CVXGEN & MPT
for Air-Conditioning Control

Akihito Kato
FL 13-10-1
Jun 24, 2013

Outline

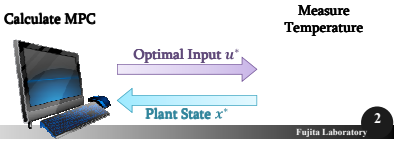
Background

Apply MPC as Air-Conditioning Control

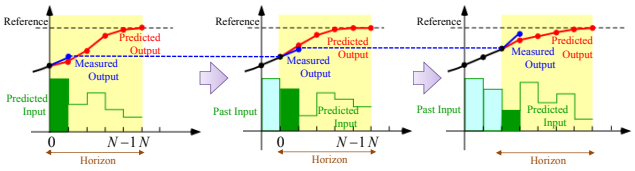
- There are two types of solver for MPC
- Computation time must be less than the sampling time

Objective

Compare the computation time



Model Predictive Control (MPC)



I. Use the combination of the most recent of the states

II. Optimize the cost function for the horizon interval

III. Only use the plant input of the first step

Problem Setting

Minimize

$$J = \sum_{i=1}^{N} ||\bar{y}(i) - y_{ref}(i)||^2 Q + \sum_{j=0}^{N-1} ||\Delta\bar{u}(j)||^2 R$$

Subject to

$$\bar{x}(k+1) = A\bar{x}(k) + B\bar{u}(k) \qquad k = 0 \ldots N$$
$$\bar{y}(k+1) = C\bar{x}(k+1) \qquad k = 0 \ldots N$$
$$\Delta\bar{u}(k+1) = \bar{u}(k) - \bar{u}(k-1) \qquad k = 1 \ldots N-1$$
$$u_{min} \le \bar{u}(k) \le u_{max} \qquad k = 0 \ldots N$$
$$\Delta u_{min} \le \bar{u}(k+1) - \bar{u}(k) \Delta u_{max} \qquad k = 0 \ldots N-1$$

Setup

$N = 10$ $\quad Q = I_9$ $\quad R = 1$

$u_{min} = -3$ $\quad u_{max} = 6$ $\quad \Delta u_{min} = -2$ $\quad \Delta u_{max} = 2$

$y_{ref} = [1, 1, 1, 1, 1, 1, 1, 1, 1]$

$x_0 = [0.5933, 0.77, -0.0733, 0.1150, 0.1017, 0.0617, -0.0133, -0.0133, 0.2633]$

Parameters

- Dynamics Matrix : $A \in \mathbb{R}^{9\times9}$
- Transfer Matrix : $B \in \mathbb{R}^{9}$
- Output Matrix : $C \in \mathbb{R}^{9\times9}$
- State Cost : $Q \in \mathbb{S}_+^9$
- Input Cost : $R \in \mathbb{S}_+^1$
- Estimated State : $\bar{x} \in \mathbb{R}^9$
- Predicted Output : $\bar{y} \in \mathbb{R}^9$
- Predicted Input : $\bar{u} \in \mathbb{R}$
- Reference : $y_{ref} \in \mathbb{R}^9$
- Horizon : N

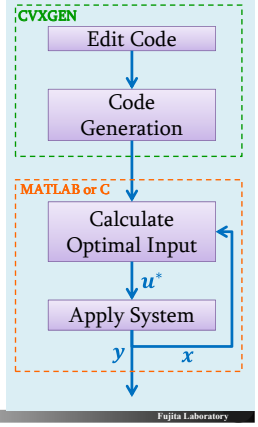
CVXGEN


```
dimensions
  n = 1; %input
  n = 9; %states
  Hp = 10; %prediction horizon
  Hc = 10; %control horizon
end

parameters
  A(n,n) %dynamics matrix
  B(n,u) %transfer matrix
  C(n,n) %output matrix
  Q(n,n) psd %state cost
  Q_final(n,n) psd
  R(u,u) psd %input cost
  x[0](n) %initial state
  y_ref(n) %reference
  u_max; u_min; u_del_min; u_del_max
end

variables
  x[i] (n), i=1..Hp+1 %state
  y[i] (n), i=1..Hp+1 %output
  u[i] (u), i=0..Hc %input
  del_u[i](n), i=0..Hc %change of input
end

minimize
  sum[i=1..Hp](quad(y[i]-y_ref,Q)) + sum[j=0..Hc-1](quad(del_u[j],R))

subject to
  y[i+1] == Cx[i+1], i=1..Hp
  del_u[i+1] == u[j]-u[j-1], j=1..Hc-1
  x[i+1] == Ax[i] + Bu[i], i=0..Hp %dynamics constraints
  u_min <= u[j] <= u_max, j=0..Hc
  u_del_min <= u[i+1]-u[i] <= u_del_max, j=0..Hc-1
end
```

Definition Dimensions

Definition Parameters

Problem Setting



Make M-File for CVXGEN


```
r0 = 1; % Reference
Ts = 31; % Sampling Time

params.A = sysd.A; % Dynamics Matrix
params.B = sysd.B; % Transfer Matrix
params.C = sysd.C; % Output Matrix
params.Q = eye(9,9); % State Cost
params.R = 1; % Input Cost

params.u_del_min = -2; % Constraint
params.u_del_max = 2; % Constraint
params.u_max = 6; % Constraint
params.u_min = -3; % Constraint
params.x_0 = [0.5933;  0.7700;
             -0.0733;  0.1150;
              0.1017; 0.06170;
             -0.0133; -0.0133;
              0.2633        ]; % Initial State
params.y_ref = [r0;r0;r0;r0;r0 ;
                r0;r0;r0;r0]; % Reference

settings.verbose = 0; % Switch of Display
settings.max_iters = 25; % Max Iteration
```

Set of System

Set of Constraint

Make M-File for CVXGEN

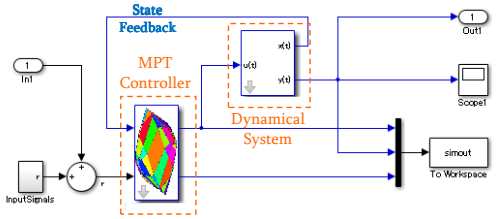
Tokyo Institute of Technology

```
for t = 31:31:31*200
    time = [time,t];

    [vars,status] = csolve(params,settings); % CVXGEN Solver

    T2 = [0,Ts];
    u2 = [vars.u_0, vars.u_0];

    [y,tt,x] = lsim(sysd,u2,T2,params.x_0); % Dynamical System

    params.x_0 = x(2,:)'; % Initialize Initial State

    y_all = [y_all; y(size(y,1),:)]; % Save Output
    u_all = [u_all; u2(1)]; % Save Input

    counter = counter+1;

    clear vars status % Initialize CVXGEN
end
```

Operation of CVXGEN

Multi – Parametric Toolbox (MPT)

Tokyo Institute of Technology



System: sysStruct = mpt_sys(source)

Solver: ctrl = mpt_control(sysStruct, probStruct, flag)

Analysis: analy = mpt_lyapunov(ctrl, type)

MPT Only R = mpt_reachSets(sysStruct, x0, u0, N)

Make M-File for MPT

Tokyo Institute of Technology

```
sysStruct.A = sysd.A; % Dynamics Matrix
sysStruct.B = sysd.B; % Transfer Matrix
sysStruct.C = sysd.C; % Output Matrix

% --Constraint
sysStruct.xmax = Inf*ones(9,1);
sysStruct.xmin = -Inf*ones(9,1);
sysStruct.umax = 6;
sysStruct.umin = -3;
sysStruct.dumax = 2;
sysStruct.dumin = -2;
```

Set of sysStruct

```
probStruct.N = 10; % Horizon
probStruct.Nc = 10; % Input Horizon
probStruct.norm = 2; % Norm
probStruct.Q = eye(9,9); % State Cost
probStruct.R = 1; % Input Cost
probStruct.P_N = P;
probStruct.subopt_lev = 0;
probStruct.Tconstraint = 0;
probStruct.tracking =0;
probStruct.yref = [1;1;1;1;1;1;1;1;1]; % Reference
probStruct.Qy = eye(9,9);
```

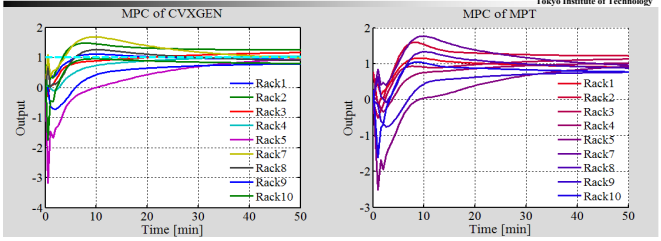
Set of probStruct

```
x0 = [ 0.5933, 0.77, -0.0793, 0.1150,...
       0.1017, 0.0617, -0.0133, -0.0138, 0.2633 ];

[ F, P, E ] = dlqr( sysStruct.A, sysStruct.B, Q, R ) ;
ctrl = mpt_control( sysStruct, probStruct, 'online' );
[X,U,Y] = sim(ctrl,x0,100);
```

Operation of MPT

Compare MPT and CVXGEN

Tokyo Institute of Technology



	CVXGEN	MPT
Calculate Speed	0.4s	2.7s
C Language	○	○
Matlab	○	○
Simulink	×	○
Time-Varying System	×	○
Nonlinear System	×	○