T) = \nabla_{x(T)} F(x(T))$ on the costate imposed.

PMP conditions for functionals: bilinear control

- For a bilinear control systems, the PMP thus demands that

$$\frac{\partial \mathbf{H}}{\partial u(t)} = \lambda \frac{\partial L(u(t))}{\partial u(t)} + \langle \phi(t), Bx(t) \rangle = 0, \quad 0 \le t \le T,$$

  for cost functionals of the Lagrange type.

- For cost functionals of the Bolza type, we have

$$\frac{\partial \mathbf{H}_B}{\partial u(t)} = \lambda \frac{\partial L(u(t))}{\partial u(t)} + \langle \phi(t), Bx(t) \rangle = 0$$

  for bilinear control systems, with the boundary condition $\phi(T) = \nabla_{x(T)} F(x(T))$ on the costate imposed.

- For cost functionals of the Mayer type, we have

$$\frac{\partial \mathbf{H}_B}{\partial u(t)} = \langle \phi(t), Bx(t) \rangle = 0$$

  for bilinear control systems, with the boundary condition $\phi(T) = \nabla_{x(T)} F(x(T))$ on the costate imposed.

PMP conditions for functionals: linear control

- For linear control systems, the PMP demands that

$$\frac{\partial \mathbf{H}}{\partial u(t)} = \lambda \frac{\partial L(u(t))}{\partial u(t)} + \langle \phi(t), \vec{b} \rangle = 0, \quad 0 \leq t \leq T,$$

for cost functionals of the Lagrange type.

PMP conditions for functionals: linear control

- For linear control systems, the PMP demands that

$$\frac{\partial \mathbf{H}}{\partial u(t)} = \lambda \frac{\partial L(u(t))}{\partial u(t)} + \langle \phi(t), \vec{b} \rangle = 0, \quad 0 \le t \le T,$$

for cost functionals of the Lagrange type.

- For cost functionals of the Bolza type, we have

$$\frac{\partial \mathbf{H}_B}{\partial u(t)} = \lambda \frac{\partial L(u(t))}{\partial u(t)} + \langle \phi(t), \vec{b} \rangle = 0$$

for linear control systems, with the boundary condition $\phi(T) = \nabla_{x(T)} F(x(T))$ on the costate imposed.

PMP conditions for functionals: linear control

- For linear control systems, the PMP demands that

$$\frac{\partial \mathbf{H}}{\partial u(t)} = \lambda \frac{\partial L(u(t))}{\partial u(t)} + \langle \phi(t), \vec{b} \rangle = 0, \quad 0 \leq t \leq T,$$

  for cost functionals of the Lagrange type.

- For cost functionals of the Bolza type, we have

$$\frac{\partial \mathbf{H}_B}{\partial u(t)} = \lambda \frac{\partial L(u(t))}{\partial u(t)} + \langle \phi(t), \vec{b} \rangle = 0$$

  for linear control systems, with the boundary condition $\phi(T) = \nabla_{x(T)} F(x(T))$ on the costate imposed.
- For cost functionals of the Mayer type, we have

$$\frac{\partial \mathbf{H}_B}{\partial u(t)} = \langle \phi(t), \vec{b} \rangle = 0$$

  for linear control systems, with the boundary condition $\phi(T) = \nabla_{x(T)} F(x(T))$ on the costate imposed.

Outline

Legendre conditions for optimality

Satisfaction of the first-order conditions following from the PMP is a necessary but not sufficient condition for optimality of a control $\varepsilon(\cdot)$. So-called *Legendre* conditions on the Hessian $\frac{\partial^2 \mathbf{H}}{\partial u(t)\partial u(t')}$, which depend on the type of cost, are also required for optimality. These are discussed further in the next lecture.

Analytic solutions to OCT problems

Lagrange and Bolza costs

CHE 597 - Quantum Control Engineering - Spring 2010

Purdue University

Feb 10, 2010

Outline

Outline

Solutions sets to Lagrange control problems

- Denote the space of admissible controls $\varepsilon(\cdot)$ by $\mathbb{K}$. Recall that the condition for optimality of quantum controls for Lagrange costs (on $\mathcal{U}(N)$) was

$$\frac{\partial \mathbf{H}}{\partial \varepsilon(t)} = \lambda \frac{\partial L(\varepsilon(t))}{\partial \varepsilon(t)} - \frac{i}{\hbar} \mathrm{Tr} \left(U^\dagger(T) \phi(T) U^\dagger(t) \mu U(t) \right) = 0, \;\; 0 \leq t \leq T$$

- Imposition of an endpoint constraint on the state (for Lagrange functionals) places restrictions on the matrix $\phi(T)$ and hence restricts admissible optimal controls to a subspace $\mathcal{S}_L \subset \mathbb{K}$. A unique optimal control is then specified.

Solutions sets to Mayer and Bolza control problems

- For Bolza-type functionals, the PMP can explicitly specify a unique optimal control $\bar{\varepsilon}(\cdot) \in \mathbb{K}$ in the absence of an endpoint constraint, since it may be possible to solve for $\bar{\varepsilon}(\cdot)$ when $\phi(T) = \nabla F(x(T)) \neq 0$; a unique control is specified there is a unique state that maximizes $F(x)$.
- For Mayer-type cost functionals, the PMP condition defines a submanifold $\mathcal{S}_M \subset \mathbb{K}$ of *co*dimension equal to the number of constraints present in the condition $\nabla F(x(T)) = 0$ (e.g., N^2, $N^2 - 1$, or 1 for unitary propagator, density matrix or observable control, respectively).

We will focus on analytical solutions to OCT problems with Bolza costs or Lagrange costs with a terminal constraint, because a unique optimal control exists for these problems.

Types of performance indices (Lagrange cost functions)

The type of Lagrange cost function plays an important role in determining the solution strategy and characteristics of closed form optimal control solutions.

- A *linear cost function* can be expressed in the general form $\int_0^T c^T x(t)\, dt$

Types of performance indices (Lagrange cost functions)

The type of Lagrange cost function plays an important role in determining the solution strategy and characteristics of closed form optimal control solutions.

- A *linear cost function* can be expressed in the general form $\int_0^T c^T x(t)\ dt$
- A *quadratic cost function* can be expressed in the general form $\frac{1}{2}\int_0^T x^T(t)Qx(t)\ dt$ where Q is a (not necc positive-definite, but symmetric), i.e., as a quadratic form.

Outline

Solving OCT problems

- Solving optimal control problems in closed form is hard because one must not only integrate systems of coupled differential equations but

Solving OCT problems

- Solving optimal control problems in closed form is hard because one must not only integrate systems of coupled differential equations but
- The differential equations are expressed parametrically in terms of controls; one must simultaneously solve for the optimal values of these parameters.

Solving OCT problems

- Solving optimal control problems in closed form is hard because one must not only integrate systems of coupled differential equations but
- The differential equations are expressed parametrically in terms of controls; one must simultaneously solve for the optimal values of these parameters.
- The solution to a control problem (either the parametric form of the controls or the explicit function) is called the *control law*.

General steps for solving OCT problems

1. Find the adjoint equations for the control system.

General steps for solving OCT problems

1. Find the adjoint equations for the control system.
2. Express the control $u(t)$ in terms of the state $x(t)$ and the costate $\phi(t)$

General steps for solving OCT problems

1. Find the adjoint equations for the control system.
2. Express the control $u(t)$ in terms of the state $x(t)$ and the costate $\phi(t)$
3. If the adjoint equations are uncoupled to the dynamical equations, a) integrate them. Express undetermined integration constants in terms of $\phi(T)$. b) Insert this solution for $\phi(t)$ into the dynamical equations and solve.

General steps for solving OCT problems (cont)

1. If the adjoint equations are coupled to the dynamical equations, solve the system simultaneously (e.g., using Laplace transforms); again express integration constants in terms of $\phi(T)$ and the known initial conditions $x(0)$.

General steps for solving OCT problems (cont)

1. If the adjoint equations are coupled to the dynamical equations, solve the system simultaneously (e.g., using Laplace transforms); again express integration constants in terms of $\phi(T)$ and the known initial conditions $x(0)$.
2. *If the cost functional is Lagrange*, with an endpoint constraint on the state, use this constraint to obtain $\phi(T)$ and hence explicit solutions for $\phi(t), x(t)$.

General steps for solving OCT problems (cont)

1. If the adjoint equations are coupled to the dynamical equations, solve the system simultaneously (e.g., using Laplace transforms); again express integration constants in terms of $\phi(T)$ and the known initial conditions $x(0)$.
2. *If the cost functional is Lagrange*, with an endpoint constraint on the state, use this constraint to obtain $\phi(T)$ and hence explicit solutions for $\phi(t), x(t)$.
3. *If the cost functional is Bolza*, use $\phi(T) = \nabla F(x(T))$ to obtain a relation between $\phi(T)$ and $x(T)$; substitute this implicit expression for $\phi(T)$ into all equations to obtain explicit expressions for all constants and determine $x(t), \phi(t)$.

General steps for solving OCT problems (cont)

1. If the adjoint equations are coupled to the dynamical equations, solve the system simultaneously (e.g., using Laplace transforms); again express integration constants in terms of $\phi(T)$ and the known initial conditions $x(0)$.
2. *If the cost functional is Lagrange*, with an endpoint constraint on the state, use this constraint to obtain $\phi(T)$ and hence explicit solutions for $\phi(t), x(t)$.
3. *If the cost functional is Bolza*, use $\phi(T) = \nabla F(x(T))$ to obtain a relation between $\phi(T)$ and $x(T)$; substitute this implicit expression for $\phi(T)$ into all equations to obtain explicit expressions for all constants and determine $x(t), \phi(t)$.
4. Use the resulting explicit solutions for $x(t), \phi(t)$ in the equation for $u(t)$ to obtain the optimal control $\bar{u}(t)$.

Solving bilinear vs. linear control problems

- For bilinear (or nonlinear) control systems, the ode's resulting from insertion of $u(t)$ in terms of $x(t), \phi(t)$ is *nonlinear*.

Solving bilinear vs. linear control problems

- For bilinear (or nonlinear) control systems, the ode's resulting from insertion of $u(t)$ in terms of $x(t), \phi(t)$ is *nonlinear*.
- Thus bilinearity of the control system leads to a nonlinear Schrödinger equation, and it is generally difficult or not possible to solve analytically for optimal controls.

Solving bilinear vs. linear control problems

- For bilinear (or nonlinear) control systems, the ode's resulting from insertion of $u(t)$ in terms of $x(t), \phi(t)$ is *nonlinear*.
- Thus bilinearity of the control system leads to a nonlinear Schrödinger equation, and it is generally difficult or not possible to solve analytically for optimal controls.
- However, for the simplest problems, alternate analytic solution strategies are possible.

Temperature control

The temperature in a room is denoted $y(t)$. It is desired to heat the room (to a target temperature) using the smallest possible amount of energy (heat). Let the ambient (external) temperature be denoted y_e. The rate of heat supply to the room is denoted $u(t)$. The dynamics of temperature change are then given by

$$\frac{dy}{dt} = -a(y(t) - y_e) + bu(t)$$

where a, b are constants depending on the insulation and rate of heat transfer. Let the total energy (heat) be given by $\frac{1}{2}\int_0^T u^2(t)\ dt$. We are given the initial temperature $x(0)$.

The problem: Calculate the control function $\bar{u}(t)$ that heats the room to temperature y_f at time T while minimizing the energy used, using two possible performance indices: a) $J = \frac{1}{2}\int_0^T u^2(t)\ dt$; b) $J = k[y(T) - y_f]^2 + \frac{1}{2}\int_0^T u^2(t)\ dt$ (i.e., the final temperature need not be precisely y_f).

Example: temperature control

- Let $x(t) = y(t) - y_e$ and $x_f = y_f - y_e$. If Lagrange,

$$J = \int_0^T L(t)\ dt$$

Problem is $\min_{u(t)} J$ subject to $\frac{dx}{dt} = -ax(t) + bu(t)$

1. $\frac{dx}{dt} = A\mathbf{x}(t) + B\mathbf{u}(t)$
2. $x(T) = x_f$

- If Bolza,

$$J = F(x(T)) + \int_0^T L(t)\ dt$$

$F(x(T)) = k[x(T) - x_f]^2$. $\min_{u(t)} J$ subject to

1. $\frac{dx}{dt} = A\mathbf{x}(t) + B\mathbf{u}(t)$
2. $\phi(T) = \nabla_{\mathbf{x}(T)} F(\mathbf{x}(T))$

Example: temperature control

- The PMP-Hamiltonian is:

$$\begin{aligned}\mathbf{H}(\mathbf{x}(t), \phi(t), \mathbf{u}(t)) &= \lambda L(\mathbf{x}(t), \mathbf{u}(t)) + \langle \phi(t), A\mathbf{x}(t) + B\mathbf{u}(t)\rangle \\ &= ku^2(t) - \phi(t)ax(t) + \phi(t)bu(t)\end{aligned}$$

- The adjoint variational equation is:

$$\begin{aligned}\frac{d\phi(t)}{dt} &= -\nabla_{\mathbf{x}}\mathbf{H}(\mathbf{x}(t), \phi(t), u(t)) \\ &= -\frac{\partial}{\partial x(t)}[-\phi(t)ax(t) + \phi(t)bu(t)] \\ &= \phi(t)a\end{aligned}$$

Example: temperature control (cont)

- Integrate above homogeneous 1st order ODE w const coeffs:

$$\frac{d\phi(t)}{dt} = \phi(t)a$$
$$\phi(t) = c\exp(at)$$

- Expressing c in terms of $\phi(T)$:

$$c = \exp(-aT)\phi(T)$$
$$\phi(t) = \exp[-a(T-t)]\phi(T)$$

Example: temperature control (cont)

$$\frac{\partial \mathbf{H}}{\partial u(t)} = \phi(t)b + \bar{u}(t) = 0, \quad 0 \leq t \leq T$$

or $\bar{u}(t) = -\phi(t)b$. Now, insert implicit expression for control (in terms of $\phi(t)$) into the dynamical equation of the variational system (1st E-L equation):

$$\begin{aligned}\frac{dx}{dt} &= -ax(t) - b^2\phi(t) \\ &= -ax(t) - b^2 \exp[-a(T-t)]\phi(T)\end{aligned}$$

Example: temperature control (cont)

The equation $\frac{dx}{dt} = -ax(t) - b^2 \exp[-a(T-t)]\phi(T)$ can be integrated analytically via Laplace transforms:

- The Laplace transform of $ax(t)$ is $aX(s)$
- The Laplace transform of $\exp(at)$ is $\frac{1}{s-a}$
- Laplace transform of $\frac{dx}{dt}$ is $sX(s) - x(0)$
- Thus, in the frequency domain,
  $X(s) = \frac{x(0)}{s+a} - b^2 \exp(-aT)\phi(T)\frac{1}{(s+a)(s-a)}$

- The inverse LT of $\frac{1}{s+a}$ ($\mathcal{L}^{-1}(\frac{1}{s+a})$) is $\exp(-at)$
- Partial fraction expansion of $\frac{1}{(s+a)(s-a)}$:

$$\frac{1}{(s+a)(s-a)} = \frac{\alpha_1}{s+a} + \frac{\alpha_2}{s-a}$$

$$1 = \alpha_1(s-a) + \alpha_2(s+a)$$

- Let $s = a$; then $\alpha_2 = \frac{1}{2a}$
- Let $s = -a$; then $\alpha_1 = -\frac{1}{2a}$
- $\frac{1}{(s+a)(s-a)} = \frac{1}{2a}\left[\frac{1}{s-a} - \frac{1}{s+a}\right]$

$$\begin{aligned}\mathcal{L}^{-1}\left[\frac{1}{(s+a)(s-a)}\right] &= \frac{1}{2a}\mathcal{L}^{-1}\left[\frac{1}{s-a} - \frac{1}{s+a}\right] \\ &= \frac{1}{2a}\left[\exp(at) - \exp(-at)\right] \\ &= \frac{1}{a}\sinh(at)\end{aligned}$$

- So

$$x(t) = x(0)\exp(-at) - \frac{b^2}{a}\exp(-aT)\phi(T)\sinh(aT)$$

- Note both optimal control and state trajectory expressed implicitly in terms of $\phi(T)$; need $\phi(T)$ to solve control problem
- Two ways to obtain $\phi(T)$:
  1. Lagrange cost: use endpoint constraint on state, i.e., $x(T) = x_f$ and solve for $\phi(T)$ from $x_f = x(0)\exp(-at) - \frac{b^2}{a}\exp(-aT)\phi(T)\sinh(aT)$
  2. Bolza cost: use boundary condition

$$
\begin{aligned}
\phi(T) &= \nabla_x F(x(T)) \\
&= 2k[x(T) - x_f]
\end{aligned}
$$

- Then, obtain optimal control $\bar{u}(t)$ by substituting the known value of $\phi(T)$ into the parametric expression for $\bar{u}(t)$:

$$
\begin{aligned}
\bar{u}(t) &= -\phi(t)b \\
&= \exp[a(T - t)]\phi(T)b \\
&= \exp[a(T - t)]2k[x(T) - x_f]b
\end{aligned}
$$

and finally insert into dynamical equation of variational system to obtain optimal trajectory (here, temperature of the room as a function of time).

Example: temperature control - Lagrange solution

- $x_f = x(0)\exp(-aT) - \frac{b^2}{a}\exp(-aT)\phi(T)\sinh(aT)$
- Solve for $\phi(T)$:

$$\phi(T) = (x_f - x(0)\exp(-aT))\,\frac{a}{b^2}\exp(aT)(\sinh(aT))^{-1}$$
$$= (x_f - x(0)\exp(-aT))\,\frac{2a}{b^2}(1+\exp(-2aT))^{-1}$$

- Insert into $\bar{u}(t)$ expression:

$$\bar{u}(t) = \exp[-a(T-t)]\phi(T)b$$
$$= \exp[-a(T-t)]\,(x_f - x(0)\exp(-aT))\,\frac{2a}{b}(1+\exp(-2aT))^{-1}$$

- Consider the case where the target temperature is $x_f = 10$ and the initial temperature is $x(0) = 0$;

$$\bar{u}(t) = \exp[-a(T-t)]\frac{20a}{b}(1+\exp(-2aT))^{-1}$$

- Now verify the solution by inserting the $\bar{u}(t)$ into $x(t)$ expression:

$$\bar{x}(t) = 10\frac{\sinh at}{\sinh aT}$$

Example: temperature control - Bolza solution

- For Bolza functionals, since $\phi(T) = \nabla_x F(x(T)) = 2k[x(T) - x_f]$
- Now solve for $x(T)$ using this expression (again assume $x(0) = 0$ and $x_f = 10$):

$$x(T) + \frac{b^2}{a}\exp(-aT)2kx(T)\sinh(at) = \frac{b^2}{a}\exp(-aT)2kx_f\sinh(aT)$$

$$x(T) = \frac{\frac{b^2}{a}\exp(-aT)2kx_f\sinh(aT)}{1 + 2k\frac{b^2}{a}\exp(-aT)\sinh(aT)}$$

$$= \frac{10b^2k\sinh(aT)}{a\exp(aT) + 2b^2k\sinh(aT)}$$

- Obtain optimal control $\bar{u}(t)$:

$$\bar{u}(t) = \exp[a(T-t)]\phi(T)b$$

$$= \exp[a(T-t)]2k[x(T) - x_f]b$$

$$= \exp[a(T-t)]2k[\frac{10b^2k\sinh(aT)}{a\exp(aT) + 2b^2k\sinh(aT)} - x_f]b$$

- Now verify the solution by inserting the $u(t)$.

Outline

Angular momentum

- Classical angular momentum: $\mathbf{L} = \mathbf{r} x \mathbf{p}$
- Quantum angular momentum: quantize by replacing $\mathbf{r}, \mathbf{p}$ by their quantum operator analogs:

$$L_x = -i\hbar(y\frac{\partial}{\partial z} - z\frac{\partial}{\partial y}),$$

  etc.
- To solve for eigenfunctions, necessary to switch to spherical coordinates; expression for Laplacian complicated, will not study
- Eigenvalues of $|\mathbf{L}|^2$ are $\hbar^2 l(l+1)$; of L_z are $\hbar m$, where $-l \leq m \leq l$

What is spin?

- Particles in quantum mechanics (including both electrons and nuclei) have an intrinsic property called *spin*, which is a form of angular momentum
- The *spin magnetic moment* (which we denote by μ_s) is proportional to the total spin $\mathbf{S}$
- Analogously to the dipole interaction with the electric field, the magnetic field-spin interaction energy is $-\mu_s \cdot \mathbf{B} = c\mathbf{S} \cdot \mathbf{B}$
- It is possible to manipulate nuclear spins in molecules without affecting the rotational, vibrational, or electronic states; thus we focus on nuclear spins
- $\sqrt{\langle|\mathbf{S}|^2\rangle}$ is the expectation value of the norm of the total spin angular momentum of the particle

Pauli spin operators

- Observables corresponding to the x, y and z components of particle spin are S_x, S_y, S_z:

$$S_x = \frac{\hbar}{2}\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \ S_y = \frac{\hbar}{2}\begin{bmatrix} 0 & -\imath \\ \imath & 0 \end{bmatrix}, \ S_z = \frac{\hbar}{2}\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

- Eigenvalues are $\frac{\hbar}{2}, \ -\frac{\hbar}{2}$ ("spin-1/2" particles)
- Commutation relations are: $[S_i, S_j] = i\hbar\epsilon_{ijk}S_k$ where ϵ_{ijk} is a completely antisymmetric tensor
- These are called the the fundamental commutation relations of angular momentum and are satisfied by any form of angular momentum

Tensor products of Hilbert spaces: vectors

- The *tensor product* (or direct product) of Hilbert spaces is denoted $\mathcal{H}^1 \otimes \mathcal{H}^2$; its dimension is $m_1 m_2$, where m_1, m_2 are the dimensions of $\mathcal{H}^1, \mathcal{H}^2$, respectively (since there are $m_1 m_2$ possible joint states)
- Consider the matrix representation of a vector in this product space: it is denoted $|\psi\rangle \otimes |\phi\rangle$, where $\otimes$ now refers to the vector *Kronecker product*
- Let $|\psi_i\rangle$ $(i = 1, \cdots, m)$ denote the basis vectors of $|\psi\rangle$ and $|\phi_j\rangle$ $(j = 1, \cdots, n)$ denote those of $|\phi\rangle$. The Kronecker product of column vectors $|\psi\rangle$, $|\phi\rangle$ has as basis vectors $|(\psi \otimes \phi)_{ni+j+1}\rangle = |\psi_i, \phi_j\rangle$. (Note this is different from the outer (tensor) product of the vectors.)

Tensor products of Hilbert spaces: operators

- The same principle holds for tensor products of the sets of operators acting on $\mathcal{H}^1$, $\mathcal{H}^2$ (i.e., $B(\mathcal{H}^1),\ B(\mathcal{H}^2)$)
- The Kronecker product of (order mxm, nxn) matrices A, B, denoted $A \otimes B \equiv C$, has the form

$$\begin{bmatrix} a_{11}B & \cdots & a_{1m}B \\ \vdots & \ddots & \\ a_{m1}B & \cdots & a_{mm}B \end{bmatrix}$$

- In particular, an operator A in $B(\mathcal{H}^1)$ has representation $A \otimes I_n$ (Kronecker product) on $\mathcal{H}^1 \otimes \mathcal{H}^2$ (direct product)
- One may also have tensor products of finite-dimensional and infinite-dimensional Hilbert spaces

Schrödinger equation for single spin in time-varying xy magnetic field

- $H_1 = cS_x B_x(t) + cS_y B_y(t)$
- The static magnetic field is much stronger than the time-varying one

$$\begin{aligned} \frac{d}{dt}\psi(t) &= -\frac{i}{\hbar}\left[c\mathbf{B}(t)\cdot\mathbf{S}\right]\psi(t) \\ &= -\frac{i}{\hbar}\left[cS_z B_z + cS_x B_x(t) + cS_y B_y(t)\right]\psi(t) \end{aligned}$$

Let $c = 1$ for convenience, and use standard notation $\varepsilon_x(t) \equiv B_x(t)$, $\varepsilon_y(t) \equiv B_y(t)$ for controls. Note we now have vector of controls as well as state vector (both 2-dimensional in this case).

Example: quantum state control of a single spin with minimal energy

- For Lagrange type cost functionals with bilinear qc systems, the special case of a quadratic cost on the controls is worth attention because of its interpretation in terms of the total fluence of the field. Consider case with two controls $\varepsilon_x(t), \varepsilon_y(t)$.

The problem

Find the time-varying fields $\varepsilon_x(t)$ and $\varepsilon_y(t)$ that drive the system to a specified final state ψ_f at time T using minimal energy. The dynamical equation is

$$\begin{aligned} \frac{d}{dt}|\psi(t)\rangle =& -\frac{i}{\hbar}\mathbf{S}\cdot\mathbf{B}(t)|\psi(t)\rangle \\ & -\frac{i}{\hbar}\left[S_z B_z + S_x\varepsilon_x(t) + S_y\varepsilon_y(t)\right]\psi(t)\rangle \end{aligned}$$

Example: quantum spin state control (cont)

- The cost functional is $L(\varepsilon_x(t), \varepsilon_y(t)) = \frac{1}{2}(\varepsilon_x^2(t) + \varepsilon_y^2(t))$, Let $F(\psi(T)) = \Re\langle\psi_f|\psi(t)\rangle$. We want $F(\psi(T)) = 1$, i.e., achieve $\psi(T) = \psi_f$ within a global phase.
- Lagrange formulation:

$$J = \frac{1}{2}\int_0^T \varepsilon_x^2(t) + \varepsilon_y^2(t)\ dt$$

$$\mathbf{H}(\psi(t), \phi(t), \vec{\varepsilon}(t)) = \frac{1}{2}(\varepsilon_x^2(t) + \varepsilon_y^2(t)) - \langle\phi(t)|\frac{i}{\hbar}[S_z B_z + S_x\varepsilon_x(t) + S_y\varepsilon_y(t)|\psi(t)\rangle$$

Example: quantum spin state control (cont)

The costate equation is

$$\begin{aligned}\frac{d\phi^\dagger(t)}{dt} &= -\nabla_{\psi(t)}\mathbf{H}(\psi(t),\phi(t),\varepsilon(t)) \\ &= \frac{i}{\hbar}\phi^\dagger(t)\left[S_z B_z + S_x\varepsilon_x(t) + S_y\varepsilon_y(t)\right]\end{aligned}$$

or

$$\frac{d\phi(t)}{dt} = -\frac{i}{\hbar}\left[S_z B_z + S_x\varepsilon_x(t) + S_y\varepsilon_y(t)\right]\phi(t)$$

Example: quantum spin state control (cont)

The PMP demands

$$\nabla_{\vec{\varepsilon}(t)} \mathbf{H}(\psi(t), \phi(t), \vec{\varepsilon}(t)) \equiv 0$$

or

$$\varepsilon_x(t) = \frac{i}{\hbar}\langle \phi(t)|S_x|\psi(t)\rangle$$
$$\varepsilon_y(t) = \frac{i}{\hbar}\langle \phi(t)|S_y|\psi(t)\rangle$$

Now we could insert these into the Schrödinger equation and solve but note that the resulting ode is nonlinear. Instead, we apply additional conditions following from the PMP. The above equations imply

$$\frac{d\varepsilon_x(t)}{dt} = \frac{i}{\hbar}\{\langle \frac{d}{dt}\phi(t)|S_x|\psi(t)\rangle + \langle \phi(t)|S_x|\frac{d}{dt}\psi(t)\rangle\}$$
$$\frac{d\varepsilon_y(t)}{dt} = \frac{i}{\hbar}\{\langle \frac{d}{dt}\phi(t)|S_y|\psi(t)\rangle + \langle \phi(t)|S_y|\frac{d}{dt}\psi(t)\rangle\}$$

Example: quantum spin state control (cont)

$$\langle \frac{d}{dt}\phi(t)|S_y|\psi(t)\rangle = \frac{i}{\hbar}\phi^\dagger(t)\left[S_z B_z + S_x \varepsilon_x(t) + S_y \varepsilon_y(t)\right] S_y \psi(t)$$

$$\langle \phi(t)|S_y|\frac{d}{dt}\psi(t)\rangle = -\frac{i}{\hbar}\phi^\dagger(t) S_y \left[S_z B_z + S_x u_x(t) + S_y u_y(t)\right]\psi(t)$$

Recall Pauli commutation relations: $[S_i, S_j] = i\hbar\epsilon_{ijk}S_k$ where ϵ_{ijk} denotes the elements of a completely antisymmetric tensor. So

$$\begin{aligned}\dot{\varepsilon}_x(t) &= -(\frac{i}{\hbar})^2(\phi^\dagger(t)[S_x, S_y]\varepsilon_y(t)\psi(t) + \phi^\dagger(t)[S_x, S_z]\psi(t)B_z) \\ &= -\frac{i}{\hbar}^2(i\phi^\dagger(t)S_z\psi(t)\varepsilon_y(t) - i\phi^\dagger(t)S_y\psi(t)B_z)\end{aligned}$$

$$\begin{aligned}\dot{\varepsilon}_y(t) &= -(\frac{i}{\hbar})^2(\phi^\dagger(t)[S_y, S_x]\varepsilon_x(t)\psi(t) + \phi^\dagger(t)[S_y, S_z]\psi(t)B_z) \\ &= \frac{i}{\hbar}^2(i\phi^\dagger(t)S_z\psi(t)\varepsilon_x(t) - i\phi^\dagger(t)S_x\psi(t)B_z)\end{aligned}$$

Additional conserved quantities: expectation value of S_z

- Note that we still have expressions for $\varepsilon_x(t), \varepsilon_y(t)$ that are implicit functions of the state and costate.

Additional conserved quantities: expectation value of S_z

- Note that we still have expressions for $\varepsilon_x(t), \varepsilon_y(t)$ that are implicit functions of the state and costate.
- According to the condition $\frac{d}{dt}\left(\frac{\partial \mathbf{H}}{\partial \varepsilon(t)}\right) = 0$, there are additional conserved quantities (as long as $\frac{\partial \mathbf{H}}{\partial \varepsilon(t)}$ is not an explicit function of time). These can help us solve the Lagrange control problem analytically if the dimension of the system is sufficiently small, so that the additionally conserved quantities provide enough additional conditions to fully specify the optimal control.

Additional conserved quantities: expectation value of S_z

- Note that we still have expressions for $\varepsilon_x(t), \varepsilon_y(t)$ that are implicit functions of the state and costate.
- According to the condition $\frac{d}{dt}\left(\frac{\partial \mathbf{H}}{\partial \varepsilon(t)}\right) = 0$, there are additional conserved quantities (as long as $\frac{\partial \mathbf{H}}{\partial \varepsilon(t)}$ is not an explicit function of time). These can help us solve the Lagrange control problem analytically if the dimension of the system is sufficiently small, so that the additionally conserved quantities provide enough additional conditions to fully specify the optimal control.
- In the present case we have $\frac{\partial}{\partial B_z}\mathbf{H}(\psi(t), \phi(t), \varepsilon(t)) = 0$ and hence $\frac{d}{dt}\left[\frac{\partial}{\partial B_z}\mathbf{H}(\psi(t), \phi(t), \varepsilon(t))\right] = 0$ giving us the conserved quantity $\langle \phi(t) | S_z | \psi(t) \rangle = K$.

Example: quantum spin state control (cont)

Applying the latter conservation law in the equations for $\dot{\varepsilon}_x(t)$ and $\dot{\varepsilon}_y(t)$, and recalling that $\frac{i}{\hbar}\phi^\dagger(t)S_y\psi(t) = \varepsilon_y(t)$ (similarly for ε_x), we obtain the coupled system of first order ODEs

$$\dot{\varepsilon}_x(t) = -(K - B_z)\varepsilon_y(t)$$
$$\dot{\varepsilon}_y(t) = (K - B_z)\varepsilon_x(t)$$

which has (parametric) solutions

$$\varepsilon_x(t) = C\cos(\omega t + \alpha)$$
$$\varepsilon_y(t) = C\sin(\omega t + \alpha)$$

Example: quantum spin state control (cont)

- Next step: Solve for C (field's temporal amplitude scale), ω (field frequency), and α (field phase) given endpoint constraint, ψ_0 (normalization is implicit in these conditions)
- Need to insert parametric solns into dynamical or costate equations and *explicitly integrate*.

$$\begin{aligned}\frac{d}{dt}|\psi(t)\rangle &= -\frac{i}{\hbar}\left[S_z B_z + CS_x\cos(\omega t+\alpha) + CS_y\sin(\omega t+\alpha)\right]|\psi(t)\rangle \\ &= -\frac{i}{\hbar}\begin{pmatrix} B_z & C\exp[-i(\omega t+\alpha)] \\ C\exp[i(\omega t+\alpha)] & -B_z\end{pmatrix}|\psi(t)\rangle\end{aligned}$$

subject to $\psi(T)=\psi_f$ (two conditions), $\psi(0)=\psi_0$ (one additional condition) In the homework, we will solve for $\psi(t)$ in 1st-order perturbation theory.

- Note the system of dynamical odes is coupled due to norm constraint on ψ

Outline

1. Solution sets to optimal control problems
2. Analytic solutions: general guidelines
   - Strategies for solving optimal control problems
   - An example linear system
3. Analytic solutions to quantum control problems
   - Nuclear magnetic resonance
   - Quantum control with quadratic cost: fluence minimization
4. The need for numerical methods

The need for numerical optimization

- Because analytical solutions to OCT problems typically require the presence of multiple symmetries, numerical methods are required to solve most problems. The best method depends on the control system and cost functional

The need for numerical optimization

- Because analytical solutions to OCT problems typically require the presence of multiple symmetries, numerical methods are required to solve most problems. The best method depends on the control system and cost functional
- For a Bolza-type cost functional, the PMP optimality condition can be solved implicitly for the control field, $\varepsilon(t) = \frac{i}{\hbar}\mathrm{Tr}\left(U^\dagger(T)\nabla_{U(T)}F(U(T))U(^\dagger t)\mu U(t)\right)$, as in the case of Lagrange costs.

The need for numerical optimization

- Because analytical solutions to OCT problems typically require the presence of multiple symmetries, numerical methods are required to solve most problems. The best method depends on the control system and cost functional
- For a Bolza-type cost functional, the PMP optimality condition can be solved implicitly for the control field, $\varepsilon(t) = \frac{i}{\hbar}\mathrm{Tr}\left(U^\dagger(T)\nabla_{U(T)}F(U(T))U(^\dagger t)\mu U(t)\right)$, as in the case of Lagrange costs.
- However, since $\Phi(T)$ is no longer free and depends on the final time propagator $U(T)$, integration of the Schrödinger equation with this implicit expression for the field is not possible and numerical optimization is needed.

The need for numerical optimization

- Because analytical solutions to OCT problems typically require the presence of multiple symmetries, numerical methods are required to solve most problems. The best method depends on the control system and cost functional
- For a Bolza-type cost functional, the PMP optimality condition can be solved implicitly for the control field, $\varepsilon(t) = \frac{i}{\hbar}\mathrm{Tr}\left(U^\dagger(T)\nabla_{U(T)}F(U(T))U(^\dagger t)\mu U(t)\right)$, as in the case of Lagrange costs.
- However, since $\Phi(T)$ is no longer free and depends on the final time propagator $U(T)$, integration of the Schrödinger equation with this implicit expression for the field is not possible and numerical optimization is needed.
- More generally, even for Lagrange costs, if the dynamical and adjoint (systems of) ODEs do not afford analytical solutions (recall we solved only special simple cases above), numerical optimization is needed.

Optimal feedback control

CHE 597

Purdue University

April 5, 2010

Outline

1. Feedback control of time-invariant linear systems
   - Kalman gain

2. Lyapunov equations
   - Riccati equations
   - Analytic solution to algebraic Riccati equation

3. Dynamic programming

Outline

1. Feedback control of time-invariant linear systems
   - Kalman gain
2. Lyapunov equations
   - Riccati equations
   - Analytic solution to algebraic Riccati equation
3. Dynamic programming

Feedback control with Bolza functionals

- Now consider linear system with nonzero $u(t)$ and cost $J = \int_0^T x^T(t)Qx(t) + u^T(t)Ru(t)\ dt + \frac{1}{2}x^T(T)S(T)x(T)$ (i.e., unlike the Lagrange functional with endpoint state constraint used for controllability analysis, use a Bolza functional with the Lagrange term also containing quadratic cost on $x(t)$). The final state is thus not constrained; and the cost on the state will enable us to formulate time-varying feedback control
- By appropriately large weighting of $\frac{1}{2}x^T(T)S(T)x(T)$, can drive $x(T)$ arbitrarily close to desired endpoint, while executing feedback along trajectory, if system controllable
- With $T \to \infty$, control system is called linear quadratic regulator (LQR)
- In these deterministic feedback control problems, we do not update state estimates with observations; we assume the state is can be directly measured at any time t; later we will discuss linear quadratic Gaussian regulator (LQG), which is stochastic feedback control problem where state must be estimated

Kalman gain

- The PMP-Hamiltonian system is:

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{d\phi}{dt} \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \begin{bmatrix} x(t) \\ \phi(t) \end{bmatrix}$$

  with the $2N \times 2N$ matrix denoted the PMP-Hamiltonian matrix H
- Recall: to assess controllability, let $-Q = 0$; to assess observability, let $-BR^{-1}B^T = 0$
- Generalizing the scalar solution, implicitly, optimal control is $\bar{u}(t) = -R^{-1}B^T S(t)x(t)$, where we have made the linear ansatz $\phi(t) = S(t)x(t)$

$$\begin{aligned} \frac{dx}{dt} &= Ax(t) - BR^{-1}B^T S(t)x(t) \\ &= (A - BK(t))x(t) \end{aligned}$$

- $K(t) = R^{-1}B^T S(t)$ is called the Kalman gain; it provides (time-varying) state-dependent feedback to the control
- To solve the problem, we need to find the matrix function $S(t)$; we will later show that $S(T)$ is the same as that which appears in the cost functional

Outline

Asymptotic convergence, Lyapunov functions

- Consider the deviation variable (error residual) $\tilde{x}(t) = x(t) - \bar{x}$, where $\bar{x}$ denotes the fixed point ($\dot{\bar{x}} = 0$)
- For a linear system, $\frac{d}{dt}\tilde{x}(t) = A\tilde{x}(t)$
- Consider the cost function $J(\tilde{x}) = \frac{1}{2}\tilde{x}^T S\tilde{x}$ with $S = D^T D$ (symmetric, positive definite)
- If $\dot{J}(\tilde{x})$ decreases monotonically in the vicinity of a fixed point (converging to the unique value), it is said to be a Lyapunov function and the neighborhood is said to be stable (for linear system, the system is stable); this definition holds for more general functions than the one above
- Then, if $\int_0^\infty \tilde{x}^T(t)S\tilde{x}(t)\,dt$ is bounded, the linear(ized) system is said to be exponentially asymptotically stable (for a linear system, globally stable) Occurs if A has only negative real parts to all its eigenvalues.
- Exponential convergence (stability):

$$||\tilde{x}(t)|| = ||\exp(At)\tilde{x}(0)||$$
$$||\tilde{x}(t)|| \leq k\exp(-\lambda_i t)\,||\tilde{x}(0)||\,,$$

  where λ_i denotes the smallest (in absolute value) real part of an eigenvalue of A

Lyapunov equations

- A (differential) Lyapunov equation with Lyapunov function $J(x) = \frac{1}{2}x^T S(t)x$ is of the form $\dot{S}(t) = S(t)A + A^T S(t) + T$, where T is positive definite; solve with either $S(0)$ or $S(T)$ given
- Soln to diff Lyapunov equation converges to constant matrix S if the system is asymp. stable. There, $\dot{S} = 0$
- An algebraic Lyapunov equation is derived from steady state condition $\dot{S} = 0$; it is the resulting Lyapunov equation with $S = S(0)$ (for a backwards integrated differential Lyapunov equation)
- To see the origin of the (algebraic) Lyapunov equation, compute $\dot{J}(x(t))$ for a linear dynamical system:

$$
\begin{aligned}
\dot{J}(x(t)) &= \frac{1}{2}\dot{x}^T Sx \\
&= x^T S\dot{x} \\
&= x^T SAx \\
&= x^T (SA + A^T S)x
\end{aligned}
$$

where the last line follows since the scalar $(x^T SAx)^T = x^T A^T Sx$. For $\dot{J}(x) < 0$, must have $SA + A^T S$ negative definite

Lyapunov equations in optimal control

- In either case, solve for $S(0)$ or $S(t)$; solve algebraic Lyapunov equation to obtain steady-state (asymptotic) cost and steady-state feedback gain (latter through a minor variation called Riccati eqn)
- Time-invariant control strategies (i.e., $u(t) = c$, a constant) often chosen to stabilize otherwise unstable dynamical systems; are based on steady-state gain
- Optimal feedback control strategies $u(x(t))$, discussed below, are based on appropriate choice of cost function, esp Lagrange term $L(x(t)) = \frac{1}{2}x^T(t)Qx(t) + \frac{1}{2}u^T(t)Ru(t)$, through choice of weighting matrices Q and R

Riccati equation

$$
\begin{aligned}
\frac{d}{dt}\phi(t) &= \frac{d}{dt}\left[S(t)x(t)\right] = \dot{S}(t)x(t) + S(t)\dot{x}(t) \\
&= \dot{S}(t)x(t) + S(t)(Ax(t) + Bu(t)) \\
-Qx - A^T\phi(t) &= \dot{S}(t)x(t) + S(t)(Ax(t) - BR^{-1}B^T\phi(t)) \\
-Qx - A^TS(t)x(t) &= \dot{S}(t)x(t) + S(t)(Ax(t) - BR^{-1}B^TS(t)x(t)) \\
\dot{S}(t)x(t) &= (A^TS(t) - S(t)A + S(t)BR^{-1}B^TS(t) - Q)x(t) \\
\dot{S}(t) &= -A^TS(t) - S(t)A + S(t)BR^{-1}B^TS(t) - Q
\end{aligned}
$$

- For this system, the optimal feedback gain is time varying: $K(t) = R^{-1}B^TS(t)$; to obtain, must solve Riccati equation for $S(t)$

Riccati equation (cont)

- Riccati equation is propagated backwards in time (since $S(T)$ specified); asymptotic limit is $T - t \to \infty$ (assume $T \to \infty$, then can set $t = 0$)
- Formal solution possible, but requires solution of complete PMP-Hamiltonian linear system, as in case of temperature control problem in HW 2; this is due to coupling (presence) of $x(t), \phi(t)$ in both state, costate odes: revisit later
- Solution $S(0)$ (by backwards integration) to (differential) Riccati equation with boundary condition $\lim_{T \to \infty} S(t)$ is a constant. The corresponding algebraic Riccati equation $-A^T S(0) - S(0)A + S(0)BR^{-1}B^T S(0) - Q = 0$ is solved for $S(0)$. Note that with $S(0)BR^{-1}B^T S(0) - Q$ positive-definite this satisfies the conditions for a algebraic -Riccati- equation
- The corresponding feedback gain is called the steady-state feedback gain; linear systems are stable with it, as long as systems are controllable

Observability Lyapunov equation

- With only $\frac{1}{2}x^T Qx$ term in J, $\dot{S}$ ode is called observability Lyapunov equation

$$\frac{d}{dt}\phi(t) = \frac{d}{dt}\left[S(t)x(t)\right] = \dot{S}(t)x(t) + S(t)\dot{x}(t)$$
$$-A^T\phi(t) - Qx = \dot{S}(t)x(t) + S(t)Ax(t)$$
$$-A^T S(t)x(t) - Qx = \dot{S}(t)x(t) + S(t)Ax(t)$$
$$\dot{S}(t) = -S(t)A - A^T S(t) - Q$$

- $\phi(t) = S(t)x(t)$ again, but solution $S(t)$ differs from LQR
- Can be formally integrated in closed form analogously to $x(t)$ for time-invariant linear system

$$S(t) = \exp[A^T(T-t)]S(T)\exp[A(T-t)] + \\ + \int_0^T \exp[A^T(T-t)]Q\exp[A(T-t)]\,dt$$

Controllability Lyapunov equation

- With only $u^T Ru$ term in J, the corresponding matrix ode is called controllability Lyapunov equation
- Controllability Lyapunov equation not expressed in terms of S, rather

$$\dot{P}(t) = P(t)A + A^T P(t) + BR^{-1}B^T$$

- Formal solution similar:

$$P(t) = \exp[At]P(0)\exp[A^T t] + \int_0^T \exp[At]BR^{-1}B^T \exp[A^T t]\ dt$$

  (unlike Ricatti and observability Lyapunov equations, propagated forward in time)
- By using $P(0) = 0$, solution for $P(t)$ provides controllability Gramian: may enable simple solution of linear, quadratic control cost problems
- Next time will discuss stabilizability, which involves choosing a (feedback) control strategy that causes the system to converge asymptotically to a fixed point. In so doing we will discuss the relationship between the optimal control time-domain and frequency domain control formulations (latter typically not optimal)

Cost "to-go"

- The cost to-go $J(t)$ is the cost incurred over the trajectory portion $[t, T]$; minimized over the remaining trajectory, irrespective of the prior trajectory, in closed-loop feedback.
- Being a Lyapunov function, $J(t)$ must decrease monotonically over time (if the system is stable)
- Example: $J(t) = \frac{1}{2}x^T(t)S(t)x(t)$ for observability Lyapunov equation; check:

$$x^T(T)S(T)x(T) = \int_0^T \frac{d}{dt}(x^T(t)S(t)x(t))\ dt + x^T(0)S(0)x(0)$$

$$= \int_0^T \dot{x}^T(t)S(t)x(t) + x^T(t)\dot{S}(t)x(t) + x^T(t)S(t)\dot{x}(t)\ dt + x^T(0)S(0)x(0$$

$$= \int_0^T x^T(t)A^TS(t)x(t) + x^T(t)(-S(t)A - A^TS(t) - Q)x(t) +$$

$$+ x^T(t)S(t)Ax(t)\ dt + x^T(0)S(0)x(0)$$

$$x^T(0)S(0)x(0) = x^T(T)S(T)x(T) + \int_0^T x^T(t)Qx(t)\ dt$$

Cost "to-go" (cont)

- $J(t) = \frac{1}{2}x^T(t)S(t)x(t) + \int_0^T \left|\left|R^{-1}B^TQx - u(t)\right|\right|_R \, dt$ for Lyapunov equation with suboptimal feedback; $\frac{1}{2}x^T(t)S(t)x(t)\ dt$ for Riccati equation (optimal feedback)
- In both cases, J is a Lyapunov function; (since) $S(t)$ is positive definite and the Lyapunov condition is satisfied with positive-definite Q, $\dot{J}(t)$ is negative definite for all t; allows us to assess asymptotic stability through cost function alone

Hamiltonian matrices

- Our goal is to find steady-state optimal control $\bar{u}(x(t))$ such that system, if unstable, is stabilized. Need to solve the PMP-Hamiltonian system.
- Let

$$J = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix}$$

Note $J^{-1} = J^T = -J$.

- A *Hamiltonian matrix* H satisfies $JHJ = H^T$
- Any matrix of the form

$$\begin{bmatrix} A & B \\ C & -A^T \end{bmatrix}$$

where $B = B^T$, $C = C^T$, is a Hamiltonian matrix (verify).

Left/right eigenvalues and eigenvectors (of Hamiltonian matrices)

- Since Hamiltonian matrices are not symmetric, they can have complex eigenvalues
- They will also have left and right eigenvectors, each associated with the same set of complex eigenvalues
- A left eigenvector ω satisfies $\omega^T H = \alpha \omega^T$, where ω is a scalar
- Let $H\nu = \lambda\nu$, where λ is the eigenvalue associated with eigenvector ν. Then

$$\begin{aligned} \nu^T H^T &= \lambda \nu^T \\ \nu^T J^T H J^T &= \lambda \nu^T \\ -\nu^T J^T H &= -\lambda \nu^T J^T \\ (J\nu)^T H &= -\lambda (J\nu)^T \end{aligned}$$

  Thus $J\nu$ is a left eigenvector of H with eigenvalue $-\lambda$ (the eigenvalues thus come in pairs). (note we thus only have to solve for the right eigenvectors and automatically obtain the left).
- Note that for a general linear system $\dot{x} = Ax$, A will also have left/right eigenvectors and complex eigenvalues. The (open loop) system is (asymptotically) stable if all eigenvalues have negative real parts

Diagonalization of Hamiltonian matrices

- Based on the above result, the $2N \times 2N$ Hamiltonian matrix H, when diagonalized, should look like

$$\begin{bmatrix} \lambda_1 & & & & & \\ & \ddots & & & & \\ & & \lambda_N & & & \\ & & & -\lambda_1 & & \\ & & & & \ddots & \\ & & & & & -\lambda_N \end{bmatrix} \equiv \begin{bmatrix} M & \\ & -M \end{bmatrix}$$

with $\lambda_i \in \mathbb{C}$.

- Recall the PMP-Hamiltonian system of $2N$ odes was

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\phi}(t) \end{bmatrix} = H \begin{bmatrix} x(t) \\ \phi(t) \end{bmatrix}.$$

- Substitute the expression for H in terms of its eigenvalue matrix, $H = E\tilde{H}D^T$, where E denotes the matrix whose columns are right eigenvectors, and D the matrix whose columns are left eigenvectors

Solving for steady-state gain and optimal feedback control

- Now can solve this ode system with time-invariant Hamiltonian as

$$\begin{bmatrix} x(t) \\ \phi(t) \end{bmatrix} = \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix} \begin{bmatrix} \exp(M) & \\ & \exp(-M) \end{bmatrix} \begin{bmatrix} D_{11}^T & D_{21}^T \\ D_{12}^T & D_{22}^T \end{bmatrix} \begin{bmatrix} x(0) \\ Sx(0) \end{bmatrix}$$

- Since feedback controlled system stable,

$$D_{11}^T x(0) + D_{21}^T Sx(0) = 0$$

  for all $x(0)$ so that unstable modes do not contribute to the dynamics; otherwise, $x(t)$ will diverge as $t \to \infty$

- Solving for S,

$$S = (D_{21}^T)^{-1} D_{11}^T$$

- Thus the steady-state feedback gain is

$$K(\infty) = R^{-1}B^T S = R^{-1}B^T (D_{21}^T)^{-1} D_{11}^T$$

  and the optimal steady-state control is
  $u(x(t)) = -K(\infty)x(t) = -R^{-1}B^T(D_{21}^T)^{-1}D_{11}^T x(t)$

Stabilization of the closed loop system

- Assuming the system is controllable (depends on A, B) and Q, R positive definite, the closed loop system with steady-state optimal feedback is stable irrespective of how many modes (eigenvectors) of A are unstable.
- The associated steady-state closed loop matrix $A^{cl} = A - BK(\infty)$ has N stable eigenvalues, which happen to be the N stable eigenvalues $-\lambda_1, \cdots, -\lambda_N$ of the Hamiltonian matrix H. The eigenvectors of $A - BK(\infty)$ are the columns of the matrix E_{11}.
- Recall the definition of the open loop transfer function for a single input (control), single output (observation) system:

$$\frac{y(s)}{u(s)} = C(sI - A)^{-1}B$$

where C is $1 \times N$ and B is $N \times 1$

Stabilization of the closed loop system (cont)

- The characteristic polynomial of the open loop matrix A is given by the determinant $|sI - A|$; solve for the *poles* of the open loop transfer function
- The feedback stabilized system, the poles vary as a function of the elements of Q and R in the cost functional; the plot of the poles versus these parameters is analogous to the *root locus* plot in frequency domain control, where the poles are plotted versus constant gain parameters to design the controller The closed loop characteristic polynomial is

  $$|sI - A + BK(\infty)|$$

  whose roots all have negative real parts (reside on left half complex plane).
- With time-varying state feedback, the poles change over time

Outline

Dynamic programming formulation of optimal feedback control

- Except for linear feedback control, methods we have studied based on PMP not suitable for feedback control since they provide "open loop" optimal controls and trajectories based on known initial state x_0; for linear systems, our ansatz $\phi(t) = S(t)x(t)$ was essential for obtaining state feedback
- PMP max/minimizes $J(0)$
- Cost-to-go $J(t)$ does not directly enter PMP formulation; useful to formulate general nonlinear optimal feedback control law in terms of cost-to-go
- Make J a function of x, u, t instead of just u as in original PMP formulation
- Recall $\mathbf{H} = \mathbf{H}(x, \phi, u, t)$
- By adding x, t parameters to J, will see we can express Lagrange multiplier $\phi(t)$ as partial derivative $\frac{\partial J(x,t)}{\partial x}$; note this is function of t like $\phi(t)$

Hamilton-Jacobi-Bellman equation

- Cost-to-go is now expressed as $J(x, u, t)$ rather than $J(u, t)$; let
$$J(x, u, t) = F(x(T), T) + \int_t^T L(x(t'), u(t'), t')\ dt'$$
This is fn of x through x_t
- Then
$$\frac{dJ(x,t)}{dt} = -L(x(t), u(t), t)$$
- For any control and associated trajectory,
$$\begin{aligned}\frac{dJ(x,t)}{dt} &= \frac{\partial J(x,t)}{\partial t} + \frac{\partial J(x,t)}{\partial x}\frac{dx}{dt} \\ &= \frac{\partial J(x,t)}{\partial t} + \frac{\partial J(x,t)}{\partial x} f(x,u,t) \quad = -L(x(t), u(t), t)\end{aligned}$$
$$\frac{\partial J(x,t)}{\partial t} = -L(x(t), u(t), t)) - \frac{\partial J(x,t)}{\partial x} f(x,u,t)$$
- Hamiltonian now defined as
$\mathbf{H}(x, \frac{\partial J(x,t)}{\partial x}, u, t) = L(x(t), u(t), t) + \frac{\partial J(x,t)}{\partial x} f(x, u, t)$ instead of
$\mathbf{H}(x, \phi, u, t) = L(x(t), u(t), t) + \phi^T(t) f(x, u, t)$
- For optimal trajectory,
$\mathbf{H}(\bar{x}(t), \frac{\partial J(x,t)}{\partial x}, \bar{u}(t), t) = \min_{u(t)} \mathbf{H}(x(t), \frac{\partial J(x,t)}{\partial x}, u(t), t)$ as before

Hamilton-Jacobi-Bellman equation (cont)

- So HJB equation is

$$\frac{\partial J^*(x,t)}{\partial t} = -\min_{u(t)} \mathbf{H}(x(t), \frac{\partial J^*(x,t)}{\partial x(t)}, u(t), t)$$

  where J^* denotes the optimal cost-to-go, which we will denote by simply J

- Partial differential equation for $J(x,t)$; propagated backward in time (since t is lower limit of Lagrange integral) from $J(x(T), T) = F(x(T), T)$ (at all pts on surface of admissible final states $x(T)$)
- Note equivalence between costate $\phi(t)$ and $\frac{\partial J(x,t)}{\partial x}$
- Solve for vector field of extremals $\bar{u}(x,t)$ rather than a single optimal control $\bar{u}(t)$; vector field of extremals sometimes called optimal policy (since control conditional on x)
- Note x_0 not explicitly specified

Optimal control solution strategy using HJB equation

- Follow these steps:
  1. Set up Hamiltonian as for PMP but with $\frac{\partial J(x,t)}{dx}$ replacing $\phi^T(t)$
  2. Use PMP condition $\frac{\partial \mathbf{H}}{\partial u(t)} = 0$ to express $\bar{u}(t)$ in terms of $\frac{\partial J(x,t)}{dx}$ (recall previously, we expressed in terms of $\phi(t)$)
  3. Substitute $\bar{u}(t)$ into Hamiltonian to obtain $\min_{u(t)} \mathbf{H}(x, u, \frac{\partial J(x,t)}{dx}, t)$
  4. Write corresponding HJB equation and solve analytically or numerically for $J(x, t)$; if analytic solution exists, obtain feedback control law (vector field) $\bar{u}(x, t)$ from $\bar{u}(x, t) = \frac{\partial J(x,t)}{dx}$

Comparing the Hamilton-Jacobi-Bellman equation with the PMP

- HJB replaces $\phi^T(t)$ with $\frac{\partial J(x,t)}{\partial x}$
- HJB provides $\bar{u}$ in state feedback form directly
- Solve a scalar pde with $N+1$ independent variables x, t rather than $2N$-dim vector ode (PMP-Hamiltonian system) with 1 independent variable t (latter is two-point boundary value problem)
- Depending on solution method, PMP may not provide control in state feedback form; e.g., with $L(u(t)) = \frac{1}{2}u^T(t)Ru(t)$, $\bar{u}(t) = BR^{-1}B^T\phi(t)$, not a function of x since $\phi(t)$ not a fn of x
- For certain integrable problems, e.g., LQR, PMP provides identical results to HJB since it can provide optimal controls analytically in feedback form
- HJB essential for optimal control of stochastic processes (which we study later) since control must always be formulated in terms of state feedback

HJB applied to linear quadratic regulator

- Derived LQR feedback control law above using state, costate equations and PMP
- With minor variations can show HJB gives same result; start with $J(x,t) = \frac{1}{2}x^T S(t)x(t)$ instead of $\phi(t) = S(t)x(t)$; then $\frac{\partial J(x,t)}{\partial x(t)} = S(t)x(t)$; $\bar{u}(x,t) = -BR^{-1}B^T S(t)x(t)$ as before
- Now use HJB equation $\frac{\partial J(x,t)}{\partial t} = -\min_{u(t)} \mathbf{H}(x(t), \frac{\partial J(x,t)}{\partial x(t)}, u(t), t)$:

$$
\begin{aligned}
\frac{\partial}{\partial t}[\frac{1}{2}x^T(t)S(t)x(t)] = & -\frac{1}{2}x^T(t)Qx(t)- \\
& -\frac{1}{2}(BR^{-1}B^T S(t)x(t))^T RBR^{-1}B^T S(t)x(t) \\
& -[S(t)x(t)]^T(A - BR^{-1}B^T S(t))x(t)
\end{aligned}
$$

  subject to $J(x(T),T) = \frac{1}{2}x^T(T)S(T)x(T)$
- Simplify and eliminate $x(t)$ to obtain Riccati equation as above (note: without using adjoint dynamical equation), with terminal boundary condition $S(T)$
- Can solve steady-state case analytically as above

Numerical methods for dynamic programming (discrete time)

- For nonlinear problems, typically no analytic solutions to HJB pdes. Can discretize control, state, and time and apply the following backwards-time algorithm to find the optimal feedback controls:

$$J^*(x, t_k) = \min_{u(x,t_k)} [L(x, u(x, t_k), t_k)\Delta t + J^*(x + \Delta x, t_{k+1})]$$

  with $\Delta x \equiv f(x, u(x, t_k), t_k)\Delta t$ and $J(x(T), T) = F(x(T), T)$

-
  1. For each t_k, find $J(x, t_k)$ for all x, by computing $J(u, x, t_k)$ for all x
  2. Choose $J^*(x, t_k)$ by choosing the u that gives the lowest cost for each x
  3. For each (x, t_k) pair you will then have associated optimal cost $J^*(x, t_k)$ to be used in subsequent steps
  4. Step backwards in time to t_{k-1} and repeat

Next time

- Next time: how to optimally update a state estimate $\hat{x}(t)$ for a noisy (stochastic) system based on observations made according to law $y(t) = Cx(t)$; will find "filtering" equations (Kalman-Bucy equations) are dual to those for feedback control
- Ultimately, will combine optimal state estimation and control for stochastic feedback control

Methods for state estimation

CHE 597

Purdue University

April 26, 2010

Outline

SET DEFAULT SPACING BET BULLETS

Outline

Outline

1. Review of concepts from classical probability
2. From probability to statistical inference: Properties of estimators
   - Least squares estimation of parameter vectors
3. The Kalman filter
   - Adaptive Kalman filter
4. Maximum likelihood estimation
   - MLE examples
   - Algorithms for MLE estimation

Consistency, invariance and asymptotic normality

- *Consistency:* An estimator $\hat{\theta}^m$ is *consistent* for the parameter θ (written as plim $\hat{\theta}^m = \theta_0$) if for every $\epsilon > 0$,

$$\lim_{m \to \infty} P_\theta \left\{ |\hat{\theta}^m - \theta_0| \geq \epsilon \right\} = 0.$$

- *Invariance:* For an invariant estimator, $c(\theta)$ is $c(\hat{\theta}^m)$, for a continuous and continuously differentiable function $c(\cdot)$.

- *Asymptotic Normality.* For a sequence of estimators $\hat{\theta}^m$, if $k_m \left(\hat{\theta}^m - \theta_0 \right) \xrightarrow{d} N(0, \Sigma)$ as $m \to \infty$, where $\xrightarrow{d}$ denotes convergence in distribution and k_m is any function of m, $\hat{\theta}^m$ is said to be $\sqrt{k_m}$-consistent for θ and has an asymptotic normal distribution with asymptotic covariance matrix Σ.

Least squares parameter (state) estimation

- Extending our discussion of observability; goal is to: estimate (parameters of) state x in presence of noise/random measurement outcomes, based on m measurements
- Again use deterministic observation law

$$y = Cx$$

(mean observation law), but now assume $m \times N$ matrix C has $m \geq N$ (enables estimation of all parameters) and add noise such that

$$z = y + w = Cx + w,$$

with w a m-dimensional Gaussian noise vector; $z = Cx + w$ is now the stochastic observation law

- Note C is in general not a change of basis even if $N \times N$ since not necc orthogonal (i.e. $CC^T \neq I$)

Least squares parameter estimation (cont)

- If m measurements are iid, matrix C has identical rows and pdfs of w_i's are identical, and no covariance/correlation between measurement outcomes
- Let $\hat{x}$ denote the estimated state; minimize least squares objective function of error residuals (sum of squared measurement errors over all state parameters/components)

$$J = \frac{1}{2}(z - C\hat{x})^T(z - C\hat{x})$$

- Note this only incorporates information about means y_i of observations through C, no other information about probability distributions (pdfs) of w components; thus we can only obtain parameter estimates $\hat{x}$ (means of estimate distributions if estimator unbiased), but not their uncertainties
- Set $\frac{dJ}{d\hat{x}} = 0$ for minim; solve for $\hat{x}$

$$\begin{aligned}
\frac{dJ}{d\hat{x}} &= \frac{d}{d\hat{x}}\frac{1}{2}(z^Tz - z^TC\hat{x} - \hat{x}^TC^Tz + \hat{x}^TC^TC\hat{x}) \\
&= -\frac{1}{2}(C^Tz + C^Tz) + C^TC\hat{x} = 0 \\
\hat{x} &= (C^TC)^{-1}C^Tz
\end{aligned}$$

- Thus state estimate is

$$\hat{x} = (C^T C)^{-1} C^T z$$

  $(C^T C)^{-1} C^T$ is called left pseudoinverse of C: compare $(A^T A)^{-1} A^T = A^{-1}$ for square A; result would be same if we had deterministic measurements (no noise) and we solved for x from $Cx = z$; estimator minimizes mean square error between estimates $\hat{x}_i$ and corresponding measurement outcomes $(C^T C)^{-1} C^T z + w_i$ across all i
- Note $C^T C$ must be full rank (rank N);
- For nonlinear observer (nonlinear least squares) must generally solve for minimum of J numerically; application of optimization to estimation

Weighted least squares estimation

- Incorporates information about variances of observations (e.g., if pdfs of w_i known) in order to provide (estimate of) (co)variance of parameter estimates $\hat{x}_i$: $\Sigma = \mathrm{E}[(x - \hat{x})(x - \hat{x})^T]$
- Measurement residual covariance matrix $(m \times m)$: $R = \mathrm{E}[ww^T]$, assuming zero mean; note if this matrix has nonzero off-diagonal elements, the measurements are correlated and hence not independent (and not iid)
- Each measurement z_i / w_i has different pdf (assumed to be Gaussian) with variance R_{ii}, and covariance between w_i, w_j is $R_{ij} = R_{ji}$
- For weighted least squares objective fn, let $J = \frac{1}{2}(z - C\hat{x})^T R^{-1}(z - C\hat{x})$; provides greater weights to measurements with lower variances in providing parameter estimates and estimator minimizes weighted mean square error between estimates $C\hat{x}_i$ and corresponding measurement outcomes z_i where weights are proportional to (co)variances

Weighted least squares estimation

- Setting $\frac{dJ}{d\hat{x}} = 0$ and solving

$$\begin{aligned}\frac{dJ}{d\hat{x}} &= \frac{d}{d\hat{x}}\frac{1}{2}(z^TR^{-1}z - z^TR^{-1}C\hat{x} - \hat{x}^TR^{-1}C^Tz + \hat{x}^TC^TR^{-1}C\hat{x}) \\ &= -\frac{1}{2}(C^TR^{-1}z^T + C^TR^{-1}z^T) + C^TR^{-1}C\hat{x} = 0 \\ \hat{x} &= (C^TR^{-1}C)^{-1}C^TR^{-1}z\end{aligned}$$

  Thus state estimate is $\hat{x} = (C^TR^{-1}C)^{-1}C^TR^{-1}z$; the matrix left multiplying z is called the weighted left pseudoinverse of C

- R is consistently estimated by sample covariance of measurements (residuals); in simplest case is diagonal matrix of inverse weights when measurements uncorrelated; but note this requires

Weighted least squares estimation (cont)

- We obtain an *estimate* of the covariance matrix of the parameter estimates as well:
$$\hat{\Sigma} = (C^T R^{-1} C)^{-1},$$
since $(\hat{\Sigma})^{-1} = C^T R^{-1} C$; note if C is $N \times N$ identity matrix (each measurement provides information on exactly one parameter), $\hat{\Sigma} = R$
- If the pdfs of w_i are Gaussian, and R is the true covariance matrix, then we obtain the true covariance matrix of parameter estimates from $\Sigma = (C^T R^{-1} C)^{-1}$
- Note that if the pdfs of w_i are Gaussian, all information about them is included within the means y_i and the (co)variances R_{ij}; but if not, information about the stochastic observation law is lost

Dynamic (recursive) linear least squares estimation

- Now consider successive measurement "sets" z_1, z_2 (indexed by time); and, where the measurement errors and observation law is changing between sets; for our purposes will assume m measurements (e.g. z_1) all made at time t_1, though need not be iid
- As before $\hat{x}_1 = (C^T R_1^{-1} C)^{-1} C^T R_1^{-1} z_1$; update to $\hat{x}_2$ with measurement set z_2; $\hat{x}_2$ estimate obtained using all info, but weighting t_1 and t_2 measurements appropriately
- Again formulate least squares objective

$$J = \frac{1}{2}[z_1 - C_1\hat{x}_2, z_2 - C_2\hat{x}_2] \begin{bmatrix} R_1^{-1} & 0 \\ 0 & R_2^{-1} \end{bmatrix} \begin{bmatrix} z_1 - C_1\hat{x}_2 \\ z_2 - C_2\hat{x}_2 \end{bmatrix}$$

- Write $\frac{dJ}{d\hat{x}} = 0$: by direct extension of above, obtain

$$\hat{x}_2 = (C_1^T R_1^{-1} C_1 + C_2^T R_2^{-1} C_2)^{-1} (C_1^T R_1^{-1} z_1 + C_2^T R_2^{-1} z_2);$$

- We are interested in how to update state estimate given new info; hence want $\hat{x}_2$ in terms of $\hat{x}_1$

Dynamic (recursive) linear least squares estimation

$$(C_1^T R_1^{-1} C_1 + C_2^T R_2^{-1} C_2)^{-1} = (\Sigma_1^{-1} + C_2^T R_2^{-1} C_2)^{-1}$$

- Apply variant of Sherman-Morrison matrix inversion lemma; recall $(A + u \otimes v)^{-1} = A^{-1} - \frac{(A^{-1}u)\otimes(A^{-1}v)}{1 - v^T A^{-1} u}$
- Here,

$$(A + B^T C^{-1} B)^{-1} = A^{-1} - A^{-1} B^T (B A^{-1} B^T + C)^{-1} B A^{-1}$$

- Thus

$$(\Sigma_1^{-1} + C_2^T R_2^{-1} C_2)^{-1} = \Sigma_1 - \Sigma_1 C_2^T (C_2 \Sigma_1 C_2^T + R_2)^{-1} C_2 \Sigma_1$$

so

$$x_2 = [\Sigma_1 - \Sigma_1 C_2^T (C_2 \Sigma_1 C_2^T + R_2)^{-1} C_2 \Sigma_1](C_1^T R_1^{-1} z_1 + C_2^T R_2^{-1} z_2)$$

Dynamic (recursive) linear least squares estimation

- Multiplying the terms in the left bracket with the first term on the right, obtain $\hat{x}_1 - \Sigma_1 C_2^T (C_2 \Sigma_1 C_2^T + R_2)^{-1} C_2 \hat{x}_1$
- Let $\Sigma_1 C_2^T (C_2 \Sigma_1 C_2^T + R_2)^{-1} \equiv K_2$; thus have $\hat{x}_1 - K_2 C_2 \hat{x}_1$
- Doing the same with the second term in the right bracket gives $K_2 z_2$
- Thus the recursive least squares state estimate update is $\hat{x}_2 = \hat{x}_1 + K_2(z_2 - C_2 \hat{x}_1)$; K_2 is called the Kalman gain for the estimator; updates estimate based on new observations z_2
- In continuous time obtain $\frac{d\hat{x}(t)}{dt} = K(t)(z(t) - C(t)\hat{x}(t))$ for a constant state vector that is measured with time-varying error and observation law

Propagation of the state and covariance estimates (without new observations)

- Assume we have state estimate $\hat{x}(0)$ and associated covariance matrix of state estimates $\Sigma(0)$
- Now turn on noisy linear dynamics governed by

$$\frac{dx}{dt} = Ax(t) + Bu(t) + Dn(t)$$

  where n is a N-dimensional white noise vector with covariance matrix $\mathrm{E}[nn^T] = Q$ (note this Q differs from that used in observability analysis)
- How are the state estimates and covariance matrix propagated through time given these dynamics? Want $\hat{x}(t)$ and $\Sigma(t)$

Propagation of the state and covariance estimates (without new observations)

- $\hat{x}(t)$ follows directly from our formal solution to linear vector ode:

$$\hat{x}(t) = \exp(At)\hat{x}(0) + \int_0^T \exp[A(t-t')]Bu(t')dt' +$$

$$\mathrm{E}\left[\int_0^T \exp[A(t-t')]Dn(t')dt'\right]$$

$$= \exp(At)\hat{x}(0) + \int_0^T \exp[A(t-t')]Bu(t')dt'$$

- For covariance update, omit control for now for simplicity

$$\Sigma(t) = \mathrm{E}[(x(t)-\hat{x}(t))(x(t)-\hat{x}(t))^T] =$$

$$= \mathrm{E}\left\{ \left[\int_0^T \exp[A(t-t')]Dn(t')dt' + \exp(At)(x(0)-\hat{x}(0))\right]\right.$$

$$\left. * \left[\int_0^T \exp[A(t-t')]Dn(t')dt' + \exp(At)(x(0)-\hat{x}(0))\right]^T \right\}$$

Propagation of the state and covariance estimates (without new observations)

$$\Sigma(t) = \exp(At)\Sigma(0)\exp(A^T t) + \mathrm{E}\left\{\left[\int_0^t \exp[A(t-t')]Dn(t')\;dt'\right]\left[\int_0^t \exp[A(t-t')]Dn(t')\;dt'\right]^T\right\}$$

$$\mathrm{E}\left\{\left[\int_0^t \exp[A(t-t')]Dn\;dt'\right]\left[\int_0^t \exp[A(t-t')]Dn\;dt'\right]^T\right\} =$$

$$= \int_0^t \exp[A(t-t')](Dn)(Dn)^T \exp[A^T(t-t')]\;dt'$$

$$= \int_0^t \exp[A(t-t')]\mathrm{E}[Dnn^T D^T]\exp[A^T(t-t')]\;dt'$$

$$= \int_0^t \exp[A(t-t')]DQD^T \exp[A^T(t-t')]\;dt'.$$

Propagation of the state estimate (with new observations)

- Next time: will look at evolution of state estimate with new observations $z(t)$:

$$\begin{aligned}\frac{d\hat{x}}{dt} &= A\hat{x} + K(z - C\hat{x}) \\ &= (A - KC)\hat{x} + Kz\end{aligned}$$

- Note similarity to state feedback form of control law; now using measurements to update state estimate rather than control the state
- Recall: observations are dual to controls

Outline

1. Review of concepts from classical probability
2. From probability to statistical inference: Properties of estimators
   - Least squares estimation of parameter vectors
3. The Kalman filter
   - Adaptive Kalman filter
4. Maximum likelihood estimation
   - MLE examples
   - Algorithms for MLE estimation

Filtering: optimal state estimation of dynamical systems

- Since the state covariance of a stochastic dynamical system increases with time of evolution, "optimal" feedback control based on state estimate $\hat{x}(t)$ is prone to error
- *Filtering* methods update the state estimate and its covariance matrix optimally based on additional measurements made during evolution; based on combination of i) state estimate / covariance matrix updates in presence of measurements, but absence of evolution; ii) state estimate / covariance matrix updates in presence of evolution, but absence of measurements
- Filters can be based on different estimators for the state and its covariance; we are studying the simplest, the least squares filter
- Kalman developed optimal least squares filter for linear dynamical systems (previously we studied Kalman controllability and observability rank conditions for linear systems)
- Applications

Recursive least squares estimators: from discrete to continuous time

- Recall

$$\hat{x}_2 - \hat{x}_1 = K_2(z_2 - C_2 x_1);$$
$$\Sigma_2 - \Sigma_1 = -K_2 C_2 \Sigma_1$$

- R_2 in K_2 represents effect of instantaneous measurement noise; now let us assume that noise enters measurement process continuously, building over time
- This is 1st step toward formulating continuous observations/continuous state update; even though we are still measuring at discrete times we need a continuous time representation of our noise

- Let $R(t_2)$ denote the total measurement error (covariance) that has built up over the interval Δt due to noise error rate R_2; so

$$R_2 \to \frac{1}{\Delta t} R(t_2)$$
$$R_2^{-1} \to R^{-1}(t_2)\Delta t$$

- How to incorporate into expression for Kalman gain:

$$K_2 = \Sigma_1 C_2^T [C_2 \Sigma_1 C_2^T + R_2]^{-1}$$

Would be useful to have an expression “proportional” to R_2^{-1}

Recursive least squares estimators: from discrete to continuous time

- Rewrite K_2 in a form “proportional” to the measurement error matrix R_2

$$
\begin{aligned}
K_2 &= \Sigma_1 C_2^T [C_2 \Sigma_1 C_2^T + R_2]^{-1} \\
&= \Sigma_1 C_2^T [(C_2 \Sigma_1 C_2^T R_2^{-1} + I) R_2]^{-1} \\
&= \Sigma_1 C_2^T R_2^{-1} [I + C_2 \Sigma_1 C_2^T R_2^{-1}]^{-1}
\end{aligned}
$$

$$K_2[I + C_2 \Sigma_1 C_2^T R_2^{-1}] = \Sigma_1 C_2^T R_2^{-1}$$

$$
\begin{aligned}
K_2 &= \Sigma_1 C_2^T R_2^{-1} - K_2 C_2 \Sigma_1 C_2^T R_2^{-1} \\
K_2 &= (I - K_2 C_2) \Sigma_1 C_2^T R_2^{-1}
\end{aligned}
$$

- By substituting $\Sigma_2 = \Sigma_1 - K_2 C_2 \Sigma_1 = (I - K_2 C_2)\Sigma_1$: we can eliminate K_2 on the rhs and get the form of K_2 that we want:

$$K_2 = \Sigma_2 C_2^T R_2^{-1}$$

- Making the substitution $R_2^{-1} \to R^{-1}(t_2)\Delta t$, we obtain the form of the gain we want:

$$K(t_2) = \Sigma(t_2) C^T(t_2) R(t_2)^{-1} \Delta t$$

Recursive least squares estimators: from discrete to continuous time

- Now move to continuous updating of the state estimate by taking $\lim_{\Delta t \to \infty}$ in $K(t_2) = \Sigma(t_2) C^T(t_2) R(t_2)^{-1} \Delta t$

$$\lim_{\Delta t \to \infty} \frac{\hat{x}(t_2) - \hat{x}(t_1)}{\Delta t} = \lim_{\Delta t \to \infty} K(t_2)[z(t_2) - C(t_2)\hat{x}(t_1)]$$
$$\frac{d\hat{x}(t)}{dt} = K(t)[z(t) - C(t)\hat{x}(t)]$$

- Similarly, get $\frac{d\Sigma(t)}{dt} = -K(t)C(t)\Sigma(t)$

Differential equation for state covariance matrix with propagation but no measurements

- Recall without measurements,

$$\Sigma(t) = \exp(At)\Sigma(0)\exp(A^T t) + \\ + \exp(At)[\int_0^t \exp(-At')DQD^T \exp(-A^T t')\ dt']\exp(A^T t)$$

- Let $\exp(At)[\int_0^t \exp(-At')DQD^T \exp(-A^T t')\ dt']\exp(A^T t) \equiv H(t)$. So

$$\frac{d\Sigma(t)}{dt} = A\exp(At)\Sigma(0)\exp(A^T t) + \\ \exp(At)\Sigma(0)\exp(A^T t)A^T + AH(t) + H(t)A^T + DQD^T \\ = A\Sigma(t) + \Sigma(t)A^T + DQD^T$$

- This is for time-invariant A, D, Q; for time-variant $A(t), D(t), Q(t)$, replace $\exp(At)$ with formal propagator $U(t)$; same form obtained for $\frac{d\Sigma(t)}{dt}$ but with time-varying matrices

Differential equation for state covariance matrix with propagation and measurements

- Denote the covariance matrix with measurements but without propagation (dynamics) $\Sigma_1(t)$ and that without measurements but with propagation $\Sigma_2(t)$; putting them together and replacing Σ_1, Σ_2 on the rhs w $\Sigma(t)$

$$
\begin{aligned}
\frac{d\Sigma^{1+2}(t)}{dt} &= \frac{d\Sigma^1}{dt} + \frac{d\Sigma^2(t)}{dt} \\
&= -K(t)C(t)\Sigma(t) + A\Sigma(t) + \Sigma(t)A^T + DQD^T \\
&= A\Sigma(t) + \Sigma(t)A^T + DQD^T - \Sigma(t)C^T(t)R^{-1}(t)C(t)\Sigma(t)
\end{aligned}
$$

  with $\Sigma(0) = \Sigma_0$; where we have used $K(t) = \Sigma(t)C^T(t)R^{-1}(t)$
- Hence with continuous least squares state estimation, obtain a Riccati equation rather than a Lyapunov equation
- Again, for time-varying linear systems, replace A, D, Q with $A(t), D(t), Q(t)$

Differential equation for state estimate with propagation and measurements

- Similarly, we get

$$\begin{aligned}\frac{d\hat{x}^{1+2}(t)}{dt} &= A\hat{x}(t) + K(t)[z(t) - C(t)\hat{x}(t)] \\ &= A\hat{x}(t) + \Sigma(t)C^T(t)R^{-1}(t)[z(t) - C(t)\hat{x}(t)] \\ &= A\hat{x}(t) + \Sigma(t)C^T(t)R^{-1}(t)z(t) - \Sigma(t)C^T(t)R^{-1}(t)C(t)\hat{x}(t)\end{aligned}$$

Kalman filter equations

- So the *Kalman filter equations* for optimal updating of the state estimate and its error during dynamical evolution of a linear system are

$$\frac{d\hat{x}(t)}{dt} = A\hat{x}(t) + \Sigma(t)C^T(t)R^{-1}(t)z(t) - \Sigma(t)C^T(t)R^{-1}(t)C(t)\hat{x}(t);\ \hat{x}(0) = \hat{x}_0$$

$$\frac{d\Sigma(t)}{dt} = A\Sigma(t) + \Sigma(t)A^T + DQD^T - \Sigma(t)C^T(t)R^{-1}(t)C(t)\Sigma(t);\ \Sigma(0) = \Sigma_0$$

- Kalman filter minimizes state estimate covariance (mean square error) by optimally mixing old and new measurements

Duality: Kalman filter equations vis-a-vis linear quadratic regulator

- Compare the Kalman filter equations to those for optimal feedback control of linear systems to obtain a duality:

$$\frac{d\hat{x}(t)}{dt} = A(t)\hat{x}(t) + \Sigma(t)C^T(t)R^{-1}(t)z(t) - \Sigma(t)C^T(t)R^{-1}(t)C(t)\hat{x}(t);$$
$$\hat{x}(0) = \hat{x}_0$$
$$\frac{d\Sigma(t)}{dt} = A(t)\Sigma(t) + \Sigma(t)A^T(t) + D(t)Q(t)D^T(t) -$$
$$\Sigma(t)C^T(t)R^{-1}(t)C(t)\Sigma(t);\ \Sigma(0) = \Sigma_0$$

vs

$$\frac{dx(t)}{dt} = A(t)x(t) - B(t)K(t)x(t);\ x(0) = x_0$$
$$= A(t)x(t) - B(t)R^{-1}(t)B^T(t)S(t)x(t)$$
$$\frac{dS(t)}{dt} = S(t)A(t) + A^T(t)S(t) + Q(t) - S(t)B(t)R^{-1}(t)B^T(t)S(t);$$
$$S(T) = S_T$$

Duality: Kalman filter equations vis-a-vis linear quadratic regulator

- Ignoring the $z(t)$ term, they are dual with the mappings

$$C^T(t) \to B(t)$$
$$\Sigma(t) \to S(t)$$
$$\hat{x}(t) \to x(t)$$

and time reversed for the Riccati equation (in Riccati equation, duality is more precise with $A(t) \to A^T(t)$).

Adaptive Kalman filter equations

- *Adaptive Kalman filter* enables estimation of dynamical parameters as well as states; updates parameter estimates and their covariances in real time
- Let p denote a l-dimensional vector of parameters
- Dynamical parameters could be elements of the matrices A or D; i.e., we have $A(p)$ or $D(p)$; we consider the case $D(p)$ because of its interpretation in terms of the intensity of noise entering the system: now $\frac{dx}{dt} = Ax(t) + D(p(t))n(t)$
- Accomplishes this by "augmenting" the state vector $x(t)$ with the parameters $p(t)$; obtain a $N + l$-dimensional augmented vector $x_a(t) = (x(t), p(t))$; assume true $p(t) = p(0)$ although estimate $\hat{p}$ will evolve with time
- The corresponding covariance matrix is then $(N + l) \times (N + l)$: denote Σ_a
- Want differential equations for $\frac{\hat{x}_a(t)}{dt}$ and $\frac{\Sigma_a(t)}{dt}$
- Adaptive filtering introduces nonlinearities in the filtering equations
- Applications

Adaptive Kalman filter equations

- The *adaptive Kalman filter equations* for optimal updating of the state estimate and its error during dynamical evolution of a linear system are

$$\frac{d\hat{x}_a(t)}{dt} = \frac{d(\hat{x}(t), \hat{p}(t))}{dt}$$

$$= A_a\hat{x}_a(t) + \Sigma_a(t)C_a^T(t)R_a^{-1}(t)(z_a(t) - \hat{x}_a(t)); \;\; \hat{x}_a(0) = \hat{x}_{a,0}$$

$$\frac{d\Sigma_a(t)}{dt} = A_a\Sigma_a(t) + \Sigma_a(t)A_a^T + D_a(\hat{p}(t))Q_aD_a^T(\hat{p}(t)) -$$

$$\Sigma_a(t)C_a^T(t)R_a^{-1}(t)C_a(t)\Sigma_a(t); \;\; \Sigma_a(0) = \Sigma_{a,0}$$

Augmented matrices in the adaptive Kalman filter

- The $(N+l) \times (N+l)$ matrix Q_a is block diagonal with an $N \times N$ block equal to Q and the other $l \times l$ block zero
- The $(N+l) \times (N+l)$ A_a is block diagonal with an $N \times N$ block equal to A and the other $l \times l$ block zero
- The $(m+l) \times (m+l)$ matrix R_a is block diagonal with an $m \times m$ block equal to R and the other $l \times l$ block zero
- The $(N+l) \times m$ matrix C_a has the upper $N+l$ rows equal to C and the other l rows zero
- $\hat{x}_a(0) = (\hat{x}(0), \hat{p}(0))$
- Note that the matrix $D_a(p)$ in the Riccati equation for $\Sigma_a(t)$ must be expressed in terms of the parameter estimates $\hat{p}(t)$; this increases the nonlinearity of the differential equation

Example of adaptive Kalman filter

- Consider the linear stochastic system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix} n(t)$$

- The parameter values d_1, d_2 are unknown, though we have initial estimates and a covariance matrix for them
- The dynamical equation for x_a is

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{d}_1 \\ \dot{d}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ d_1 \\ d_2 \end{bmatrix} + \begin{bmatrix} d_1 & 0 & 0 & 0 \\ 0 & d_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} n_a$$

- In the Ricatti equation for the covariance update, we use

$$D(\hat{p}(t)) = \begin{bmatrix} \hat{d}_1(t) & 0 & 0 & 0 \\ 0 & \hat{d}_2(t) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Outline

1. Review of concepts from classical probability
2. From probability to statistical inference: Properties of estimators
   - Least squares estimation of parameter vectors
3. The Kalman filter
   - Adaptive Kalman filter
4. Maximum likelihood estimation
   - MLE examples
   - Algorithms for MLE estimation

Maximum likelihood

- MLE vs least squares: in least squares, can only estimate parameters that are linear functions of the means of the pdfs of the observations; in MLE can estimate any parameters that specify the pdfs for the observations
- y_i's are means of z_i pdfs, σ_i's are variances; can estimate $y_i's$ and x_i's (latter are linear fns of the $y_i's$, but not σ_i's, by least squares theory); MLE provides a theory for estimation of σ_i's as well
- Achieves this by maximizing a function of all the parameters (here y_i or x_i's, σ_i's)
- By maximizing the log likelihood, the ML estimator minimizes the Kullback-Leibler distance between the estimated and true probability distributions.
- Will show how this allows estimation of the variances σ_i^2 in the expressions $z_i = y_i + w_i$, where $w_i \sim \mathcal{N}(0, \sigma_i^2)$ in addition to the means y_i
- Consider example of iid samples from a univariate Gaussian distribution: have just one σ, one y

Likelihood function; necessary conds for lhood

- The *likelihood function* $L(\theta|z)$ is the joint density of the sample defined as a function of the unknown parameter vector θ

- Let $z = (z_1, \cdots, z_m)$ be an *i.i.d.* sample of size m from a population with probability density function $p(z|\theta)$ which depends on the unknown parameter vector θ whose true value is θ_0. Typically, the logarithm of the likelihood function, $\ln L(\theta|z)$, is easier to maximize numerically because of its separability.
- The value of the parameter vector that maximizes the (log) likelihood function is called the ML estimator of θ:

$$\widehat{\theta}^m_{ML} = \arg\max_{\theta\in\Theta} L(\theta|z) = \arg\max_{\theta\in\Theta} \left(\prod_{i=1}^{m} p(z_1|\theta)\cdots p(z_m|\theta) \right),$$

where Θ denotes the admissible parameter space.

Asymptotic efficiency of an estimator

Asymptotically efficient. A sequence of consistent estimators $\hat{\theta}^m$ is asymptotically efficient if $\hat{\theta}^m - \theta_0 \xrightarrow{d} \mathcal{N}[0, I^{-1}(\theta_0)]$ where $I(\theta) = -\mathrm{E}\left[\frac{\partial^2 \ln L(\theta|z)}{\partial\theta\partial\theta'}\right]$; $[I(\theta_0)]^{-1}$ is called the *Cramer-Rao lower bound (CRB)* for consistent estimators.
In practice, can usually use

$$\widehat{I}_1(\hat{\theta}^m) = -\left[\frac{\partial^2 \ln L(\hat{\theta}^m|z)}{\partial\theta\partial\theta'}\right]$$

Example: MLE of Gaussian distribution parameters

- We now show how to apply MLE to the case discussed above for a single component state vector, observations of which are distributed normally; the goal is to estimate the mean μ (as done above by least squares, called x_1 or y_1 above) and also the variance σ_1^2 of the distribution
- Parameter estimates: mean μ of Gaussian distribution; first assume σ^2 is known

$$p(z|\mu) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(z-\mu)^2}{2\sigma^2}\right]; \; L(\mu|z) = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(z_i-\mu)^2}{2\sigma^2}\right]$$

$$\ln L = \sum_{i=1}^{m} \ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{(z_i-\mu)^2}{2\sigma^2}$$

$$\frac{d \ln L(\mu|z)}{d\mu} = \sum_{i=1}^{m} \frac{z_i-\mu}{\sigma^2} = 0$$

$$\hat{\mu} = \frac{1}{m} \sum_{i=1}^{m} z_i$$

Example: MLE of Gaussian distribution parameters (cont)

- Parameter estimates: variance σ^2; now assume mean μ is known

$$\frac{d \ln L(\sigma|z)}{d\sigma} = \frac{d}{d\sigma}\left[\sum_{i=1}^{m} - \ln\sqrt{2\pi\sigma^2} - \frac{(z_i-\mu)^2}{2\sigma^2}\right] = 0$$

$$\frac{m}{\sigma} = \sum_{i=1}^{m} \frac{(z_i-\mu)^2}{\sigma^3}$$

$$\hat{\sigma}^2 = \sum_{i=1}^{m} \frac{(z_i-\mu)^2}{m}$$

- This is just the variance of the observations; note this could not be obtained directly from least squares theory
- If both μ, σ^2 were simultaneously estimated, would substitute their estimated rather than true values in the expressions above

Example: MLE of Gaussian distribution parameters (cont)

- We compute (asymptotic) $\hat{\mu}$ estimator uncertainty based on Fisher information:

$$\left[-\frac{d^2 \ln L(\mu|z)}{d\mu^2}\right]^{-1} = \left[-\sum_{i=1}^{m} -\frac{d}{d\mu}\frac{\mu}{\sigma^2}\right]^{-1} = \frac{\sigma^2}{m}$$

- Note this is the same result as that used (though not derived) above in least squares and also coincides with the variance σ^2 of the Gaussian distribution itself
- Also, can show this is equivalent to result obtained from $\mathrm{E}\left[\left(\frac{d \ln L(\mu|z)}{d\mu}\right)^2\right]^{-1}$
- Could even compute uncertainty in the estimate of σ^2
- Use MLE for constant state estimation, but we will use LS for dynamic state estimation because like prev OCT theory minimizes quadratic objective function and will exploit duality between control and estimation

Properties of maximum likelihood estimators

1. The ML estimator is consistent: plim $\widehat{\theta}^m_{ML} = \theta_0$.

2. The ML estimator is asymptotically normally distributed (and asymptotically efficient):

$$\begin{aligned} \sqrt{m}\;[\hat{\theta}^m_{ML} - \theta_0] \quad &\rightarrow \quad \mathcal{N}[0, I^{-1}(\theta_0)], \\ \text{where } I(\theta_0) \quad &= \quad -\mathrm{E}\left[\frac{\partial^2 \ln L(\theta_0|x)}{\partial\theta\partial\theta'}\right]. \end{aligned}$$

3. The ML estimator of θ is invariant; e.g., as in least squares if I estimate x_i's, obtain y_i estimates via $\hat{y} = C\hat{x}$

The need for numerical algorithms

- In the above examples, we were able to solve the score function equations for the parameter estimates in closed form.
- Typically, this is not possible, and the zeroes must be found using numerical methods.

Constrained optimization: Lagrange multipliers

- Many MLE problems require imposition of constraints on parameters.

Constrained optimization: Lagrange multipliers

- Many MLE problems require imposition of constraints on parameters.

- Requires constrained optimization, using a Lagrangian function

Constrained optimization: Lagrange multipliers

- Many MLE problems require imposition of constraints on parameters.
- Requires constrained optimization, using a Lagrangian function
- Denote the vector of parameters $(\theta, \lambda, \gamma) \equiv \mathbf{t}$. Finding the constrained optimum corresponding to this Lagrangian entails searching for parameters $\mathbf{t}$ θ_i and slack variables γ_j that render the gradient vectors $\nabla L(\theta)$ and a linear combination of $\nabla(a_j(\theta) - \gamma_j)$, $j = 1, ..., N$ parallel.

Algorithms for MLE estimation

There are two common approaches to solving this problem:

1. Minimization of the "sum of squares" (of the first-order conditions) function $\sum_i \left(\frac{\partial \mathcal{L}}{\partial t_i}\right)^2$;
2. Finding the roots of the system of nonlinear equations $\frac{\partial \mathcal{L}}{\partial \mathbf{t}} = 0$ using the Newton-Raphson (NR) method.

In fact, methods 1) and 2) may be combined to produce a globally convergent NR algorithm.

Newton-Raphson method

The *Newton-Raphson* method is ideal... Writing $\frac{\partial \mathcal{L}}{\partial \mathbf{t}} = \mathbf{H}(\mathbf{t})$, the Newton step for

$$\mathbf{H}(\mathbf{t}) = 0$$

is

$$\mathbf{t}_{\text{new}} = \mathbf{t}_{\text{old}} + \delta \mathbf{t},$$

with $\delta \mathbf{t} = -\mathbf{J}^{-1}\mathbf{H}$, where $J_{ij} = \frac{\partial H_i}{\partial t_j}$ is the Jacobian matrix.

- Denoting the rows of **H** by H_i, we have: REPLACE W GENERAL FORM

$$\begin{aligned} H_i(\theta) &= \frac{\partial \mathcal{L}(\theta,\lambda,\gamma|x)}{\partial \theta_i} = \frac{\partial \ln L(\theta|x)}{\partial \theta_i} = 0,\ 1 \le i \le N^2-1, \\ H_{N^2+j-1}(\theta) &= \frac{\partial \mathcal{L}(\theta,\lambda,\gamma|x)}{\partial \lambda_j} = a_j(\theta) = 0,\ 1 < j \le N-1, \\ H_{N^2+N+j-2}(\lambda,\gamma) &= \frac{\partial \mathcal{L}(\theta,\lambda,\gamma|x)}{\partial \gamma_j} = 2\lambda_j\gamma_j = 0\ 1 < j \le N-1. \end{aligned}$$

- Denoting the rows of $\mathbf{H}$ by H_i, we have: REPLACE W GENERAL FORM

$$
\begin{aligned}
H_i(\theta) &= \frac{\partial \mathcal{L}(\theta, \lambda, \gamma | x)}{\partial \theta_i} = \frac{\partial \ln L(\theta | x)}{\partial \theta_i} = 0, \; 1 \leq i \leq N^2 - 1, \\
H_{N^2+j-1}(\theta) &= \frac{\partial \mathcal{L}(\theta, \lambda, \gamma | x)}{\partial \lambda_j} = a_j(\theta) = 0, \; 1 < j \leq N - 1, \\
H_{N^2+N+j-2}(\lambda, \gamma) &= \frac{\partial \mathcal{L}(\theta, \lambda, \gamma | x)}{\partial \gamma_j} = 2\lambda_j \gamma_j = 0 \; 1 < j \leq N - 1.
\end{aligned}
$$

- In order to faciliate global convergence of the Newton-Raphson algorithm, the "sum-of-squares" function $h = \mathbf{H} \cdot \mathbf{H}$ is evaluated after each iteration, and the step length progressively shortened until the value of this function is found to decrease (the existence of such a step length is guaranteed)

(Provide some further details on NR from Press)

Stochastic processes and algorithms

CHE 597 - Quantum Control Engineering - Spring 2010

Purdue University

March 5-10, 2010

Introduction to stochastic search algorithms

- Deterministic search algorithms based on the gradient or Hessian may be trapped in local optima of the objective function

Introduction to stochastic search algorithms

- Deterministic search algorithms based on the gradient or Hessian may be trapped in local optima of the objective function
- The number of local optima typically increases with the dimensionality of the parameter space, the nonlinearity of the objective function and constraints on the objective function

Introduction to stochastic search algorithms

- Deterministic search algorithms based on the gradient or Hessian may be trapped in local optima of the objective function
- The number of local optima typically increases with the dimensionality of the parameter space, the nonlinearity of the objective function and constraints on the objective function
- Unlike deterministic algorithms, *stochastic search algorithms* (SSA's) do not rely on derivatives of the objective function, but rather only on the function itself

Introduction to stochastic search algorithms

- Deterministic search algorithms based on the gradient or Hessian may be trapped in local optima of the objective function
- The number of local optima typically increases with the dimensionality of the parameter space, the nonlinearity of the objective function and constraints on the objective function
- Unlike deterministic algorithms, *stochastic search algorithms* (SSA's) do not rely on derivatives of the objective function, but rather only on the function itself
- SSA's are thus less prone to being trapped in local optima, though they may converge slowly

Introduction to stochastic search algorithms

- Deterministic search algorithms based on the gradient or Hessian may be trapped in local optima of the objective function
- The number of local optima typically increases with the dimensionality of the parameter space, the nonlinearity of the objective function and constraints on the objective function
- Unlike deterministic algorithms, *stochastic search algorithms* (SSA's) do not rely on derivatives of the objective function, but rather only on the function itself
- SSA's are thus less prone to being trapped in local optima, though they may converge slowly
- SSA's use a family of "walkers" that randomly traverse the parameter space, accepting or rejecting moves based on comparison of objective function values at different points

Introduction to stochastic search algorithms

- Deterministic search algorithms based on the gradient or Hessian may be trapped in local optima of the objective function
- The number of local optima typically increases with the dimensionality of the parameter space, the nonlinearity of the objective function and constraints on the objective function
- Unlike deterministic algorithms, *stochastic search algorithms* (SSA's) do not rely on derivatives of the objective function, but rather only on the function itself
- SSA's are thus less prone to being trapped in local optima, though they may converge slowly
- SSA's use a family of "walkers" that randomly traverse the parameter space, accepting or rejecting moves based on comparison of objective function values at different points
- Depending on the ruggedness of the objective function, either deterministic, hybrid deterministic/stochastic, or stochastic algorithms may be used

Expectation, (co)variance, correlation

- A *random variable* is a map $M : X \to \mathbb{R}$, where X is called the sample or state space
- *Expectation* of a random variable: $\mathrm{E}[a] = \langle a \rangle = \int_A \; ap(a) \; da$
  Sample mean: $\sum_{i=1}^{N} \frac{a_i}{N}$
- *Covariance* of random variables a and b:
  $\int_A \int_B (a - \langle a \rangle)(b - \langle b \rangle) p(a, b) \; da \; db$
  The covariance matrix of a multivariate random vector x (sample space is vector space) is

$$\mathrm{E}[(x - \langle x \rangle)(x - \langle x \rangle)^T] = \int_X (x - \langle x \rangle)(x - \langle x \rangle)^T p(x) dx$$

- *Correlation* of random variables a and b:
  $\mathrm{Cor(a, b)} = \frac{\mathrm{Cov(a,b)}}{\sqrt{\mathrm{Var(a)}}\sqrt{\mathrm{Var(b)}}}$. I.e., a "normalized" covariance.
  Sample correlation: $\sum_i \frac{(a_i - \bar{a})(b_i - \bar{b})}{N\sqrt{\sum_i (a_i - \bar{a})^2/N}\sqrt{\sum_i (b_i - \bar{b})^2/N}}$.

Statistical (in)dependence and conditional distributions

- *Joint distribution* of random variables a and b: $p(a, b)$
- Independently distributed: $p(a, b) = p(a)p(b)$
- Independently and identically distributed: $p(a, b) = p(a)p(b) = p(a)p(a)$
- *Conditional distribution* of random variable a given b: $p(a|b) = \frac{p(a,b)}{p(b)}$
- *Marginal (unconditional) distribution* of random variable a (in a multivariate distribution): $\int_B p(a, b)\ db$
- Bayes' rule: $p(a|b) = \frac{p(b|a)p(a)}{p(b)}$

Stochastic processes: definitions

- A stochastic sequence (discrete time *stochastic process*) is a sequence (indexed set) of random variables, i.e. $x(t_i), \ i = 1, 2, 3, \cdots$, where each $x(t_i)$ is a random variable and where the index set is countable.
- A *continuous time stochastic process* is one where the index set is uncountable (e.g., $t \in [0, T]$).

Stochastic processes (cont)

- *Stationary (ergodic) stochastic process*: $p(x(t)) = p(x(t')) \equiv \pi(x), \forall t \geq t'$; there is a unique unconditional distribution, which is called the stationary distribution, to which the unconditional density converges over time.
- *Nonstationary stochastic process*: may be different distribution functions $p(x(t))$ at different times t: no unique unconditional distribution.
- *Autocovariance*:

$$\mathrm{E}_x[(x(t) - \mathrm{E}_x[x(t)])(x(t') - \mathrm{E}_x[x(t')]] = \int_X \int_X (x(t) - \langle x(t) \rangle)(x(t') - \langle x(t') \rangle) p(x(t), x(t'))\ dx(t)\ dx(t'),\ \ t \geq t'$$

- *Autocorrelation*:

$$\mathrm{E}_x[(x(t) - \langle x(t) \rangle)(x(t') - \langle x(t') \rangle)]/\sigma(x(t))\sigma(x(t')) = \frac{\int_X \int_X (x(t) - \bar{x}(t))(x(t') - \bar{x}(t') p(x(t), x(t'))\ dx(t)\ dx(t')}{\sigma(x(t))\sigma(x(t'))}, \quad t \geq t'$$

Stochastic processes (stationary)

- For a sp that has converged to stationarity, joint distribution $p(x(t), x(t'))$ only depends on $t - t'$
- An ergodic sp can also be represented by an ensemble of chains; at any given time this ensemble is characterized by an unconditional distribution function $p(x(t))$ (frequency of walkers in state x at time t), which may not be the stationary distribution, depending on each chain's initial state $x(t_0)$
- White noise stationary sp: autocorrelation 0 for all $t' \neq t$; for Gaussian white noise, conditional and unconditional probabilities equal, i.e., $p(x(t)|x(t')) = p(x(t)), \ t \geq t'$

Markov chains

- For algorithms, we will be most interested in discrete time sp's
- A general (discrete time) *vector autoregressive process* of order n can be written:
  $x(t_{i+1}) = A_1 x(t_i) + A_2 x(t_{i-1}) + \cdots + A_n x(t_{i-n+1}) + Qu(t_{i+1})$
  u_{i+1} denotes a zero-mean n-variate white noise, and $QQ^T = Q^T Q$ denotes the covariance matrix if $u(t_i)$ each have unit standard deviation
- A *Markov process* is a discrete time autoregressive process of order 1 (compare first-order deterministic differential equation), i.e., $x(t_{i+1}) = Ax(t_i) + Qu(t_{i+1})$(this equation is called a *stochastic difference equation*).
- A *Markov chain path* is a sequence of points $(x(t_1), ..., x(t_m))$ (draws) corresponding to a Markov process.
- In general a Markov process is not stationary
- From here on, we will use the notation $x(t_i) \equiv x_i$ (note we are not referring to vector component indices with this subscript)

Markov chain transition rules, matrices

- For Markov chains on discrete state spaces, a *transition matrix* defines the conditional probability of the various possible states at time $t = i + 1$ depending on the state at time $t = i$.
- An example of a transition probability matrix (also called a *stochastic matrix*) for a 3-d state space is:

$$P = \begin{bmatrix} 0.5 & 0.5 & 0.25 \\ 0.25 & 0 & 0.25 \\ 0.25 & 0.5 & 0.5 \end{bmatrix}$$

- The transition matrix must have columns summing to one, and operates on either state vectors or probability vectors (those with elements summing to one).
- For a discrete state space, a state vector is of the form $x = (0, \cdots, 1, \cdots, 0)^T$
- Eigenvectors and eigenvalues of P are important for characterizing dynamics: these need not be probability vectors.

Chapman-Kolmogorov equation

- When P operates on a state vector, the result is a conditional probability :$Px_i = P(x_{i+1}|x_i)$; when P operates on an (unconditional) probability vector, the result is an unconditional probability: $Pp(x_i) = p(x_{i+1})$
- This is called the *Chapman-Kolmogorov equation* for evolution of the marginal distribution associated with a stochastic process
- For Markov chains on continuous state spaces (supports), we have a *transition rule* which takes the form of a function $p(x_1|x_2)$; the requirement of columns summing to one is equivalent to $\int_X p(x_1|x_2)\ dx_1 = 1$
- For continuous state space:

$$p(x_n) = \int_X p(x_n|x_{n-1}) \int_X p(x_{n-1}|x_{n-2}) \cdots \\ \cdots \int_X p(x_1|x_0)\ p(x_0) dx_0 \cdots dx_{n-2}\ dx_{n-1}$$

- For discrete state space: $p(x_n) = P^n p(x_0)$, with P^n a stochastic matrix

Sufficient conditions for ergodicity

- Compare the Kolmogorov equation to the action of the discrete time dynamical propagator for deterministic dynamical systems (e.g., quantum systems): $U(t_n) = V(t_{n-1})V(t_{n-1})\cdots V(t_0)$. Recall each $V(t_i)$ is identical for a time-independent Hamiltonian; compare P^n.

- Conditions (on P) for ergodicity:
  1. *Irreducible*: P has one unit eigenvalue $\lambda_1 = 1$ (unique stationary distribution)
  2. *Aperiodic*: P does not have any eigenvalues $\lambda = -1$ (equilibria are stable, so system does not oscillate between states in infinite time limit).

Sufficient conditions for ergodicity: continuous distributions

- It can be shown that any scalar Markov process whose stochastic difference equation is $x_{i+1} = ax_i + qu_{i+1}$ with $|a| = 1$ is nonstationary (random walk), with $a = 1$ violating irreducibility and $a = -1$ violating aperiodicity; moreover, a sufficient condition for ergodicity is that $|a| < 1$.
- For continuous state spaces (still discrete time), transition operator is an integral operator and its eigenvalue spectrum (hence convergence rate) is more difficult to determine analytically.
- a is generally not directly known
- However, there is a convenient condition for stationarity expressed in terms of the transition probabilities $p(x_j|x_i)$ and the unconditional distribution $\pi(x)$.
- Any Markov chain that satisfies the *detailed balance* condition $p(x_2|x_1)\pi(x_1) = p(x_1|x_2)\pi(x_2)$, where $\pi(x_i)$ denotes the stationary distribution and $p(x_2|x_1)$ denotes an element of the transition matrix (transition probability for continuous state spaces), is ergodic.

Sufficient conditions for ergodicity (cont)

- The detailed balance condition implies

$$p(x_2) = \int_X p(x_2|x_1)\pi(x_1)\ dx_1 = \pi(x_2)\int_X p(x_1|x_2)\ dx_1 = \pi(x_2)$$

  for continuous state space, i.e., if the unconditional distribution for x at time $t = 1$ was $\pi(x)$, then the unconditional distribution at time $t = 2$ is also $\pi(x)$
- This shows $\pi(x)$ is an eigenvector of the transition probability operator with eigenvalue 1; we will not prove convergence to this distribution
- Convergence to stationarity occurs in the infinite time limit for continuous state spaces; for discrete state spaces, convergence can occur in finite time: need $P^n p(x_0)$; then columns of P^n are nothing but the stationary distribution $\pi(x)$ and $P^{n+1} = PP^n = P^n$.

Autocorrelation of stationary Markov processes

- Since $\mathrm{E}[x_{i-1}u_i] = \mathrm{E}[x_{i-1}]\mathrm{E}[u_i] = 0$, for a Markov process that has converged to stationarity the 1st-order autocorrelation function (omitting the means and scale factors)

$$\begin{aligned}\mathrm{E}[x_i^T x_{i-1}] &= \mathrm{E}[(Ax_{i-1} + Qu_i)^T x_{i-1}] \\ &= \mathrm{E}[(Ax_{i-1})^T x_{i-1}] + \mathrm{E}[(Qu_i)^T x_{i-1}] \\ &= \mathrm{E}[(Ax_{i-1})^T x_{i-1}]\end{aligned}$$

- In the scalar case, with $|a| < 1$, for the $k - th$ order autocorrelation function (lag k), we have

$$\mathrm{E}[x_i x_{i-k}] = a^{k-1}\mathrm{E}[x_{i-k+1}x_{i-k}] = a^k \mathrm{E}[x_{i-k}^2]$$

; i.e. the acf decays geometrically with k (time)

- Note that for a stationary Markov process, $\lim_{k\to\infty} \mathrm{acf}\,(t_i - t_{i-k}) = 0$.

Stationary Markov process: mixing rate

- For a stationary Markov process, we have $\lim_{i\to\infty} p(x_i) = \pi(x)$.
- The *mixing rate* of a Markov process is the rate at which the limit is approached
- The critical value of k at which the acf decays to approximately 0 is related to the mixing time for the Markov process
- Note that the decay of the autocorrelation function with lag depends only on a, but the mixing time also depends on the noise/error term u_i since that also contributes to the eigenvalue spectrum of the transition operator P
- But, by estimating the autocorrelation function numerically, one can obtain insight into the mixing rate/time

Monte Carlo sampling: introduction

- The simplest stochastic search algorithm is *Monte Carlo (MC) sampling*

Monte Carlo sampling: introduction

- The simplest stochastic search algorithm is *Monte Carlo (MC) sampling*
- MC methods integrate or optimize functions based on a) generation of random points in the parameter space b) incorporation of these points into the integral or search trajectory based on the function values at those points

Monte Carlo sampling: introduction

- The simplest stochastic search algorithm is *Monte Carlo (MC) sampling*
- MC methods integrate or optimize functions based on a) generation of random points in the parameter space b) incorporation of these points into the integral or search trajectory based on the function values at those points
- a) is achieved by sampling from a *proposal distribution*

Monte Carlo sampling: introduction

- The simplest stochastic search algorithm is *Monte Carlo (MC) sampling*
- MC methods integrate or optimize functions based on a) generation of random points in the parameter space b) incorporation of these points into the integral or search trajectory based on the function values at those points
- a) is achieved by sampling from a *proposal distribution*
- In the Metropolis algorithm (discussed further in 2nd half of term), the proposal distribution is (typically) a function of the difference between the current point and the previously sampled point, i.e., $q(x_{i+1} - x_i)$; a typical form is $q(x_{i+1} - x_i) = \frac{1}{\sqrt{2\pi}} \exp[-\frac{1}{2}(x_{i+1} - x_i)^2]$.

Monte Carlo sampling: introduction

- The simplest stochastic search algorithm is *Monte Carlo (MC) sampling*
- MC methods integrate or optimize functions based on a) generation of random points in the parameter space b) incorporation of these points into the integral or search trajectory based on the function values at those points
- a) is achieved by sampling from a *proposal distribution*
- In the Metropolis algorithm (discussed further in 2nd half of term), the proposal distribution is (typically) a function of the difference between the current point and the previously sampled point, i.e., $q(x_{i+1} - x_i)$; a typical form is $q(x_{i+1} - x_i) = \frac{1}{\sqrt{2\pi}} \exp[-\frac{1}{2}(x_{i+1} - x_i)^2]$.
- The aim of MC is to sample, through correlated sequential draws, from a stationary unconditional distribution $\pi(x)$ that is otherwise difficult for impossible to sample

Metropolis-Hastings MC sampling

- In Metropolis-Hastings sampling, the acceptance probability of a move $x_1 \to x_2$ is

$$\alpha(x_1, x_2) = \min\left[1, \frac{\pi(x_2)q(x_1|x_2)}{\pi(x_1)q(x_2|x_1)}\right]$$

  (Metropolis sampling omits the factor $\frac{q(x_1|x_2)}{q(x_2|x_1)}$, which $= 1$ for a symmetric proposal distribution)
- A common choice for $\pi(x)$ is $\exp\{-\frac{1}{kT}f(x)\}$ where $\beta = \frac{1}{kT}$ is an adjustable parameter called the inverse temperature in analogy with thermodynamics; thus obtain

$$\min\left[1, \exp\{-\frac{1}{kT}[f(x_{i+1}) - f(x_i)]\}\right]$$

Metropolis-Hastings MC sampling

- In Metropolis-Hastings sampling, the acceptance probability of a move $x_1 \to x_2$ is

$$\alpha(x_1, x_2) = \min\left[1, \frac{\pi(x_2)q(x_1|x_2)}{\pi(x_1)q(x_2|x_1)}\right]$$

(Metropolis sampling omits the factor $\frac{q(x_1|x_2)}{q(x_2|x_1)}$, which $= 1$ for a symmetric proposal distribution)

- A common choice for $\pi(x)$ is $\exp\{-\frac{1}{kT}f(x)\}$ where $\beta = \frac{1}{kT}$ is an adjustable parameter called the inverse temperature in analogy with thermodynamics; thus obtain

$$\min\left[1, \exp\{-\frac{1}{kT}[f(x_{i+1}) - f(x_i)]\}\right]$$

Metropolis-Hastings MC sampling

- In Metropolis-Hastings sampling, the acceptance probability of a move $x_1 \to x_2$ is

$$\alpha(x_1, x_2) = \min\left[1, \frac{\pi(x_2)q(x_1|x_2)}{\pi(x_1)q(x_2|x_1)}\right]$$

(Metropolis sampling omits the factor $\frac{q(x_1|x_2)}{q(x_2|x_1)}$, which $= 1$ for a symmetric proposal distribution)

- A common choice for $\pi(x)$ is $\exp\{-\frac{1}{kT}f(x)\}$ where $\beta = \frac{1}{kT}$ is an adjustable parameter called the inverse temperature in analogy with thermodynamics; thus obtain

$$\min\left[1, \exp\{-\frac{1}{kT}[f(x_{i+1}) - f(x_i)]\}\right]$$

- The M-H transition probability $p(x_2|x_1) = q(x_2|x_1)\alpha(x_1, x_2)$ satisfies the detailed balance principle and hence the chain converges to stationarity

Metropolis-Hastings MC sampling satisfies detailed balance

- Need to show $\pi(x_1)p(x_2|x_1) = \pi(x_2)p(x_1|x_2)$
- We have $p(x_2|x_1) = q(x_2|x_1)\alpha(x_1, x_2)$

$$
\begin{aligned}
\pi(x_1)q(x_2|x_1)\alpha(x_1, x_2) &= \min\left[\pi(x_1)q(x_2|x_1), \pi(x_2)q(x_1|x_2)\right] \\
&= \min\left[\pi(x_2)q(x_1|x_2), \pi(x_1)q(x_2|x_1)\right] \\
&= \pi(x_2)q(x_1|x_2)\alpha(x_2, x_1)
\end{aligned}
$$

Monte Carlo sampling as an optimization algorithm

- With fixed β, Metropolis MC chains converge to draws from the stationary distribution $\pi(x) = \exp\{-\beta f(x)\}$

Monte Carlo sampling as an optimization algorithm

- With fixed β, Metropolis MC chains converge to draws from the stationary distribution $\pi(x) = \exp\{-\beta f(x)\}$
- Lower β drive the search towards lower values of $f(x)$; higher β increase the probability of transitions to higher values of $f(x)$

Monte Carlo sampling as an optimization algorithm

- With fixed β, Metropolis MC chains converge to draws from the stationary distribution $\pi(x) = \exp\{-\beta f(x)\}$
- Lower β drive the search towards lower values of $f(x)$; higher β increase the probability of transitions to higher values of $f(x)$
- There is thus a finite probability of escaping local optima while minimizing $f(x)$ for MC search, unlike gradient-based optimization

Monte Carlo sampling as an optimization algorithm

- With fixed β, Metropolis MC chains converge to draws from the stationary distribution $\pi(x) = \exp\{-\beta f(x)\}$
- Lower β drive the search towards lower values of $f(x)$; higher β increase the probability of transitions to higher values of $f(x)$
- There is thus a finite probability of escaping local optima while minimizing $f(x)$ for MC search, unlike gradient-based optimization
- Can be used on either discrete or continuous parameter spaces, unlike gradient optimization

Monte Carlo sampling as an optimization algorithm

- With fixed β, Metropolis MC chains converge to draws from the stationary distribution $\pi(x) = \exp\{-\beta f(x)\}$
- Lower β drive the search towards lower values of $f(x)$; higher β increase the probability of transitions to higher values of $f(x)$
- There is thus a finite probability of escaping local optima while minimizing $f(x)$ for MC search, unlike gradient-based optimization
- Can be used on either discrete or continuous parameter spaces, unlike gradient optimization
- Test whether the system has reached equilibrium/stationarity at given β by (i) (if running one chain) checking geometric decay of autocorrelation function; (ii) (if running m multiple chains in parallel) comparing unconditional variance σ^2 of each parameter x_i: $\sigma_{x_i}^2 = \frac{1}{n}\sum_i (x_i - \bar{x}_i)^2$ *within a chain* to that *between chains* indexed by j: $\frac{n}{m}\left(\bar{x}_i^j - \frac{1}{m}\sum_j \bar{x}_i^j\right)^2$. At stationarity, they should be approximately the same for runs with large number of iterations n.

Monte Carlo sampling as an optimization algorithm

- With fixed β, Metropolis MC chains converge to draws from the stationary distribution $\pi(x) = \exp\{-\beta f(x)\}$
- Lower β drive the search towards lower values of $f(x)$; higher β increase the probability of transitions to higher values of $f(x)$
- There is thus a finite probability of escaping local optima while minimizing $f(x)$ for MC search, unlike gradient-based optimization
- Can be used on either discrete or continuous parameter spaces, unlike gradient optimization
- Test whether the system has reached equilibrium/stationarity at given β by (i) (if running one chain) checking geometric decay of autocorrelation function; (ii) (if running m multiple chains in parallel) comparing unconditional variance σ^2 of each parameter x_i: $\sigma^2_{x_i} = \frac{1}{n}\sum_i (x_i - \bar{x}_i)^2$ *within a chain* to that *between chains* indexed by j: $\frac{n}{m}\left(\bar{x}_i^j - \frac{1}{m}\sum_j \bar{x}_i^j\right)^2$. At stationarity, they should be approximately the same for runs with large number of iterations n.
- Latter method helps assess convergence to stationary distributions with multiple peaks. Early on, within-chain variance will be smaller than (scaled) between-chain variance because of high correlation between successive steps

Simulated annealing

- Convergence to the stationary distribution can be extremely slow for multimodal stationary distributions $\pi(x)$ (equivalently, functions $f(x)$ with multiple local optima when using $\pi(x) = \exp\{-\beta f(x)\}$)

Simulated annealing

- Convergence to the stationary distribution can be extremely slow for multimodal stationary distributions $\pi(x)$ (equivalently, functions $f(x)$ with multiple local optima when using $\pi(x) = \exp\{-\beta f(x)\}$)
- MC *simulated annealing* mimics the process of "slow cooling" that nature uses to escape local optima in molecular energy functions

Simulated annealing

- Convergence to the stationary distribution can be extremely slow for multimodal stationary distributions $\pi(x)$ (equivalently, functions $f(x)$ with multiple local optima when using $\pi(x) = \exp\{-\beta f(x)\}$)
- MC *simulated annealing* mimics the process of "slow cooling" that nature uses to escape local optima in molecular energy functions
- Simulated annealing either a) gradually increases β (lowers T) over a range $[\beta_2, \beta_1]$; or b) repeatedly cycles β between $[\beta_2, \beta_1]$

Simulated annealing

- Convergence to the stationary distribution can be extremely slow for multimodal stationary distributions $\pi(x)$ (equivalently, functions $f(x)$ with multiple local optima when using $\pi(x) = \exp\{-\beta f(x)\}$)
- MC *simulated annealing* mimics the process of "slow cooling" that nature uses to escape local optima in molecular energy functions
- Simulated annealing either a) gradually increases β (lowers T) over a range $[\beta_2, \beta_1]$; or b) repeatedly cycles β between $[\beta_2, \beta_1]$
- Equilibrate at each β

Simulated annealing

- Convergence to the stationary distribution can be extremely slow for multimodal stationary distributions $\pi(x)$ (equivalently, functions $f(x)$ with multiple local optima when using $\pi(x) = \exp\{-\beta f(x)\}$)
- MC *simulated annealing* mimics the process of "slow cooling" that nature uses to escape local optima in molecular energy functions
- Simulated annealing either a) gradually increases β (lowers T) over a range $[\beta_2, \beta_1]$; or b) repeatedly cycles β between $[\beta_2, \beta_1]$
- Equilibrate at each β
- Can run multiple walkers in parallel from random initial guesses, discarding those at each temperature that display low variance (indicative of traps)

Simulated annealing

- Convergence to the stationary distribution can be extremely slow for multimodal stationary distributions $\pi(x)$ (equivalently, functions $f(x)$ with multiple local optima when using $\pi(x) = \exp\{-\beta f(x)\}$)
- MC *simulated annealing* mimics the process of "slow cooling" that nature uses to escape local optima in molecular energy functions
- Simulated annealing either a) gradually increases β (lowers T) over a range $[\beta_2, \beta_1]$; or b) repeatedly cycles β between $[\beta_2, \beta_1]$
- Equilibrate at each β
- Can run multiple walkers in parallel from random initial guesses, discarding those at each temperature that display low variance (indicative of traps)
- Do not artificially select walkers based on their values of the objective function $f(x)$, since the function may be rugged and we only want to encourage thorough sampling of the landscape; walks are already "biased" toward lower values of f

Simulated annealing

- Convergence to the stationary distribution can be extremely slow for multimodal stationary distributions $\pi(x)$ (equivalently, functions $f(x)$ with multiple local optima when using $\pi(x) = \exp\{-\beta f(x)\}$)
- MC *simulated annealing* mimics the process of "slow cooling" that nature uses to escape local optima in molecular energy functions
- Simulated annealing either a) gradually increases β (lowers T) over a range $[\beta_2, \beta_1]$; or b) repeatedly cycles β between $[\beta_2, \beta_1]$
- Equilibrate at each β
- Can run multiple walkers in parallel from random initial guesses, discarding those at each temperature that display low variance (indicative of traps)
- Do not artificially select walkers based on their values of the objective function $f(x)$, since the function may be rugged and we only want to encourage thorough sampling of the landscape; walks are already "biased" toward lower values of f
- Assess convergence to global maximum / stationary distribution by (i) number of times same local maxima are resampled, starting from different initial conditions (different chains); or (ii) comparing between- and within-chain variances (which should gradually align with cooling)

Setting the proposal covariance matrix

- The proposal distribution

$$p(x_{i+1} - x_i)$$

  is typically taken to be a multivariate normal distribution.
- A general multivariate normal distribution can be written

$$p(x) = C \exp\left[-\frac{1}{2}(x - \langle x \rangle)^T \Sigma^{-1} (x - \langle x \rangle)\right]$$

  where Σ denotes the $N \times N$ covariance matrix, $\langle x \rangle$ denotes the mean vector, x denotes the vector of random variables, and $C = \left(\frac{1}{2\pi}\right)^{n/2} |\Sigma|^{-\frac{1}{2}}$
- For the Gaussian proposal distribution, $\langle x \rangle$ is taken to be the current parameter vector

Numerical methods for sampling from probability distributions

- *Transformation methods* for drawing from nonstandard pdf $p(y)$ rely on choosing function $x = f(y)$ and drawing from $p(x)$
- Since infinitesimal area element under each pdf must be conserved, $p(y)dy = p(x)dx$ or $p(y) = p(x)\ \frac{dx}{dy}$; choice of $p(x)$ specifies $f(y)$
- Let $p(x) = U(0,1)$ (uniform distribution between 0 and 1); what is $f(y)$?
- Then $p(y) = \frac{dx}{dy}$; and $x = P(y)$, where $P(y)$ is indefinite integral of $p(y)$
- Then $y = P^{-1}(x)$; draws from $U(0,1)$ can be converted to draws from $p(y)$ if $P^{-1}(x)$ can be computed

Numerical methods for sampling from multivariate probability distributions (cont)

- For multivariate distributions $p(y_1, \cdots, y_n)$, let $x = f(y)$ denote a system of n nonlinear equations in the y_i
- Then $p(y_1, \cdots, y_n) = p(x_1, \cdots, x_n)\,|J(y)|$, where the determinant of the Jacobian of the transformation ($J_{ij} = \frac{\partial x_i}{\partial y_j}$) represents the scaling factor for transformation of volume elements $dx_1, \cdots, dx_n;\ dy_1, \cdots, dy_n$.
- Simplfies when f is linear transformation

Sampling from the multivariate Gaussian proposal distribution

- Two approaches can be used to draw from such a distribution (transformation methods): (i) Cholesky decomposition, $\Sigma = QQ^T$ (Q is lower triangular for any symmetric Σ; possibly on homework), with $X_{i+1} = X_i + Qu_i$, where u_i is a multivariate Gaussian "white noise" process with unit variance, or (ii) eigenvector decomposition $\Sigma = O\tilde{\Sigma}O^T$, with $\tilde{X}_{i+1} = \tilde{X}_i + v_i$, where v_i has variances equal to the diagonal elemetns of $\tilde{\Sigma}$, followed by rotation back to the original basis
- Note that here, only a linear transformation of x is necessary since it is possible to sample directly from univariate Gaussians

Rejection sampling

- Choose a “comparison function” $f(x) \geq p(x)$
- Use transformation method to sample x from $f(x)$ using uniform sampling of x
- Draw uniformly in interval $[0, f(x)]$ and accept if below $p(x)$, reject if above $p(x)$
- Above method is equivalent to sampling from $p(x)$, although may be computationally inefficient based on how close $f(x)$ and $p(x)$ are

Setting the proposal covariance matrix (cont)

- The covariance Σ, and hence Q, can be set "adaptively": let S_n denote the sample covariance matrix over the last n steps; then let $\Sigma_{new} = \alpha\Sigma_{old} + (1-\alpha)S_n$
- This allows the algorithm to "learn" the topography of the landscape by favoring trial moves that step in the directions that have been accepted previously
- This method can be used to facilitate convergence; if the autocorrelation function is decaying too slowly,
- Note that this adaptation constitutes control of the evolution of the stochastic difference equation (to accelerate the evolution of the Kolmogorov system to a fixed point) by modulation of the noise term

Setting the annealing schedule

- Compute the *heat capacity* to determine the ideal annealing (cooling) schedule, based only on statistics at current temperature T:

$$
\begin{aligned}
C(T) &= \frac{d}{dT}\langle E\rangle(T) \\
&= \frac{d}{dT}\left[\frac{\sum_i E_i \exp(-E_i/kT)}{\sum_i \exp(\frac{-E_i}{kT})}\right] \\
&= \frac{1}{Z^2}\Bigg[\frac{1}{T^2}\sum_i E_i \exp(-E_i/kT)(\sum_i \exp(-E_i/kT)) - \\
&\qquad - \sum_i E_i \exp(-E_i/kT)\frac{d}{dT}\sum_i \exp(-E_i/kT)\Bigg] \\
&= \frac{1}{T^2}[\langle E^2\rangle - \langle E\rangle^2]
\end{aligned}
$$

- If the heat capacity is sharply rising between successive temperatures, reduce the annealing rate to avoid becoming trapped in a local optimum.

Stochastic optimal control

CHE 597

Purdue University

April 27, 2010

Outline

Outline

1. Further details on stochastic processes
2. From deterministic to stochastic control
3. Stochastic optimal control without filtering
   - Linear stochastic optimal control (without filtering)
4. Stochastic optimal control with filtering
   - Linear stochastic optimal control with filtering

Stochastic processes: from discrete to continuous time

- In the stochastic process lectures we studied dynamics in discrete time:

$$x_{k+1} - x_k = Ax_k + Dn_{k+1}$$

where n_k was called a N-variate white noise vector

- Consider case with $A = 0$:

$$x_{k+1} - x_k = Dn_{k+1};$$

the corresponding stochastic process x is called a Wiener process, Brownian motion or a random walk

- However for filtering we subsequently worked in continuous time, in order to connect with our previous results on continuous time linear dynamical systems; we wrote $\frac{dx}{dt} = Ax(t) + Dn(t)$ For $A = 0$,

$$\frac{dx}{dt} = Dn(t)$$

Stochastic differential equations (sdes)

- The Brownian motion in continuous time is $x(t)$. Rigorously, though, the continuous time white noise $n(t)$ does not exist, since $x(t)$ can change position by a finite amount instantaneously and hence is not differentiable
- *Stochastic differential equations* are thus properly written $dx(t) = Dn(t)dt$ or more generally

$$dx(t) = Ax(t)\ dt + D\ n(t)dt = Ax(t)\ dt + D\ d\omega(t)$$

  where $d\omega(t) \propto \sqrt{dt}$ (the constant vector of proportionality is a standard deviation vector)
- The definition of $d\omega(t)$ in terms of $\sqrt{dt}$ rather than dt avoids the problem of singularity in the derivative and avoids continuous time white noise; since $\sqrt{dt}$ is larger than dt, it is not infinitesimally small

Stochastic differential equations (cont)

- Practically, the important point is that when one computes $\mathrm{E}[d\omega(t)d\omega^T(t)]$, one obtains

$$\mathrm{E}[d\omega(t)d\omega^T(t)] = N(t)\ dt$$

  where $N(t)$ is a covariance matrix (previously called Q but now because mixing estimation and control Q will be used in OCT cost functional); note the dt arises from two factors of $d\omega(t)$; thus the variance of the increment of Brownian motion is infinitesimally small, even though the increment itself may not be
- We can continue to use our old notation of continuous time white noise, as long as we recognize:

$$\mathrm{E}[n(t)dt\ n^T(t)dt] = N(t)\ dt;$$

  since we always integrate over time for our solutions, we will replace stochastic differential equations with ordinary differential equations bearing this rule of *stochastic calculus* in mind

Outline

1. Further details on stochastic processes
2. From deterministic to stochastic control
3. Stochastic optimal control without filtering
   - Linear stochastic optimal control (without filtering)
4. Stochastic optimal control with filtering
   - Linear stochastic optimal control with filtering

Stochastic optimal control objectives

- For stochastic dynamics, can no longer aim to drive the system to a precise final state
- Goal: to control moments of a deterministic cost functional (cost-to-go): e.g. its (unconditional) mean (expectation value) or its variance; we focus on mean:

$$\min_{u(t)} \mathrm{E}\left[F(x(T)) + \frac{1}{2}\int_0^T x^T(t)Q(t)x(t) + u^T(t)R_c(t)u(t)\ dt \right]$$

- The dynamical constraint for optimization is now a stochastic differential equation
- Optimal control must always be expressed in feedback form $\bar{u}(x(t))$ or $\bar{u}(\hat{x}(t))$

Stochastic optimal control: with and without filtering

- Two different frameworks:
  1. Direct observation of the state - e.g. $y = Cx$ observation law with $\mathrm{rank}\ C \geq N$ (here conditional covariance matrix Σ comes from dynamical noise alone)
  2. Noisy observation of the state - if linear observer, $z = Cx + w$ (example: quantum observations through $\langle \Theta \rangle = \mathrm{Tr}(\rho(t)\Theta)$, where $\rho(t)$ is state)
- **Case 1**: despite noisy dynamics, at any given time apply the optimal $\bar{u}(x(t))$ since we can observe the state directly
- **Case 2**: requires a method of filtering to obtain $\hat{x}(t)$ for the feedback law and then combine control with filtering for $\bar{u}(\hat{x}(t))$

Outline

1. Further details on stochastic processes
2. From deterministic to stochastic control
3. Stochastic optimal control without filtering
   - Linear stochastic optimal control (without filtering)
4. Stochastic optimal control with filtering
   - Linear stochastic optimal control with filtering

Stochastic HJB partial differential equation

- Recall deterministic HJB equation:

$$\frac{\partial J^*(x,t)}{\partial t} = -\min_{u(t)} \mathbf{H}(x(t), \frac{\partial J^*(x,t)}{\partial x(t)}, u(t), t)$$

- Consider case with direct measurement of state at each time $x(t)$
- Stochastic HJB equation has an additional term that is a function of the process noise covariance matrix DND^T
- For stochastic systems, need to do second order expansion: will find second order term will contribute to $\frac{\partial J^*(x,t)}{\partial t}$

Stochastic HJB pde (cont)

$$\frac{dJ(x,t)}{dt} = \frac{\partial J(x,t)}{\partial t} + L(x(t),u(t),t) + \frac{\partial J(x,t)}{\partial x}(f + Dn(t)) +$$

$$+ \frac{1}{2}(f + Dn(t))^T \frac{\partial^2 J(x,t)}{\partial x^2}(f + Dn(t))dt$$

$$\mathrm{E}\left[\frac{\partial J(x,t)}{\partial t}\right] = -\mathrm{E}\Bigg[L(x(t),u(t),t) + \frac{\partial J(x,t)}{\partial x}(f + Dn(t)) -$$

$$- \frac{1}{2}(f + Dn(t))^T \frac{\partial^2 J(x,t)}{\partial x^2}(f + Dn(t))\Bigg]$$

$$\frac{\partial J(x,t)}{\partial t} = -\left[L(x(t),u(t),t) + \frac{\partial J(x,t)}{\partial x} f(x,u)\right] -$$

$$- \frac{1}{2}\mathrm{E}[\mathrm{Tr}[\frac{\partial^2 J(x,t)}{\partial x^2} Dn(t)n^T(t)D^T]dt]$$

Stochastic HJB pde (cont)

$$\frac{\partial J^*(x,t)}{\partial t} = -\min_{u(t)} \mathbf{H}(x(t), u(t), \frac{\partial J^*(x,t)}{\partial x}, t) - $$
$$- \frac{1}{2}\mathrm{E}[\mathrm{Tr}[\frac{\partial^2 J^*(x,t)}{\partial x^2} Dn(t)n^T(t)D^T]]dt$$
$$= -\min_{u(t)} \mathbf{H}(x(t), u(t), \frac{\partial J^*(x,t)}{\partial x}, t) - $$
$$\frac{1}{2}\mathrm{Tr}[\frac{\partial^2 J^*(x,t)}{\partial x^2} D\mathrm{E}[n(t)n^T(t)]D^T]dt$$
$$\frac{\partial J^*(x,t)}{\partial t} = -\min_{u(t)} \mathbf{H}(x(t), u(t), \frac{\partial J^*(x,t)}{\partial x}, t)] - $$
$$\frac{1}{2}\mathrm{Tr}[\frac{\partial^2 J^*(x,t)}{\partial x^2} DN(t)D^T]$$

since $\mathrm{E}[n(t)n^T(t)]dt = N(t)$ for continuous-time white noise

Linear-quadratic stochastic optimal control problems: no filtering

- As in deterministic case start with ansatz $J(x,t) = \frac{1}{2}x^T(t)S(t)x(t)$ but add stochastic increment $\int_t^T \mathrm{Tr}[S(t')N(t')D^T]\, dt'$
- Substitute into HJB equation:

$$\frac{\partial J^*(x,t)}{\partial t} = -\min_{u(t)} \mathbf{H}(x(t), u(t), \frac{\partial J^*(x,t)}{\partial x}, t) + \mathrm{Tr}[S(t)DN(t)D^T]$$

$$\begin{aligned}\frac{\partial J^*(x,t)}{\partial t} = & -\frac{1}{2}x^T(t)Qx(t) - \\ & -\frac{1}{2}(R^{-1}B^TS(t)x(t))^T R(R^{-1}B^TS(t)x(t)) \\ & -[S(t)x(t)]^T(A - BR^{-1}B^TS(t))x(t) + \mathrm{Tr}[S(t)DN(t)D^T]\end{aligned}$$

subject to $J^*(x(T),T) = \frac{1}{2}x^T(T)S(T)x(T)$

- Now derive Riccati equation:

$$
\begin{aligned}
&\frac{1}{2}[x^T(t)\dot{S}(t)x(t)] + \mathrm{Tr}[S(t)DQ(t)D^T] = \\
&\qquad -\frac{1}{2}x^T(t)[A^TS(t) + S(t)A + Q - S(t)BR^{-1}B^TS(t)]x(t) - \\
&\qquad - \mathrm{Tr}[S(t)DQ(t)D^T] + \mathrm{Tr}[S(t)DN(t)D^T] \\
&\quad = -\frac{1}{2}x^T(t)[A^TS(t) + S(t)A + Q - S(t)BR^{-1}B^TS(t)]x(t) + \\
&\qquad\qquad + \mathrm{Tr}[S(t)DN(t)D^T] \\
&\dot{S}(t) = -A^TS(t) - S(t)A - Q + S(t)BR^{-1}B^TS(t)
\end{aligned}
$$

- Time-varying state-feedback control law:

$$\bar{u}(x(t)) = -R^{-1}B^TS(t)x(t)$$

Outline

1. Further details on stochastic processes
2. From deterministic to stochastic control
3. Stochastic optimal control without filtering
   - Linear stochastic optimal control (without filtering)
4. Stochastic optimal control with filtering
   - Linear stochastic optimal control with filtering

Filtrations

- When the state must be estimated based on noisy measurements, optimal decisions/controls must be based on the *information set* at any given time t:

$$\mathcal{I}(t_0, t) = (z(t_0, t), u(t_0, t))$$

  , i.e. conditional (conditioned) on all past observations; this is also referred to as a filtration
- Filters (e.g. Kalman filter) are used to translate the filtration $\mathcal{I}(t_0, t)$ into derived state and covariance estimate histories; these histories constitute the derived information set $\mathcal{I}_D(t_0, t) = (\hat{x}(t_0, t), \Sigma(t_0, t))$ which is used by the controller (note this is dependent on the type of estimator/filter used); we will use the notations $\mathcal{I}$ and $\mathcal{I}_D$ interchangeably
- For a Markovian stochastic process, $\mathcal{I}(t_0, t) = \mathcal{I}(t) = (\hat{x}(t), \Sigma(t))$ since the future evolution depends explicitly only on the the current state and covariance matrix

Linear stochastic optimal control problems with filtering

- Linear stochastic control problem analogous to LQR, with process as well as measurement uncertainty, is called linear quadratic gaussian regulator (LQG regulator)
- Write expected cost function given incomplete information set (filtration):

$$\mathrm{E}\left[x^T(T)S(T)x(T) + \int_0^T x^T(t)Q(t)x(t) + u^T(t)R(t)u(t)\ dt\right] =$$

$$= \mathrm{E}\Big\{\mathrm{E}\left[x^T(T)S(T)x(T)|\mathcal{I}(t)\right] +$$

$$+ \quad \int_0^T \mathrm{E}\left[x^T(t)Q(t)x(t) + u^T(t)R(t)u(t)|\mathcal{I}(t)\right]\ dt\Big\}$$

$$= \mathrm{E}\Big\{\mathrm{Tr}[S(T)x(T)x^T(T)|\mathcal{I}(T)] + \int_0^T \mathrm{Tr}[Q(t)x(t)x^T(t)|\mathcal{I}(t)] +$$

$$\mathrm{Tr}[R(t)u(t)u^T(t)]\ dt\Big\}$$

Linear stochastic optimal control problems with filtering (cont)

- Note that $\mathrm{E}[x(t)x^T(t)|\mathcal{I}(t)]$ appears in the cost functional; rewrite this in terms of

$$\begin{aligned}\Sigma &= \mathrm{E}[(x(t)-\hat{x}(t))(x(t)-\hat{x}(t))^T|\mathcal{I}(t)] \\ &= \mathrm{E}[x(t)x^T(t)|\mathcal{I}(t)] - 2\mathrm{E}[x(t)\hat{x}^T(t)|\mathcal{I}(t)] + \mathrm{E}[\hat{x}(t)\hat{x}(t)|\mathcal{I}(t)]) \\ &= \mathrm{E}[x(t)x^T(t)|\mathcal{I}(t)] - \mathrm{E}[\hat{x}(t)\hat{x}(t)|\mathcal{I}(t)]\end{aligned}$$

- So $\mathrm{E}[x(t)x^T(t)|\mathcal{I}(t)] = \Sigma(t) + \hat{x}(t)\hat{x}(t)$

- Thus

$$
\begin{aligned}
2\mathrm{E}[J] &= \mathrm{E}\Big\{\mathrm{Tr}[S(T)\hat{x}(T)\hat{x}^T(T)] + \mathrm{Tr}[S(T)\Sigma(T)]+ \\
&\quad + \int_0^T \mathrm{Tr}[Q(t)\hat{x}(t)\hat{x}^T(t)] + \mathrm{Tr}[R(t)u(t)u^T(t)] + \mathrm{Tr}[Q(t)\Sigma(t)]\; dt\Big\} \\
&= \mathrm{E}\Big\{\mathrm{Tr}[S(T)\hat{x}(T)\hat{x}^T(T)] + \int_0^T \mathrm{Tr}[Q(t)\hat{x}(t)\hat{x}^T(t)]+ \\
&\quad \mathrm{Tr}[R(t)u(t)u^T(t)]\; dt\Big\} + \mathrm{E}\Big\{\mathrm{Tr}[S(T)\Sigma(T)] + \int_0^T \mathrm{Tr}[Q(t)\Sigma(t)]\; dt\Big\} \\
&= J_{CE} + J_S
\end{aligned}
$$

- J_{CE} is called “certainty-equivalent” cost functional; **note it is same as stochastic cost functional with but with x replaced by $\hat{x}$**
- For the control systems we are studying, control does not affect J_S - can formulate optimization problem based only on minimization of J_{CE} (however, in certain applications $\Sigma(t)$ can be controlled)
- Covariance matrix Σ includes contributions from both estimation error and noisy dynamics

Certainty-equivalence principle for linear Gaussian systems

- Recall dynamical constraint for deterministic LQR controller was

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t),$$

whereas dynamical constraint for stochastic system with direct state observation was

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t) + Dn(t)$$

- Dynamical constraint for LQG controller is

$$\frac{d\hat{x}(t)}{dt} = A\hat{x}(t) + Bu(t) + K_e(t)(z(t) - C(t)\hat{x}(t));$$

control problem is $\min_{u(t)} J_{CE}$ subject to this constraint

- Note that in expectation, the term $K_e(t)(z(t) - C(t)\hat{x}(t))$ is distributed normally with mean zero, just like $Dn(t)$ in stochastic control with direct state observation; thus Riccati equation is identical and doesn't depend on $\hat{x}(t)$ or $\Sigma(t)$
- The feedback controller Riccati equation is (propagated backward in time from $S(T)$):

$$\frac{dS(t)}{dt} = -A^T S - SA - Q + S(t)BK_c(t)$$

Certainty-equivalence principle for linear Gaussian systems (cont)

- Certainty-equivalence means that the control problem can be solved as if the state $\hat{x}(t)$ were directly observed
- The optimal feedback control is $\bar{u}(\hat{x}(t)) = R^{-1}B^T S(t)\hat{x}(t)$
- The state estimate $\hat{x}(t)$ is continuously updated through the Kalman filter Riccati equation
- Implementation steps:
  1. Solve controller Riccati equation by backwards propagation from $S(T)$ (appears in cost functional)
  2. Propagate $\frac{d\hat{x}(t)}{dt} = A\hat{x}(t) + Bu(\hat{x}(t)) + K_e(t)(z(t) - C(t)\hat{x}(t))$ forward from $\hat{x}(0)$, simultaneously with propagation of filter Riccati equation forward from $\Sigma(0)$

Solving nonlinear stochastic optimal control problems with noisy measurements

- HJB solution method presented above assumes state can be observed directly without error; this allows us to replace $\mathrm{E}[J(x(t), t)]$ with $J(x(t), t)$ and similarly $\mathrm{E}[x(t)]$ with its known values at all times
- For nonlinear systems with measurement error, $\mathrm{E}[J(x(T), T)] = \mathrm{E}[F(x(T))]$ not known with certainty and depends on measurements made over all time $[0, T]$
- Thus filter, which is integrated forward in time, is coupled, and both cannot be solved simultaneously
- For linear systems, decoupling of estimator from controller occurs; for nonlinear systems, generally not possible and error is incurred by assuming information set up to time t is sufficiently similar to information set over all time
- Filter determines the filtration forward in time, while the controller (HJB solution) determines the optimal state-dependent feedback laws backward in time

Solving nonlinear stochastic optimal control problems with noisy measurements

- Decoupling for linear systems occurs because $\dot{\Sigma}(t)$ Riccati equation is unaffected by control $u(t)$; thus controller Riccati equation can be solved first, backwards from $S(T)$, while filter Ricatti equation (covariance update) can be solved separately; note $\Sigma(t)$ is required for integration of $\frac{d\hat{x}(t)}{dt}$ but not vice versa
- LQG derivation relies on equivalence between dynamical constraint including filter and linear Markovian sde (like that used in LQR derivation); in expectation the noise (innovation) term does not appear hence optimal $\bar{u}(\hat{x}(t))$ law is identical to that for LQR and controller gain can be computed offline
- For nonlinear systems $u(t)$ can affect $\Sigma(t)$
- Even if we ignore this we still need to solve HJB equation

Neighboring optimal (perturbative) feedback control

- We have seen that solving for optimal feedback controls for nonlinear (stochastic) systems is difficult; require HJB pde solution for field of extremals; but these are most common circumstance
- Neighboring optimal methods are based on linearization of system around deterministic trajectory - can apply linear estimation and control methodology locally
- Preliminary steps:
  1. Solve for optimal controls in absence of measurements or noise for nonlinear system (need not be expressed in feedback form; use PMP)
  2. Now linearize nonlinear system around the reference trajectory:
  $$A(t) = \frac{\partial f}{\partial x}[\hat{x}_r(t), u_r(t), t],$$
  $$B(t) = \frac{\partial f}{\partial u}[\hat{x}_r(t), u_r(t), t]$$
  Note this means to substitute the optimal state and control trajectories $\hat{x}_r(t), u_r(t)$ in after analytic differentiations of nonlinear vector functions f; although the resulting expressions $A(t), B(t)$ will not be analytic, they can be used in numerical integration of the corresponding Riccati equations
  3. Define deviation variables
  $\Delta\hat{x}(t) = \hat{x}_r(t) - \hat{x}(t),\ \ \Delta u(t) = u_r(t) - u(t)$

Neighboring optimal feedback control methods (cont)

- Filtering and control steps:
  1. Solve the corresponding *linear* feedback control problem by integrating Riccati equations for controller and filter and updating deviation $\Delta\hat{x}(t)$ based on observations. Cost functional:

$$F(\Delta\hat{x}(T)) + \frac{1}{2}\int_0^T \Delta\hat{x}^T(t)Q\Delta\hat{x}(t) + \Delta u^T(t)R\Delta u(t)\ dt$$

  2. Optimization of this cost functional subject to the linearized dynamical constraint

$$\Delta\dot{x}(t) = A(t)\Delta\hat{x}(t) + B(t)\Delta u(t) + K_e(t)[z(t) - C(t)\Delta\hat{x}(t)]$$

     provides the Riccati equation above for LQG
  3. Solve the corresponding *linear* filtering problem by integrating the filter Riccati equation:

$$\dot{\Sigma}(t) = A(t)\Sigma(t) + \Sigma(t)A^T(t) + DQD^T - \Sigma(t)C^T R^{-1}(t)C\Sigma(t)$$

     where the covariance matrix is now defined by $\Sigma(t) = \mathrm{E}[(\Delta x(t) - \Delta\hat{x}(t))(\Delta x(t) - \Delta\hat{x}(t))^T]$
  4. Update the state estimates in real-time in response to observations $z(t)$ according to above dynamical equation (here we have assumed a linear observation law); at each time apply the feedback control $\Delta u(\Delta\hat{x}(t)) = -R^{-1}B^T S(t)\Delta\hat{x}(t)$

Asymptotic stability of the Kalman filter

- Recall the quadratic cost used for derivation of the Kalman filter was $J = \frac{1}{2}(z(t) - C\hat{x}(t))^T R^{-1}(z(t) - C\hat{x}(t))$
- The appropriate Lyapunov function for assessment of stability of the Kalman filter is $J(t) = \frac{1}{2}(x(t) - \hat{x}(t))^T \Sigma^{-1}(t)(x(t) - \hat{x}(t))$
- The corresponding algebraic Riccati equation can be derived from the Riccati equation for Σ: (simply left/right multiply by Σ^{-1}):

$$\dot{\Sigma}^{-1}(t) = \Sigma^{-1}(t)A + A^T\Sigma^{-1}(t) + \Sigma^{-1}(t)DND^T\Sigma^{-1}(t) - C^T R^{-1} C$$

- $\Sigma^{-1}(t)$ plays the role of $S(t)$ in the feedback control Riccati equation
- Note use of Σ^{-1} in the Lyapunov function parallels use of R^{-1} in objective function

Asymptotic stability of the Kalman filter (cont)

- Note this Riccati equation is a function of A, C; the condition for stability is observability of the system
- Letting $\epsilon(t) \equiv x(t) - \hat{x}(t)$, the time-derivative of the Lyapunov function is

$$\dot{J}(\epsilon(t)) = -\epsilon(t)^T[\Sigma^{-1}(0)DND^T\Sigma^{-1}(0) + C^TR^{-1}C]\epsilon(t),$$

  which is negative definite
- Thus the estimation error decays to zero as the time over which the measurements are made approaches infinity.
- By extending our results on stability of controllable linear feedback controllers and observable linear filters, the deviation of $x(t)$ and $x(t) - \hat{x}(t)$ from zero decay asymptotically for the LQG
- Since estimation dynamics governed by $\frac{d}{dt}\epsilon(t) = (A - K_e(\infty)C)\epsilon$ (in steady-state case, omitting noise terms; check), stability can be checked by looking at eigenvalues of $A - K_e(\infty)C$

Equilibrium points of (linearized) dynamical systems

- *Static equilbria*: $x(t)$ does not change with time; i.e., $\frac{dx}{dt} = 0$
- For constant control u^*, the equilibrium point is where $x^* = A^{-1}Bu^*$
- More generally can have *quasistatic equilibria* where we subdivide $x(t)$ into $x_1(t)$ and $x_2(t)$, and only $\frac{dx_1(t)}{dt} = 0$ at the equilibrium; this occurs if A is singular
- Note that in general, due to insensitivity of the location of the origin, we define the origin to be the zero state vector in $\frac{dx}{dt} = Ax + Bu$, but can generalize to $x \to x + v$; simply shifts the equilibrium point by v