Dynamic Programming of Pneumatic Systems

Takayuki Nishi
FL10_25_02
4th , February, 2011

Introduction (Pneumatic Systems)

Pneumatic Systems

Systems actuated by the air

Characteristic

- Low acquisition cost
- High power
- Lower weight

BUT … Rarely used for servo drive

- Difficult controllability due to the nonlinearities
  - Compressibility of the air
  - Flow characteristic of the servo valve
  - Friction by the cylinder

Introduction

Work about pneumatic systems

- Modeling and identification
- Control of pneumatic actuators

Optimization methods for the control

Some optimization methods are conducted

Test dynamic programming for pneumatic systems

- Used for nonlinear systems
- Able to consider constraints

Objective

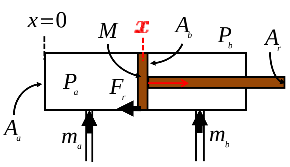
Apply dynamic programming algorithm to pneumatic actuators

Outline

- Introduction
- System Description
- Dynamic Programming Algorithm
- Simulations
- Conclusion

System Parameters

The model of the pneumatic actuator



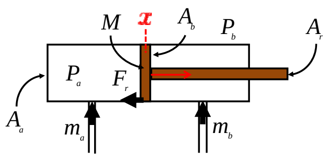
- M : Mass of the piston and the rod
- $A_{(a,b)}$: Area of each of the piston
- A_r : Cross-sectional area of the piston rod
- $V_{(a,b)}$: Volume of each of the piston

Air : polytropic process (Ideal gas)

$$PV^n = \text{const}$$

- n : Polytropic coefficient
- x : Position of the piston
- $P_{(a,b)}$: Pressure of the chamber
- F_r : Friction force
- $\dot{m}_{(a,b)}$: Mass flow rate of each side of the cylinder

System Dynamics



- P_{atm} : Pressure of the atmosphere
- R : Gas constant
- T_0 : Fluid temperature

Motion dynamics of the piston

$$M\ddot{x} + F_r = P_a A_a + P_b A_b - P_{atm} A_r \quad (1)$$

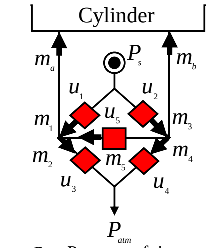
Change of the pressure inside the chambers

$$\dot{P}_{(a,b)} = \frac{nRT_0}{V_{(a,b)}}\dot{m}_{(a,b)} - \frac{nP_{(a,b)}}{V_{(a,b)}}\dot{V}_{(a,b)} \quad (2)$$

State vector

$$\mathbf{x}^T = \begin{bmatrix} x & \dot{x} & P_a & P_b \end{bmatrix} \quad (3)$$

System Inputs



P_s : Pressure of the supply

P_{atm} : Pressure of the atmosphere

Inputs: Valve signals $u_i \in \{0,1\}$

i : Number of the valves

Valve configuration

$$\dot{m}_a = \dot{m}_1 - \dot{m}_3$$
$$\dot{m}_b = \dot{m}_2 - \dot{m}_4 \quad (4)$$

Consider 4 valve configuration

$$\mathbf{u}(t) = [\ u_1 \quad u_2 \quad u_3 \quad u_4\ ]^T$$

Cost function: Consumption of the air

$$H(\mathbf{x}(t), \mathbf{u}(t), t) = \dot{m}_1 + \dot{m}_2$$

State equation

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

Control objective

- Reach the states to desired values
- Save the air consumption

Outline

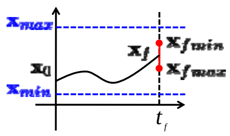
- Introduction
- System Description
- Dynamic Programming Algorithm
- Simulations
- Conclusion

Control Problem

Discretized model $\mathbf{x}_{k+1} = F(\mathbf{x}_k, \mathbf{u}_k)t_s + \mathbf{x}_k$ t_s : Step time

$$\mathbf{x}_{k+1} = F_k(\mathbf{x}_k, \mathbf{u}_k) \qquad k = 0, 1, \cdots, N-1$$

with constraints $\mathbf{x}_k \in \mathcal{X}_k \quad \mathbf{u}_k \in \mathcal{U}_k$



Control policy

$$\pi = \{\mu_0, \cdots, \mu_{N-1}\}$$
$$\mathbf{u}_k = \mu_k(\mathbf{x}_k)$$

Discretized overall cost using π with the initial state $\mathbf{x}_0$ t_f : Final time

$$J_\pi(\mathbf{x}_0) = g_N(\mathbf{x}_N) + \sum_{k=0}^{N-1} (h_k(\mathbf{x}_k, \mu_k(\mathbf{x}_k)) + \phi_k(\mathbf{x}_k))$$

Final cost Cost of air consumption Penalty function

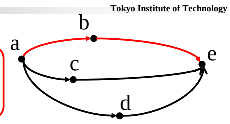
Optimal policy π^* $J_{\pi^*}(\mathbf{x}_0) = \min_{\pi \in \Pi} J_\pi(\mathbf{x}_0)$ Π : Set of admissible policies

Control objective: Find Optimal policy π^*

Procedure of Dynamic Programming

Basic idea : Principle of optimality

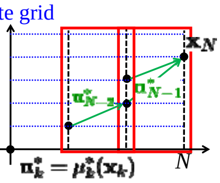
If a-b-e is the optimal path from a to e, then the path b-e is the optimal from b to e.



Procedure

1. Discretize the states
2. Evaluate the optimal cost function $\mathcal{J}_k$ for each stage k (Backward calculation)
   - Final cost: $\mathcal{J}_N(\mathbf{x}_N) = g_N(\mathbf{x}_N)$
   - Subproblem involving two stages

State grid



$$\mathcal{J}_k(\mathbf{x}_k) = \min_{\mathbf{u}_k \in \mathcal{U}_k} \{h_k(\mathbf{x}_k, \mathbf{u}_k) + \phi_k(\mathbf{x}_k) + \mathcal{J}_{k+1}(F_k(\mathbf{x}_k, \mathbf{u}_k))\}$$

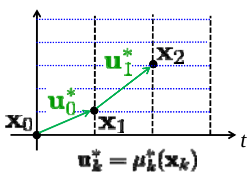
Output: Optimal control policy π^* $\pi^* = \{\mu^*_0, \cdots, \mu^*_{N-1}\}$

Procedure of Dynamic Programming

Procedure

3. Forward calculation

Apply optimal control policy π^* and generate optimal trajectories starting from $\mathbf{x}_0$



Characteristic

- Provide valuable insights about the structure of the optimal solution

Benchmark to all other controllers which can be compared

- Time consuming
- Excessive computational burden for complex problems

Depend on the number of state grids and time steps

Outline

- Introduction
- System Description
- Dynamic Programming Algorithm
- Simulations
- Conclusion

Simulation

State trajectories

Step time : $t_s = 0.005$

Number of steps : $N = 81$

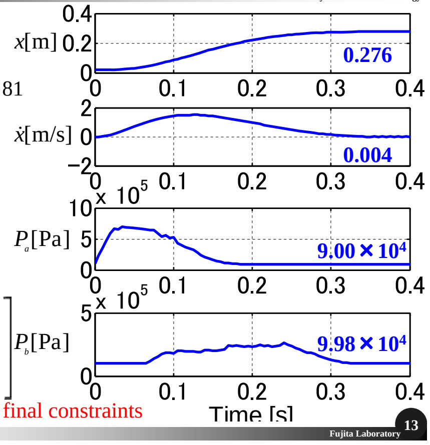
Boundary condition

$$\mathbf{x}_0 = \begin{bmatrix} 0.02 \\ 0.0 \\ P_{atm} \\ P_{atm} \end{bmatrix}$$

$P_s = 7.0 \times 10^5$ [Pa]

$P_{atm} = 1.0 \times 10^5$ [Pa]

$$\mathbf{x}_f = \begin{bmatrix} 0.27 \\ 0.0 \\ 0.9 \sim 1.1 \times 10^5 \\ 0.9 \sim 1.1 \times 10^5 \end{bmatrix}$$



Trajectories satisfy the final constraints

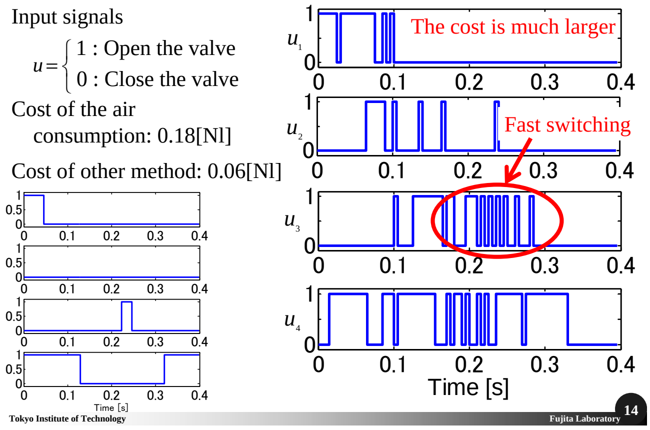
Simulation

Input signals

$$u = \begin{cases} 1 : \text{Open the valve} \\ 0 : \text{Close the valve} \end{cases}$$

Cost of the air consumption: 0.18[Nl]

Cost of other method: 0.06[Nl]



Conclusion

Summary

- Create the dynamic programming algorithm
- Apply dynamic programming algorithm to the pneumatic system
  - Get state trajectories which satisfy the final constraints
  - It isn't sure that the inputs are optimal

Future Works

- Analyze why the inputs are not optimal

References

[1] Olle Sundström and Lino Guzzella, "A Generic Dynamic Programming Matlab Function", *18th IEEE International Conference on Control Applications, Part of 2009 IEEE Multi-conference on Systems and Control*, Saint Petersburg, Russia, July, 2009

Reference for dynamic programming algorithm

[2] Xiangrong Shen and Michael Goldfarb, "Energy Saving in Pneumatic Servo Control Utilizing Interchamber Cross-Flow", *Journal of Dynamic Systems, Measurement, and Control*, 2007

[3] Xiangrong Shen and Michael Goldfarb, "Energy Saving in Pneumatic Servo Control Utilizing Interchamber Cross-Flow", *Journal of Dynamic Systems, Measurement, and Control*, 2007

Reference for understanding pneumatic systems

[Appendix] Pressure Dynamics

First law of thermodynamics

$$dU = \delta Q - \delta W$$

Internal energy Heat Work by the air (空気がした仕事)

Consider the pneumatic actuator in the slides

$$\frac{d}{dt}(c_v \rho V_{(a,b)} T_0) = c_p \dot{m}_{(a,b)} T_0 - P_{(a,b)} \frac{dV_{(a,b)}}{dt}$$

$\frac{c_p}{c_v} = n \quad c_p = \frac{nR}{n-1}$ c_v : Specific heat at constant pressure

c_p : Specific heat at constant pressure

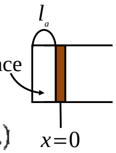
Change of the pressure inside the chambers

$$\dot{P}_{(a,b)} = \frac{nRT_0}{V_{(a,b)}}\dot{m}_{(a,b)} - \frac{nP_{(a,b)}}{V_{(a,b)}}\dot{V}_{(a,b)} \qquad (2)$$

[Appendix] System Dynamics

Friction force : Newton and Coulumb friction model

$$F_r = f_n \mathrm{sign}(\dot{x}) + f_v \dot{x}$$

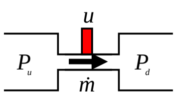


Volume of each of the piston : $V_{(a,b)}$

$$V_a = A_a(x + l_a) \quad V_b = A_b(L - x + l_b)$$

$l_{(a,b)}$: Length of the dead space

Mass flow rate : $\dot{m}$



$$\dot{m} = \psi(P_u, P_d) C_v P_u \rho_0 u$$

ρ_0 : Density of the air C_v : Sonic conductance

P_u : Pressure of the upper P_d : Pressure of the lower

ψ : Flow function b ：臨界圧力比

$$\psi(P_u, P_d) = \begin{cases} 1 & \frac{P_d}{P_u} \le b \quad \text{(chocked)} \\ \sqrt{1 - \left(\frac{P_d/P_u - b}{1-b}\right)^2} & \text{otherwise} \quad \text{(unchocked)} \end{cases}$$

[Appendix] System Parameters

Value of simulation parameters

$M = 5[\mathrm{kg}]$ | $f_n = 13.54[\mathrm{N}]$

$A_a = 201[\mathrm{mm}^2]$ | $f_v = 10.94[\mathrm{N \cdot s/m}]$

$A_b = 173[\mathrm{mm}^2]$ | $\rho = 1.189[\mathrm{kg/m}^3]$

$A_r = 28[\mathrm{mm}^2]$ | $R = 288[\mathrm{J/kg \cdot K}]$

$l = 0.32[\mathrm{m}]$: Length of the cylinder | $C_v = 0.3e-8[\mathrm{m}^3/\mathrm{s \cdot Pa}]$

$l_a = 0.035[\mathrm{m}]$ | $T_0 = 293[\mathrm{K}]$

$l_a = 0.109[\mathrm{m}]$ | $b = 0.5$

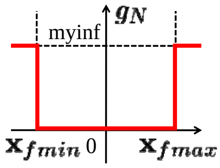
$n = 1.25$

[Nl]とは基準状態(0℃,1atm {大気圧})での体積

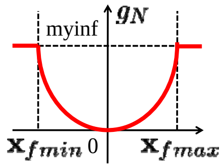
[Appendix] Cost Function

Final cost: $g_N(\mathbf{x}_N)$ myinf = 1000

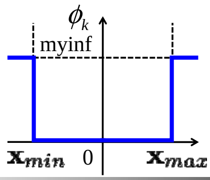
Option1: Step function



Option2: Quadratic function

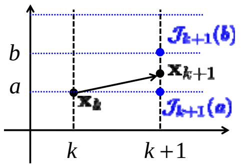


Penalty function: $\phi_k(\mathbf{x}_k)$

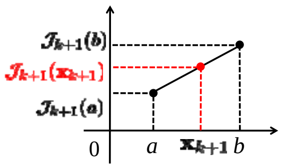


[Appendix] Interpolation

Use interpolation to calculate the cost



Ex.) Linear interpolation



[Appendix] Simulation Settings

Number of the state grid

x : 33 $\dot{x}$: 15 $P_{(a,b)}$: 21

Settings of discretization

Final time : $t_f = 0.4$ Step time : $t_s = 0.005$

Number of steps : $N = 81$

States constraints

$$\mathbf{x}_{min} = [\ 0.0 \quad -2.0 \quad 9.0 \times 10^4 \quad 9.0 \times 10^4\ ]^T$$

$$\mathbf{x}_{max} = [\ 0.32 \quad 2.0 \quad 7.0 \times 10^5 \quad 7.0 \times 10^5\ ]^T$$