Optimal Control, Guidance and Estimation

Lecture – 40

Take Home Material : Summary – II

Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore

Dynamic Programming

Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore

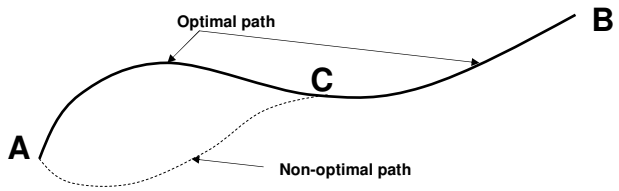
Hamilton-Jacobi-Bellman (HJB) Theory

Motivation / Objective

To obtain a "state feedback" optimal control solution

Fundamental Theorem

Any part of an optimal trajectory is an optimal trajectory!



Optimal Control Problem

Minimize $$J = \int_{t_0}^{t_f} L(t, X, U)\, dt$$

Subject to $\dot{X} = f(t, X, U)$, $X(0) = X_0$: Fixed

t_f: Fixed X_f: Free

where the input $U \in \Omega$: an admissible set

(which may be finite or infinite)

Summary of HJB Equation

- Define optimized cost function V as:

$$V(t,X) = \int_t^{t_f} L(t,X,U)\,dt$$

- Then $V(t)$ must satisfy:

HJB equation

$$\frac{\partial V}{\partial t} + H_{opt} = 0$$

where,

$$H_{opt} = \underset{U \in \Omega}{Min}(H) = \underset{U \in \Omega}{Min}\left(L + \lambda^T f\right)$$

and $\lambda \triangleq \dfrac{\partial V(t,X)}{\partial X}$

Dynamic Programming: Some Relevant Results

1. If Ω is infinite, H_{opt} can be computed by computing U (as a function of X & λ) from $\dfrac{\partial H}{\partial U} = 0$

2. Let us consider the case when t_f is fixed and X_f is free

(a) if $J = \int_{t_0}^{t_f} L(t,X,U)dt$

then $V(t_f, X_f) = \int_{t_f}^{t_f} L(t,X,U)dt = 0$

Dynamic Programming: Some Relevant Results

(b) if $J = \phi(t_f, X_f) + \int_{t_0}^{t_f} L(t, X, U)dt$

then $V(t_f, X_f) = \phi(t_f, X_f) + \int_{t_f}^{t_f} L(t, X, U)dt = \phi(t_f, X_f)$

3. If $t_f \to \infty$, $\dfrac{\partial V}{\partial t} = 0$

Dynamic Programming: Some Important Facts

- Dynamic programming is a powerful technique in the sense that if the HJB equation is solved, it leads to a “state feedback form” of optimal control solution.
- HJB equation is both necessary and sufficient for the optimal cost function.
- At least one of the control solutions that results from the solution of the HJB equation is guaranteed to be stabilizing.

Dynamic Programming: Some Important Facts

- The resulting PDE of the HJB equation is extremely difficult to solve in general.
- Dynamic Programming runs into a "huge" Computational and storage requirements for reasonably higher dimensional problems. This is a severe restriction of dynamic programming technique, which Bellman termed as ***"curse of dimensionality"***.

Approximate Dynamic Programming

Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore

Approximate Dynamic Programming: Discrete-time Framework

Problem:

Find an admissible control U_k (at time t_k) which minimizes a "meaningful" *cost function*:

$$J = \sum_{k=1}^{N-1} \Psi_k\left(X_k,\ U_k\right)$$

subjected to the constraint of *system dynamics*:

$$X_{k+1} = f\left(X_k, U_k\right)$$

and appropriate *boundary conditions*

(Note: $N \to \infty$ leads to infinite horizon problem)

Approximate Dynamic Programming: Necessary Conditions of Optimality

- Write: $J_k = \sum_{\tilde{k}=k}^{N-1} \Psi_{\tilde{k}}\left(X_{\tilde{k}},\ U_{\tilde{k}}\right) = \Psi_k + J_{k+1}$

  Cost-to-go from time t_k — Utility Function at time t_k — Cost-to-go from time t_{k+1}

- Define: $\lambda_k \triangleq \dfrac{\partial J_k}{\partial X_k}$

- ***Optimal Control Equation:*** $\boxed{\dfrac{\partial J_k}{\partial U_k} = 0}$

Approximate Dynamic Programming: Necessary Conditions of Optimality

- ***Optimal Control Equation:***

$$\boxed{\left(\frac{\partial \Psi_k}{\partial U_k}\right) + \left(\frac{\partial X_{k+1}}{\partial U_k}\right)^T \lambda_{k+1} = 0}$$

- ***Costate Equation:***

$$\lambda_k = \left(\frac{\partial J_k}{\partial X_k}\right) = \left(\frac{\partial \Psi_k}{\partial X_k}\right) + \left(\frac{\partial J_{k+1}}{\partial X_k}\right)$$

$$= \left[\left(\frac{\partial \Psi_k}{\partial X_k}\right) + \left(\frac{\partial X_{k+1}}{\partial X_k}\right)^T \lambda_{k+1}\right] + \left(\frac{\partial U_k}{\partial X_k}\right)^T \left[\left(\frac{\partial \Psi_k}{\partial U_k}\right) + \left(\frac{\partial X_{k+1}}{\partial U_k}\right)^T \lambda_{k+1}\right]$$

- ***Costate Equation on Optimal Path:***

$$\boxed{\lambda_k = \left(\frac{\partial \Psi_k}{\partial X_k}\right) + \left(\frac{\partial X_{k+1}}{\partial X_k}\right)^T \lambda_{k+1}}$$

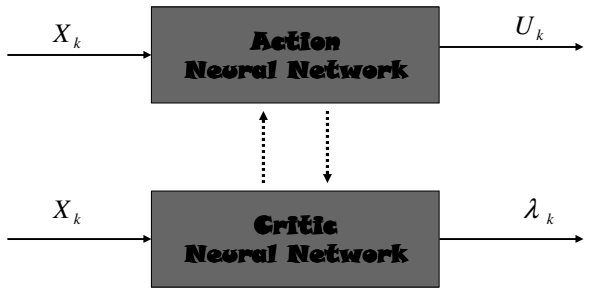
Summary: Necessary Conditions of Optimality

- State Equation: $X_{k+1} = f\left(X_k,\ U_k\right)$
- Costate Equation: $\lambda_k = g\left(X_k, \lambda_{k+1}, U_k\right)$
- Optimal Control Equation: $U_k = \Phi\left(X_k, \lambda_{k+1}\right)$
- Boundary Conditions: *TPBVP* (split)

Adaptive Critic (AC) and SNAC Designs

Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore

Adaptive Critic Methodology: Philosophy



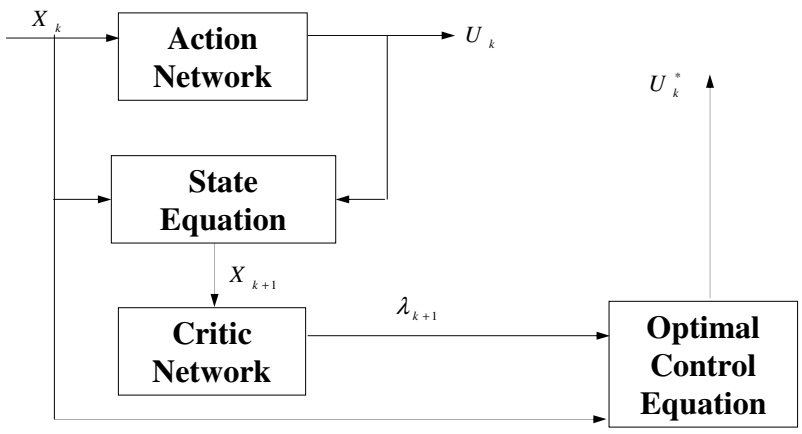
Action network leads to the optimal control solution
(after mutually consistent training of both networks)

Adaptive Critic Methodology: Advantages

- Applicable for Nonlinear problems (without any linear/quasi-linear approximations)
- Solution for a large number of initial conditions
  Feedback optimal control in the domain of interest
- Feasible computational load
  (unlike dynamic programming)
- Self-contained methodology
- Real-time control

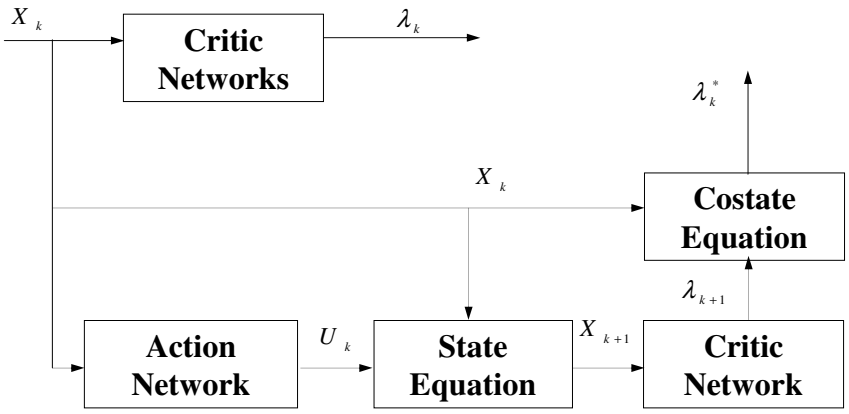
Synthesis of Action Network in AC Design

Assumption: Critic Network is Optimal



Synthesis of Critic Network in AC Design

Assumptions: *Action Network is Optimal, Critic Network is optimal at* t_{k+1}



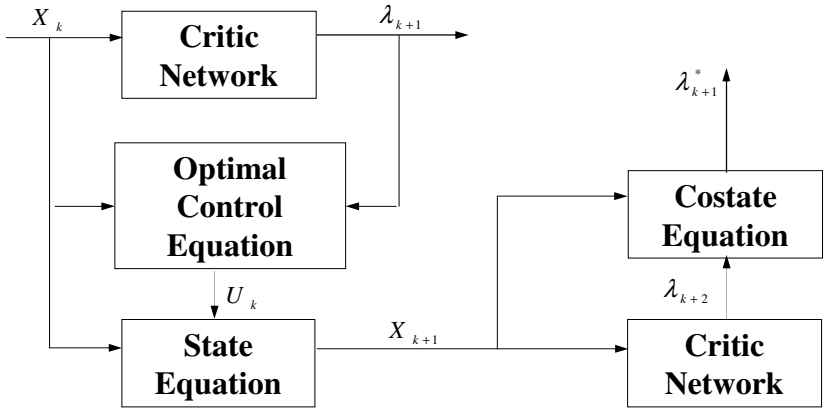
Single-Network Adaptive Critic (SNAC) Design

<u>Assumption</u>: The optimal control equation is explicitly (symbolically) solvable for control in terms of state and costate

- Retains all the advantages of adaptive-critic synthesis
- Elimination of training of action networks
- Elimination of iterative training between action and critic networks

Synthesis of Critic Network in SNAC

Assumption: *Critic Network is optimal at* t_{k+1}



Transcription Method

Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore

Philosophy of Transcription Method

- Convert the dynamic system variables into a finite set of static variables (or parameters)
- Pose an equivalent “static optimization” problem
- Solve this static optimization problem using static (paramter) optimization methods [e.g. using Nonlinear Programming (NLP)]
- Assess the accuracy
- Repeat the steps if necessary

Pseudospectral Transcription

Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore

Problem

Minimize $J = E(\mathbf{x}(t_0), \mathbf{x}(t_f)) + \int_{t_0}^{t_f} L(\mathbf{x}(t), \mathbf{u}(t))dt$

Subject to $\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t))$

with end point conditions

$$e(\mathbf{x}(t_0), \mathbf{x}(t_f)) = 0$$

and path constraints

$$h(\mathbf{x}(t), \mathbf{u}(t)) \leq 0$$

Philosophy: Discretize the states (and the control) using Pseudospectral (PS) method, Convert the problem to a "lower-dimensional" nonlinear programming (NLP) problem and solve that problem in a computationally efficient manner.

Steps

- **1. Approximate x(t) and/or u(t)?**

$$\hat{x}(t) = \sum_{n=0}^{N} a_n \phi_n(t) \qquad \hat{u}(t) = \sum_{n=0}^{N} b_n \phi_n(t)$$

- **2. Selection of grid points**
  - How are these points selected?
  - Uniform grid is not a very good choice!
- **3. Discretize the differential equation using PS method**
  - Finite-difference Vs Spectral
  - Sparse Vs Dense differentiation matrix
- **4. Approximate the integral equation**
  - Quadrature rules
- **5. Apply an efficient finite optimization technique and solve the lower dimensional NLP problem.**

Selection of Grids

Grid of collocation points (or grid points) t_n, $n=0,\ldots,N$ are points such that it satisfies the state equation exactly at these points.

Increasing Generality →

Decreasing Applicability ↓

	Free Endpoints	One Endpoint Fixed	Arbitrary Endpoints
Gauss-Lobatto	Yes	Yes	Yes
Gauss-Radau	Yes	Yes	No
Gauss	Yes	No	No
Uniform	No	No	No

Approximation of System Dynamics

Approximations: $\hat{x}(t)=\sum_{n=0}^{N} a_n\phi_n(t), \quad \hat{u}(t)=\sum_{n=0}^{N} b_n\phi_n(t)$

State Equation Constraint:

$$\dot{\mathbf{x}} = f(\mathbf{x}(t),\mathbf{u}(t))$$

$$\dot{\hat{\mathbf{x}}} = f(\hat{\mathbf{x}}(t),\hat{\mathbf{u}}(t))$$

$$\sum_{n=0}^{N}\dot{\phi}_n(t)\mathbf{a}_n = f\left(\sum_{n=0}^{N} a_n\phi_n(t), \sum_{n=0}^{N} b_n\phi_n(t)\right)$$

Multiply with $\delta(t-t_n)$ on both sides:

$$\sum_{n=0}^{N}\underbrace{\dot{\phi}_n(t_n)}_{\text{A number}}\mathbf{a}_n = f\left(\sum_{n=0}^{N} a_n\underbrace{\phi_n(t_n)}_{\text{A number}}, \sum_{n=0}^{N} b_n\underbrace{\phi_n(t_n)}_{\text{A number}}\right), \quad n=0,1,\ldots,N$$

Approximation of Cost Function

A **quadrature rule** is an approximation of the <u>definite integral</u> of a <u>function</u>, usually stated as a <u>weighted sum</u> of function values at specified points within the domain of integration.

$$\int_{-1}^{1} L(x(t),u(t))dt \approx \sum w_i L(\hat{x}(t_i),\hat{u}(t_i))$$

$$\boxed{J \cong J^N = E(\hat{\mathbf{x}}(t_N),\hat{\mathbf{u}}(t_N)) + \frac{t_f - t_0}{2}\sum_{n=0}^{N} w_n L(\hat{\mathbf{x}}(t_n),\hat{\mathbf{u}}(t_n))}$$

Summary

Approximation of State & Control: $\hat{x}(t) = \sum_{n=0}^{N} a_n \phi_n(t)$ $\quad \hat{u}(t) = \sum_{n=0}^{N} b_n \phi_n(t)$

Minimize, $\quad J^N = E(\hat{\mathbf{x}}(t_0),\hat{\mathbf{x}}(t_N)) + \frac{t_f - t_0}{2}\sum_{n=0}^{N} w_n L(\hat{\mathbf{x}}(t_n),\hat{\mathbf{u}}(t_n))$

Subject to, $\quad \sum_{n=0}^{N} \underbrace{\dot{\phi}_n(t_n)}_{\text{A number}} \mathbf{a}_n = f(\hat{\mathbf{x}}(t_n),\hat{\mathbf{u}}(t_n)) \qquad 0 \le n \le N$

with end point conditions, $\quad e(\hat{\mathbf{x}}(t_0),\hat{\mathbf{x}}(t_N)) = 0$

and path constraints, $\quad h(\hat{\mathbf{x}}(t_n),\hat{\mathbf{u}}(t_n)) \le 0 \qquad 0 \le n \le N$

The optimal control problem has been simplified to a lower dimensional nonlinear programming problem.

MPSP Design

Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore

MPSP Design

System dynamics:

$$\dot{X} = f(X, U) \qquad Y = h(X) \quad \xrightarrow{\text{Discretized}} \quad X_{k+1} = F_k(X_k, U_k) \qquad Y_k = h(X_k)$$

Goal: $Y_N \to Y_N^*$ with additional (optimal) objective(s)

Objective: $\Delta Y_N \triangleq \left(Y_N - Y_N^*\right) \to 0$

MPSP Design

Minimize:

$$J_k = \frac{1}{2}\sum_{\tilde{k}=k}^{N-1}\left(U_{\tilde{k}}^0 - dU_{\tilde{k}}\right)^T R_{\tilde{k}}\left(U_{\tilde{k}}^0 - dU_{\tilde{k}}\right)$$

Subject to:

$$B_k dU_k + \quad \cdots \quad + B_{N-1} dU_{N-1} = dY_N$$

Augmented Cost Function:

$$\bar{J}_k = \frac{1}{2}\sum_{\tilde{k}=k}^{N-1}\left(U_{\tilde{k}}^0 - dU_{\tilde{k}}\right)^T R_{\tilde{k}}\left(U_{\tilde{k}}^0 - dU_{\tilde{k}}\right) + \lambda^T\left(\sum_{\tilde{k}=k}^{N-1} B_{\tilde{k}}\, dU_{\tilde{k}} - dY_N\right)$$

MPSP Design:
Mathematical Formulation

Necessary Conditions of Optimality:

$$\frac{\partial \bar{J}_k}{\partial\, dU_{\hat{k}}} = -R_{\hat{k}}\left(U_{\hat{k}}^0 - dU_{\hat{k}}\right) + B_{\hat{k}}^T \lambda = 0$$

$$\text{for } \hat{k} = k, (k+1), \cdots, (N-1)$$

$$\frac{\partial \bar{J}_k}{\partial \lambda} = 0 \quad \Rightarrow \quad dY_N = \sum_{\tilde{k}=k}^{N-1} B_{\tilde{k}}\, dU_{\tilde{k}}$$

MPSP Design:
Mathematical Formulation

Control Update:

$$U_k = \left(U_k^0 - dU_k\right) = R_k^{-1} B_k^T \lambda$$
$$\vdots$$
$$U_{N-1} = \left(U_{N-1}^0 - dU_{N-1}\right) = R_{N-1}^{-1} B_{N-1}^T \lambda$$

where $\lambda = A_\lambda^{-1}\left(dY_N - b_\lambda\right)$

$$A_\lambda \triangleq -\left(B_k R_k^{-1} B_k^T + \cdots + B_{N-1} R_{N-1}^{-1} B_{N-1}^T\right)$$
$$b_\lambda \triangleq \left(B_k U_k^0 + \cdots + B_{N-1} U_{N-1}^0\right)$$

Iteration unfolding: Update the remaining control history "*only once*" at time step *k* and go to *k+1*

MPSP Design: Reasons for Computational Efficiency

- ***Costate variable becomes "static"; i.e. only one time-independent (constant) costate vector is needed for the entire control history update!***
- Dimension of costate vector is same as the dimension of the output vector (which is much lesser than the number of states)
- The costate vector is computed *symbolically.*
- Leads to *closed form* control history update.
- The computations needed include sensitivity matrices, which are computed "recursively".

Model Predictive Spread Control (MPSC)

Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore

MPSC with Linear Parameterization of Control

- **Parameterize control as a linear function of** t_{go}

$$U_k^0 = a_0 t_{go_k} + b_0, \quad U_k = a t_{go_k} + b$$
$$dU_k = \left(U_k^0 - U_k\right) = \underbrace{(\Delta a)}_{a_0 - a} t_{go_k} + \underbrace{(\Delta b)}_{b_0 - b}$$

- **Carryout a sensitivity analysis of the output error with respect to the error in the control history**

$$\begin{aligned} dY_N &= B_1 dU_1 + \cdots\cdots + B_{N-1} dU_{N-1} \\ &= \underbrace{\left(B_1 t_{go_1} + \cdots + B_{N-1} t_{go_{N-1}}\right)}_{C_y} \Delta a + \cdots + \underbrace{\left(B_1 + \cdots + B_{N-1}\right)}_{D_y} \Delta b \\ &= C_y \Delta a + D_y \Delta b \end{aligned}$$

Note: $B_{N-1} \cdots\cdots B_1$ can be computed recursively!

MPSC with Linear Parameterization of Control

- Formulate an optimization problem

$$\text{Optimize } J = \frac{1}{2}\left(a^T R_1 a + b^T R_2 b\right) \text{ subject to}$$

$$C_y a + D_y b = \left(-dY_N + C_y a_0 + D_y b_0\right) \triangleq K_y$$

- Solve this optimization problem in closed form

$$a = -R_1^{-1} C_y^T \lambda$$

$$b = -R_2^{-1} D_y^T \lambda$$

$$\text{where } \lambda = -\left(C_y R_1^{-1} C_y^T + D_y R_2^{-1} D_y^T\right)^{-1} K_y$$

MPSC with Quadratic Parameterization of Control

Control Parameterization

$$\begin{aligned} U_k &= at_k^2 + bt_k + c \\ U_k &= U_k^0 - dU_k \end{aligned}$$

Error in control

$$\begin{aligned} dU_k &= U_k^0 - U_k \\ &= (a_0 t_k^2 + b_0 t_k + c_0) - (at_k^2 + bt_k + c) \\ &= (a_0 - a)t_k^2 + (b_0 - b)t_k + (c_0 - c) \end{aligned}$$

Substituting for *dU_k for k = 1,.....,N-1 in*

$$\begin{aligned} dY_N &= B_1 dU_1 + B_2 dU_2 + \ldots + B_{N-1} dU_{N-1} \\ &= \sum_{k=1}^{N-1} B_k dU_k \end{aligned}$$

MPSC with Quadratic Parameterization of Control

one gets

$$B_\lambda - \left(\sum_{k=1}^{N-1} B_k t_k^2\right) a - \left(\sum_{k=1}^{N-1} B_k t_k\right) b - \left(\sum_{k=1}^{N-1} B_k\right) c = dY_N$$

$$C_y a + D_y b + E_y c = B_\lambda - dY_N$$

where

$$\begin{aligned} B_\lambda &= \left(B_1 U_1^0 + B_2 U_2^0 + \ldots B_{N-1} U_{N-1}^0\right) \\ C_y &= \left(\sum_{k=1}^{N-1} B_k t_k^2\right) \\ D_y &= \left(\sum_{k=1}^{N-1} B_k t_k\right) \\ E_y &= \left(\sum_{k=1}^{N-1} B_k\right) \end{aligned}$$

MPSC with Quadratic Parameterization of Control

- If number of equations is same as number of unknowns, then

$$[C_y \ D_y \ E_y] \begin{bmatrix} a \\ b \\ c \end{bmatrix} = B_\lambda - dY_N \qquad \begin{bmatrix} a \\ b \\ c \end{bmatrix} = [C_y \ D_y \ E_y]^{-1} [B_\lambda - dY_N]$$

- **if number of unknowns is greater than the number of equations, then optimal solution can be obtained by minimizing the following objective (cost) function**

$$J = \frac{1}{2}\left(a^T R_1 a + b^T R_2 b + c^T R_3 c\right)$$

Generalized Model Predictive Static Programming (G-MPSP)

Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore

GMPSP Design: An Overview

System dynamics:

$$\dot{X} = f(X, U)$$
$$Y = h(X)$$

where, $X \in \Re^n, \quad U \in \Re^m, \quad Y \in \Re^p$

Goal: $Y(t_f) \rightarrow Y^*(t_f)$ with additional (optimal) objective(s)

Analysis of output Error: $\delta Y(t_f) = \int_{t_0}^{t_f} \left[B(t)\, \delta U(t) \right] dt$

Recursive Relation for Computation of Sensitivity Matrices

- General formula for Recursive Computation:

$$B(t) = W(t)\frac{\partial f\left(X(t),U(t)\right)}{\partial U(t)}$$

$$\dot{W}(t) = -W(t)\left(\frac{\partial f\left(X(t),U(t)\right)}{\partial X(t)}\right)$$

$$W(t_f) = \frac{\partial Y\left(X(t_f)\right)}{\partial X(t_f)}$$

GMPSP Design: Mathematical Formulation

Minimize: $$J_c = \frac{1}{2}\int_{t_0}^{t_f}\left[\left(U^0(t)-\delta U(t)\right)^T R(t)\left(U^0(t)-\delta U(t)\right)\right]dt$$

Subject to: $$\delta Y(t_f) = \int_{t_0}^{t_f}\left[B(t)\,\delta U(t)\right]dt$$

Augmented Cost Function:

$$\bar{J}_c = \frac{1}{2}\int_{t_0}^{t_f}\left[\left(U^0(t)-\delta U(t)\right)^T R(t)\left(U^0(t)-\delta U(t)\right)\right]dt + \lambda^T\left[\delta Y(t_f) - \int_{t_0}^{t_f}\left[B(t)\,\delta U(t)\right]dt\right]$$

GMPSP Design:
Mathematical Formulation

Necessary Conditions of Optimality:

$$\frac{\partial \bar{J}_c}{\partial\left(\delta U(t)\right)} = -R(t)\left(U^0(t) - \delta U(t)\right) - \left(B(t)\right)^T \lambda = 0$$

$$\frac{\partial \bar{J}_c}{\partial \lambda} = 0 \quad \Rightarrow \quad \delta Y\left(t_f\right) = \int_{t_0}^{t_f} \left[B(t)\,\delta U(t)\right] dt$$

GMPSP Design:
Mathematical Formulation

Control Solution:

$$\delta U(t) = \left(R(t)\right)^{-1}\left(B(t)\right)^T \lambda + U^0(t)$$

$$\delta Y\left(t_f\right) = \int_{t_0}^{t_f} \left[B(t)\,\delta U(t)\right] dt$$

$$\lambda = \left(A_\lambda\right)^{-1}\left[\delta Y\left(t_f\right) - b_\lambda\right]$$

where

$$A_\lambda \triangleq \int_{t_0}^{t_f} \left[B(t)\left(R(t)\right)^{-1} B^T(t)\right] dt$$

$$b_\lambda \triangleq \int_{t_0}^{t_f} \left[B(t)\,U^0(t)\right] dt$$

GMPSP Design:
Mathematical Formulation

Control Update:

$$
\begin{aligned}
U(t) &= U^0(t) - \delta U(t) \\
&= U^0(t) - \left(R(t)\right)^{-1}\left(B(t)\right)^T\left[\left(A_\lambda\right)^{-1}\left\{\delta Y\left(t_f\right) - b_\lambda\right\}\right] - U^0(t) \\
&= -\left(R(t)\right)^{-1}\left(B(t)\right)^T\left[\left(A_\lambda\right)^{-1}\left\{\delta Y\left(t_f\right) - b_\lambda\right\}\right]
\end{aligned}
$$

where $\lambda = \left(A_\lambda\right)^{-1}\left[\delta Y\left(t_f\right) - b_\lambda\right]$

LQ Observer Design

Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore

Observer Design for Linear Systems

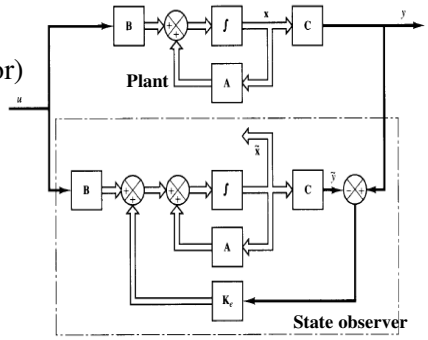
Plant : $\dot{X} = AX + BU$

$Y = CX$ (sensor output vector)

Observer Dynamics :

$$\dot{\hat{X}} = A\hat{X} + BU + K_e\left(Y - C\hat{X}\right)$$

Error : $\tilde{X} \triangleq \left(X - \hat{X}\right)$



Ref: K. Ogata: *Modern Control Engineering*, 3rd Ed., Prentice Hall, 1999.

Comparison of Control and Observer Design Philosophies

Control Design	Observer Design
• CL Dynamics $\dot{X} = \left(A - BK\right)X$	• CL Error Dynamics $\dot{\tilde{X}} = \hat{A}\tilde{X} = \left(A - K_eC\right)\tilde{X}$
• Objective $X\left(t\right) \to 0, \quad \text{as} \quad t \to \infty$	• Objective $\tilde{X}\left(t\right) \to 0, \quad \text{as} \quad t \to \infty$
	• Notice that $\lambda\left(A - K_eC\right) = \lambda\left[\left(A - K_eC\right)^T\right]$ $= \lambda\left(A^T - C^TK_e^T\right)$

Algebraic Riccati Equation (ARE) Based Observer Design

System	Dual System
$\dot{X} = AX + BU$	$\dot{Z} = A^T Z + C^T V$
$Y = CX$	$n = B^T Z$
$M = \left[B \middle\vert AB \middle\vert \cdots \middle\vert A^{n-1}B \right]$	$M = \left[C^T \middle\vert A^T C^T \middle\vert \cdots \middle\vert A^{T^{n-1}} C^T \right]$
$N = \left[C^T \middle\vert A^T C^T \middle\vert \cdots \middle\vert A^{T^{n-1}} C^T \right]$	$N = \left[B \middle\vert AB \middle\vert \cdots \middle\vert A^{n-1}B \right]$
LQR Design	
$U = -KX$	

ARE Based Observer Design

CL system (control design)

$$\dot{X} = (A - BK)X$$

$$X \to 0 \quad as \quad t \to \infty$$

$$K = R^{-1}B^T P, \quad P > 0$$

where,

$$PA + A^T P - PBR^{-1}B^T P + Q = 0$$

Error Dynamics

$$\dot{\tilde{X}} = (A - K_e C)\tilde{X}$$

$$(A - K_e C)^T = A^T - C^T K_e^T$$

Acts like a controller gain

Analogous

$$K_e^T = R^{-1}CP$$

where,

$$PA^T + AP - PC^T R^{-1} CP + Q = 0$$

Observer Dynamics

$$\dot{\hat{X}} = A\hat{X} + BU + K_e(Y - C\tilde{X})$$

Kalman Filter Design for Linear Time-Invariant (LTI) Systems

Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore

Kalman Filter Design for LTI Systems

System Model: $\dot{X} = AX + BU + GW$

Measured output: $Y = CX + V$

$X(t_0) \sim \left(\tilde{X}_0, P_0\right), \quad W(t) \sim (0, Q), \quad V(t) \sim (0, R)$

Assumptions: $W(t), V(t)$ are white noises

$W(t), V(t), X(t_0)$ are mutually orthogonal

Problem: Define $\tilde{X}(t) \triangleq \left[X(t) - \hat{X}(t)\right]$

Find $\hat{X}(t)$ such that $P = \lim_{t \to \infty} E\left[\tilde{X}\tilde{X}^T\right]$ is minimized.

Kalman Filter Design for LTI Systems

Filter Operation:

(i) Initialize $\hat{X}(0)$

(ii) Solve for Riccati matrix P from the Filter ARE:

$$AP + PA^T - PC^T R^{-1} CP + G\,Q\,G^T = 0$$

(iii) Compute Kalman Gain:

$$K_e = PC^T R^{-1}$$

(iv) Propagate the Filter dynamics:

$$\dot{\hat{X}} = A\hat{X} + BU + K_e\left(Y - C\hat{X}\right)$$

where Y is the measurement vector

Kalman Filter Design for Linear Time-Varying Systems

Prof. Radhakant Padhi

Dept. of Aerospace Engineering

Indian Institute of Science - Bangalore

Kalman Filter for Linear Time-Varying Systems

System Dynamics: $\dot{X} = A(t)X + B(t)U + G(t)W$ $\quad W(t)$: Process noise

Measured Output: $Y = C(t)X + V$ $\quad V(t)$: Sensor noise

Assumptions:

(*i*) $X(0) \sim \left(\tilde{X}_0, P_0\right)$, $W(t) \sim \left(0, Q(t)\right)$ and $V(t) \sim \left(0, R(t)\right)$ are "mutually orthogonal" $\left[X(0): \text{ initial condition for } X\right]$

(*ii*) $W(t)$ and $V(t)$ are uncorrelated non-stationary white noise

(*iii*) $E\left[W(t)\, W^T(t+\tau)\right] = Q(t)\,\delta(t-\tau), \quad Q \geq 0$ (psdf)

$E\left[V(t)\, V^T(t+\tau)\right] = R(t)\,\delta(t-\tau), \quad R > 0$ (pdf)

$E\left[V(t)\, W^T(\tau)\right] = 0$

Kalman Filter for Linear Time-Varying Systems

Optimization Problem :

Minimize $J = \frac{1}{2} Tr\left(\dot{P}\right)$ with appropriate choice of K_e

Solution :

$$\frac{\partial J}{\partial K_e} = \frac{\partial}{\partial K_e}\left[\frac{1}{2} Tr\left(\dot{P}\right)\right]$$

$$= \frac{1}{2}\frac{\partial}{\partial K_e}\left[Tr\left(K_e R K_e^T\right) - 2Tr\left(K_e C P\right)\right]$$

$$= K_e(t)R(t) - P(t)C^T(t) = 0$$

Hence $\boxed{K_e(t) = P(t)C^T(t)R^{-1}(t)}$

Kalman Filter for Linear Time-Varying Systems

(i) Initialize $\hat{X}(0)$

(ii) Propagate $P(t)$ from the Filter Riccati Equation:

$$\dot{P}(t) = A(t)P(t) + P(t)A(t)^T - P(t)C^T(t)R^{-1}(t)C(t)P(t) + G(t)Q(t)G(t)^T$$

with $P(0) = E\left[\tilde{X}(t_0)\tilde{X}^T(t_0)\right]$

(iii) Compute Kalman Gain:

$$K_e = PC^T R^{-1}$$

(iv) Propagate the Filter dynamics:

$$\dot{\hat{X}} = A\hat{X} + BU + K_e\left(Y - C\hat{X}\right)$$

where Y is the measurement vector

Discrete-time Kalman Filter (DKF)

Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore

Discrete-time Kalman Filter

Model	$X_{k+1} = A_k X_k + B_k U_k + G_k W_k$ $Y_k = C_k X_k + V_k$
Initialization	$\hat{X}(t_0) = \hat{X}_0^-$ $P_0^- = E\left[\tilde{X}_0^- \tilde{X}_0^{-T}\right]$
Gain Computation	$K_{e_k} = P_k^- C_k^T \left[C_k P_k^- C_k^T + R_k\right]^{-1}$

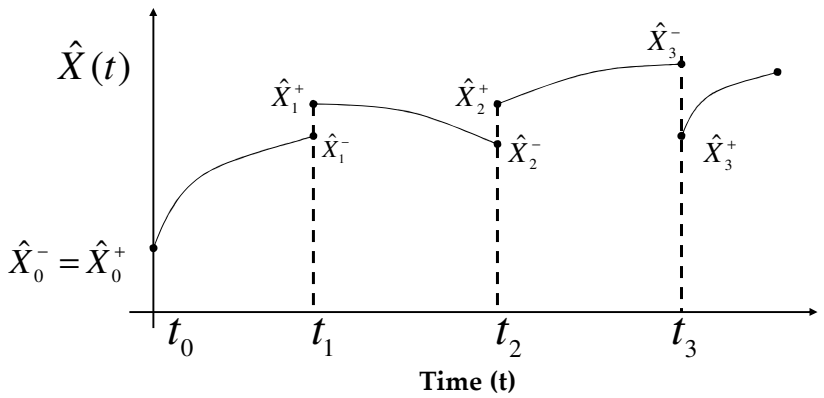
Discrete-time Kalman Filter

Updation	$\hat{X}_k^+ = \hat{X}_k^- + K_{e_k}\left[Y_k - C_k \hat{X}_k^-\right]$ $P_k^+ = \left(I - K_{e_k} C_k\right) P_k^- \left(I - K_{e_k} C_k\right)^T + K_{e_k} R_k K_{e_k}^T$ (preferable) $= \left(I - K_{e_k} C_k\right) P_k^-$ (not preferable)
Propagation	$\hat{X}_{k+1}^- = A_k \hat{X}_k^+ + B_k U_k$ $P_{k+1}^- = A_k P_k^+ A_k^T + G_k Q_k G_k^T$

Continuous-Discrete Kalman Filter

Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore

Mechanism



Continuous-Discrete Kalman Filter

Model	$\dot{X}(t) = A(t)X(t) + B(t)U(t) + G(t)W(t)$ $Y_k = C_k X_k + V_k$
Initialization	$\hat{X}(t_0) = \hat{X}_0$ $P_0^- = E\left[\tilde{X}(t_0)\,\tilde{X}^T(t_0)\right]$
Gain Computation	$K_{e_k} = P_k^- C_k^T \left[C_k P_k^- C_k^T + R_k\right]^{-1}$

Continuous-Discrete Kalman Filter

Updation	$\hat{X}_k^+ = \hat{X}_k^- + K_{e_k}\left[Y_k - C_k \hat{X}_k^-\right]$ $P_k^+ = \left(I - K_{e_k} C_k\right) P_k^- \left(I - K_{e_k} C_k\right)^T + K_{e_k} R_k K_{e_k}^T$ (preferable) $= \left(I - K_{e_k} C_k\right) P_k^-$ (not preferable)
Propagation (using high accuracy numerical integration)	$\dot{\hat{X}} = A\hat{X} + BU$ $\dot{P}(t) = AP + PA^T + GQG^T$

Continuous-Continuous EKF

Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore

Continuous-Continuous EKF

Model	$\dot{X}(t) = f(X, U, t) + G(t)W(t)$ $Y = h(X, t) + V(t)$
Initialization	$\hat{X}(t_0) = \hat{X}_0$ $P_0 = E\left[\tilde{X}(t_0)\tilde{X}^T(t_0)\right]$
Gain Computation	$K_e(t) = P(t)C^T(t)R^{-1}(t)$

Continuous-Continuous EKF

Propagation	$\dot{\hat{X}}(t) = f(\hat{X}, U, t) + K_e(t)\left[Y - h(\hat{X}, t)\right]$ $\dot{P}(t) = AP + PA^T - PC^T R^{-1} CP + GQG^T$ where, $A(t) = \left[\dfrac{\partial f}{\partial X}\right]_{\hat{X}}$; $C(t) = \left[\dfrac{\partial h}{\partial X}\right]_{\hat{X}}$

Continuous-Discrete EKF

Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore

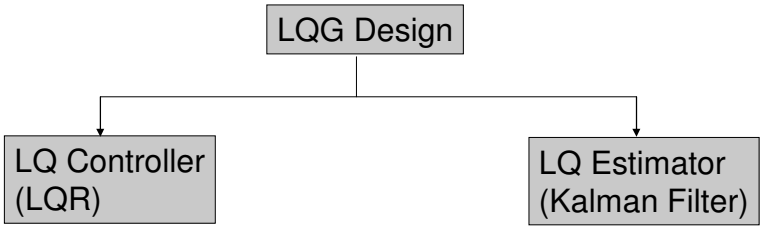
Continuous-Discrete EKF

Model	$\dot{X}(t) = f(X, U, t) + G(t)W(t)$ $Y = h(X_k) + V_k$
Initialization	$\hat{X}^-(t_0) = X_0$ $P_0^- = E\left[\tilde{X}^-(t_0)\,\tilde{X}^{-T}(t_0)\right]$
Gain Computation	$K_{e_k}(t) = P_k^- C_k^{-T}\left[C_k^- P_k^- C_k^{-T} + R\right]^{-1}$ where, $C_k^- = \left[\dfrac{\partial h}{\partial X}\right]_{\hat{X}_k^-}$

Continuous-Discrete EKF

Updation	$\hat{X}_k^+ = \hat{X}_k^- + K_{e_k}\left[Y_k - h\left(\hat{X}_k^-\right)\right]$ $P_k^+ = \left(I - K_{e_k}C_k\right)P_k^-\left(I - K_{e_k}C_k\right)^T + K_{e_k}R_k K_{e_k}^T$ (preferable) $= \left(I - K_{e_k}C_k\right)P_k^-$ (not preferable)
Propagation	$\dot{\hat{X}}(t) = f(\hat{X}, U, t); \qquad \hat{X}_k^+ \rightarrow \hat{X}_{k+1}^-$ $\dot{P}(t) = AP + PA^T + GQG^T;\ \hat{P}_k^+ \rightarrow \hat{P}_{k+1}^-$ where $A(t) = \left[\dfrac{\partial f}{\partial X}\right]_{\hat{X}(t)}$

Philosophy of LQG Design



- Controller: Linear Quadratic Regulator (LQR)
- State Estimation: Kalman Filter

LQG Design

- Design a deterministic LQR control $U = -K\,X$, assuming perfect knowledge of the states and assuming that the plant is not affected by process and sensor noises.
- Design a Kalman Filter to estimate the states and compute the control using this estimated states $U = -K\,\hat{X}$. This design philosophy is called Linear Quadratic Gaussian (LQG) design.
- Justification for the LQG design comes from the "**Separation Principle**".

Control Constrained Optimal Control

Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore

Objective

To find an "admissible" time history of control variable $U(t), t \in [t_0, t_f]$, where $\|U(t)\| \le \mathbf{U}$ $\left(\textit{or, component wise, } U_j^- \le u_j(t) \le U_j^+\right)$, which:

1) Causes the system governed by $\dot{X} = f(t, X, U)$ to follow an admissible trajectory

2) Optimizes (minimizes/maximizes) a "meanigful" performance index

$$J = \varphi(t_f, X_f) + \int_{t_0}^{t_f} L(t, X, U)\, dt$$

3) Forces the system to satisfy "proper boundary conditions".

Solution Procedure of a given Problem

Hamiltonian : $H(X,U,\lambda) = L(X,U) + \lambda^T f(X,U)$

Necessary Conditions :

(i) State Equation: $\dot{X} = \left(\frac{\partial H}{\partial \lambda}\right) = f(t,X,U)$

(ii) Costate Equation: $\dot{\lambda} = -\left(\frac{\partial H}{\partial X}\right)$

(iii) Optimal Control Equation: Minimize H with repect to $U(t) \leq \mathbf{U}$

i.e. $H(X,U^*,\lambda) \leq H(X,U,\lambda)$

(iv) Boundary conditions:

$X(0) = \text{Specified}, \quad \lambda_f = \left(\frac{\partial \varphi}{\partial X_f}\right)$

Topics Studied in Detail

- Pontryagin Minimum Principle
- Time Optimal Control of LTI Systems
  - Time Optimal Control of Double-Integral System
- Energy Optimal Control of LTI Systems

State Constrained Optimal Control

Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore

Penalty Function Method

System Dynamics :

$$\dot{X} = f\left(X, U, t\right) \quad \text{where,} \quad X \in \mathbb{R}^n, U \in \mathbb{R}^m$$

Performance Index :

$$J = \int_{t_0}^{t_f} L\left(X, U, t\right)$$

Constraints :

$$g_1\left(x_1, x_2, \cdots, x_n, t\right) \geq 0$$

$$\cdots\cdots\cdots$$

$$g_p\left(x_1, x_2, \cdots, x_n, t\right) \geq 0, \qquad p \leq n$$

Assumption : The constraint functions have continuous first and second partial derivatives with respect to X.

Penalty Function Method

Idea : Convert inequality constraints to equality constraints

Define a new variable x_{n+1}

$$\dot{x}_{n+1} \triangleq f_{n+1}(X,t) = \left[g_1(X,t)\right]^2 h(g_1) + \cdots + \left[g_p(X,t)\right]^2 h(g_p)$$

where, $h(g_i)$ is a <u>unit Heaviside step function</u>

$$i.e.\ h(g_i) = \begin{cases} 0, \text{ if } g_i(X,t) \geq 0 \\ 1, \text{ if } g_i(X,t) \leq 0 \end{cases} \qquad \text{for } i = 1, 2, \cdots, p$$

Boundary conditions: $x_{n+1}(t_0) = x_{n+1}(t_f) = 0.$

Note : This formulation makes it an infeasible problem, unless all constraints are satisfied.

Slack Variable Method

System Dynamics :

$$\dot{X} = f(X,U,t) \qquad \text{where,} \quad X \in \mathbb{R}^n, U \in \mathbb{R}^m$$

Performance Index :

$$J = \varphi(X_f, t_f) + \int_{t_0}^{t_f} L(X,U,t)\,dt$$

Constraints :

$$S(X,t) \leq 0, \qquad \text{where } S \text{ is of } p^{\text{th}} \text{ order}$$

i.e. The control U appears explicitly in the pth order derivative of S.

Slack Variable Method

(Also known as Valentine's method; basic idea is due to F. A. Valentine)

Idea : Introduce a slack variable and write

$$S(X,t)+\frac{1}{2}\alpha^2=0$$

Differentiating upto p times with respect to t

$$S_1(X,t)+\alpha\alpha_1=0$$

$$\cdots$$

$$S_p(X,U,t)+\{\text{terms involving } \alpha,\alpha_1,\cdots,\alpha_p\}=0$$

where, subscripts on S and α denote the time derivatives.

Since U is explicitly present in p^{th} derivative, one can solve for

$$\boxed{U=g(X,\alpha,\alpha_1,\cdots,\alpha_p,t)}$$

Slack Variable Method

Hence, the system dynamics can be written as:

$$\dot{X}=f\left(X,g\left(\underbrace{X,\alpha,\alpha_1,\cdots,\alpha_{p-1}}_{\text{States}},\underbrace{\alpha_p}_{\text{Control}},t\right),t\right),\quad X(t=t_0)=X_0$$

$$\dot{\alpha}=\alpha_1,\qquad \alpha(t=t_0)=\alpha(t_0)$$

$$\dot{\alpha}_1=\alpha_2,\qquad \alpha_1(t=t_0)=\alpha_1(t_0)$$

$$\cdots\cdots\cdots\cdots$$

$$\dot{\alpha}_{p-1}=\alpha_p,\qquad \alpha_{p-1}(t=t_0)=\alpha_{p-1}(t_0)$$

α_p : Control variable **(unconstrained)**

Slack Variable Method

Initial Conditions

We know: $S(X,t)+\frac{1}{2}\alpha^2=0$

$$S_1(X,t)+\alpha\alpha_1=0$$

$$S_2(X,t)+\alpha_1^2+\alpha\alpha_2=0$$

$$\cdots\cdots$$

Hence, substituting $t=t_0$,

$$\alpha(t_0)=\pm\sqrt{-2g(X_0,t_0)}$$

$$\alpha_1(t_0)=-g_1(X_0,t_0)/\alpha_0$$

$$\alpha_2(t_0)=-\left[g_2(X_0,t_0)+\alpha_1^2(t_0)\right]/\alpha_0$$

$$\cdots\cdots$$

Slack Variable Method

Cost function :

$$J=\varphi\left(X_f,t_f\right)+\int_{t_0}^{t_f}L\left(X,g\left(X,\alpha,\alpha_1,\cdots,\alpha_{p-1},\alpha_p,t\right),t\right)dt$$

New Problem (in $n+p$ dimension) :

State Vector: $Z\triangleq\left[X,\alpha,\alpha_1,\cdots,\alpha_{p-1}\right]^T$, New Control: α_p

System Dynamics:

$$\dot{Z}=F\left(Z,\alpha_p,t\right),\qquad Z(t_0)=Z_0:\ \text{Available}$$

Cost Function:

$$J=\phi\left(Z_f,t_f\right)+\int_{t_0}^{t_f}L\left(Z,\alpha_p,t\right)dt$$

Optimal Control of Distributed Parameter Systems

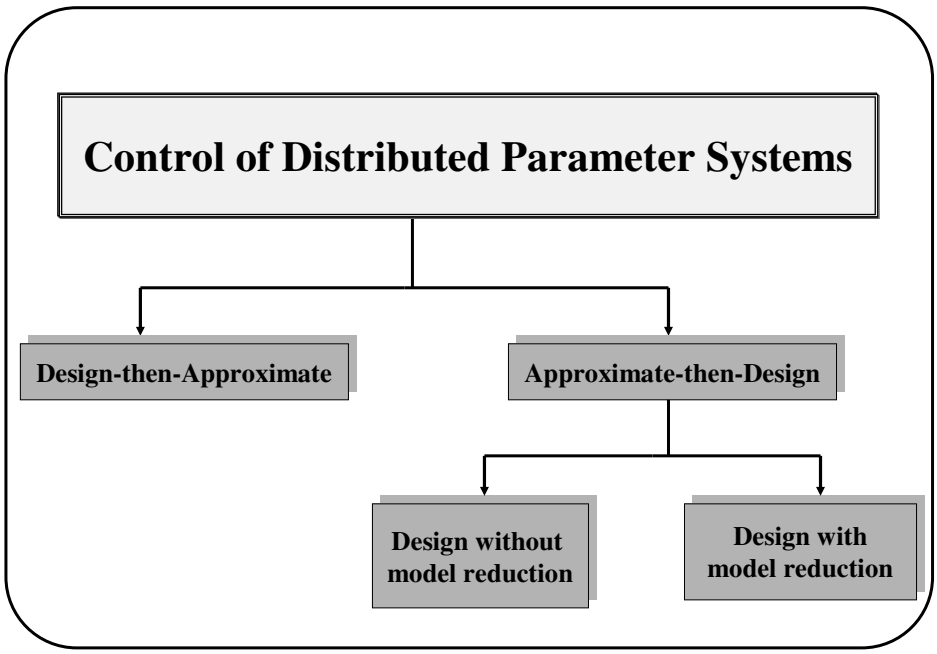
Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore

Distributed Parameter Systems (DPS)

Systems are Governed by a Set of
Partial Differential Equations

Examples:

- Heat Transfer Processes
- Fluid Flows
- Chemical Reactor Processes
- Vibration of Structures (Aeroelastic Problems)
- Ecological Problems



Topics Covered

- LQR for DPS
  - Using Finite Difference (through examples)
- Optimal Dynamic Inversion
  - Continuous Actuator
  - Set of Discrete Actuators
- SNAC
  - Using Finite Difference
  - Using Proper Orthogonal Decomposition (POD)
- Examples

Concluding Remarks: Optimal Control

- A variety of difficult real-life problems can be formulated in the framework of optimal control. Incorporation of optimal issues lead to a variety of advantages, like minimum cost, maximum efficiency, non-conservative design etc.
- Modern techniques are capable of addressing the fundamental issue of *"computational complexity".* Advances in computational power is a good advantage as well.
- A variety of classical and advanced optimal control techniques (both for linear and nonlinear systems) have been covered in this course.

Concluding Remarks: Estimation Theory

- State feedback control designs need the state information for control computation. Estimation theory enables it.
- Auxiliary information necessary (like target and obstacle information) can be collected using estimation theory.
- Many other applications (like parameter identification, fault diagnosis etc. are possible with the estimation theory).
- Nonlinear estimation theory is a nice combination of scientific and heuristic thoughts.
- Kalman Fitering, which is most commonly used in practice, has been covered in this course. Both basic fundamentals and advanced topics have been covered.

Thanks for the Attention....!!

Questions ... ??