MECH 6v29.002 – Model Predictive Control

L21 – Distributed MPC

Outline

- Noncooperative Distributed MPC
- Cooperative Distributed MPC
- Example

Noncooperative Distributed MPC

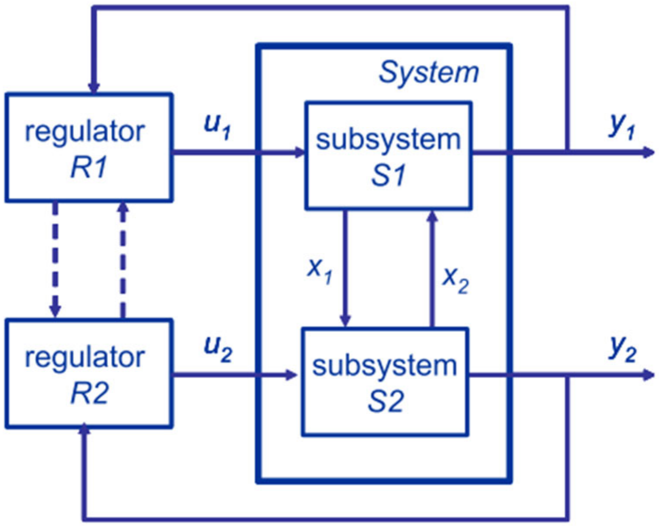
- Optimal control trajectory depends on the input trajectory of other subsystem

$$U_i(0) = K_i x_i(0) + L_{ij} U_j$$

- Communicate trajectories and iterate

$$U_i^{p+1} = w_i U_i^0 + (1 - w_i) U_i^p, \quad 0 < w_i < 1, \quad w_1 + w_2 = 1$$

- Main drawback of noncooperative distributed MPC
  - Nash equilibrium may not be stable
  - Nash equilibrium may be stable, but the closed-loop system is unstable
  - Nash equilibrium may be stable and the closed-loop system is stable



- Which case arises based on the unique combination of system dynamics and controller design
  - One has to perform this analysis for any time a system or controller parameter changes

Cooperative Distributed MPC

- Now both controllers try to minimize the (global) system objective by optimizing their own input trajectory assuming that they know the input trajectory of the other subsystem

$$V\left(x_1(0), x_2(0), U_1, U_2\right) = \rho_1 V_1\left(x_1(0), U_1, U_2\right) + \rho_2 V_2\left(x_2(0), U_1, U_2\right)$$

Same as centralized MPC

- But now each controller needs a model of all subsystem dynamics

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}$$

- Communication requirements
  - Input trajectories over entire prediction horizon
  - Initial condition
  - Cost function matrices

Mild increase in communication

Cooperative Distributed MPC (cont.)

- Controller i optimization problem

$$\min_{U_i} V\left(x_1(0),x_2(0),U_1,U_2\right)=\rho_1 V_1\left(x_1(0),U_1,U_2\right)+\rho_2 V_2\left(x_2(0),U_1,U_2\right)$$

$$s.t.$$

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix}=\begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}+\begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}\begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix},\ k\in\{0,1,\ldots,N-1\}$$

- Only difference from centralized MPC is that each controller optimizes with respect to its own input trajectory and assumes the input trajectory for the other subsystem is a known disturbance
- Using the batch approach, we know that we can reformulate this optimization problem and remove all variables except initial conditions and input trajectories
  - Modeling the extra states does not increase computation cost of solving the associated optimization problem compared to the noncooperative distributed MPC formulation

Cooperative Distributed MPC (cont.)

- With noncooperative distributed MPC we had

$$U_i^0 = K_i x_i(0) + L_{ij} U_j^p$$

- For cooperative distributed MPC, we have

$$U_i^0 = \begin{bmatrix} K_{ii} & K_{ij} \end{bmatrix} \begin{bmatrix} x_i(0) \\ x_j(0) \end{bmatrix} + L_{ij} U_j^p$$

- Using the same update law as noncooperative distributed MPC

$$U_i^{p+1} = w_i U_i^0 + (1 - w_i) U_i^p$$

we now get

$$\begin{bmatrix} U_1 \\ U_2 \end{bmatrix}^{p+1} = \underbrace{\begin{bmatrix} w_1 K_{11} & w_1 K_{12} \\ w_2 K_{21} & w_2 K_{22} \end{bmatrix}}_{K} \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} + \underbrace{\begin{bmatrix} (1 - w_1) I & w_1 L_{12} \\ w_2 L_{21} & (1 - w_2) I \end{bmatrix}}_{L} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}^p$$

No longer block diagonal

Unchanged

Cooperative Distributed MPC (cont.)

- So we still have to see if there is w_1 and w_2 such that the Nash equilibrium is stable
- But now, if the Nash equilibrium is stable, the closed-loop MPC is guaranteed to be stable as well (unlike the noncooperative case)
- Additionally, we do not have to iterate to convergence at each time step
  - The closed-loop system is stable for all values of $0 \le p$

$$U_i^{p+1} = w_i U_i^0 + \left(1 - w_i\right) U_i^p$$

- At time-step $k = 0$ and iteration p, we can use the batch approach to show that

$$V\left(x_1(0), x_2(0), U_1^{p+1}, U_2^{p+1}\right) = V\left(x_1(0), x_2(0), U_1^p, U_2^p\right)$$

$$-\frac{1}{2}\left[U^p - U^0\right]^T P\left[U^p - U^0\right]$$

- Since P is a positive definite matrix, the total system cost decreases at every iteration

$$U^p = \begin{bmatrix} U_1^p \\ U_2^p \end{bmatrix}, \quad U^0 = \begin{bmatrix} U_1^0 \\ U_2^0 \end{bmatrix}$$

Cooperative Distributed MPC (cont.)

- We can prove closed-loop stability using ideas/procedures we have used in the past
- Based on the solution at time step 0, assume the candidate input trajectory as an initial guess (warm start) at time step 1

$$U_i^p(0) = \begin{bmatrix} u_i(0) \\ u_i(1) \\ \vdots \\ u_i(N-1) \end{bmatrix} \qquad \begin{bmatrix} u_i(1) \\ \vdots \\ u_i(N-1) \\ 0 \end{bmatrix} = U_i^{[p=0]}(1)$$

- Based on the fact that the open-loop system is stable and our use of a Lyapunov equation to formulate our terminal cost functions, we can analyze the difference between the total cost at time 0 and the total cost at time 1 to show

$$V\left(x_1(1), x_2(1), U_1^{[p=0]}(1), U_2^{[p=0]}(1)\right) < V\left(x_1(0), x_2(0), U_1^p(0), U_2^p(0)\right)$$

- Then, since every iteration further decreases the total system cost, we can use this to prove closed-loop stability by using the total system cost as a Lyapunov function

Example – Unstable Nash [1]

- Consider the 2-input, 2-output system given by

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \underbrace{\begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix}}_{G(s)} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix}$$

$$G(s) = \begin{bmatrix} \dfrac{1}{s^2 + 0.4s + 1} & \dfrac{0.5}{0.225s + 1} \\ \dfrac{-0.5}{(0.5s+1)(0.25s+1)} & \dfrac{2}{s^2 + 1.6s + 4/3} \end{bmatrix}$$

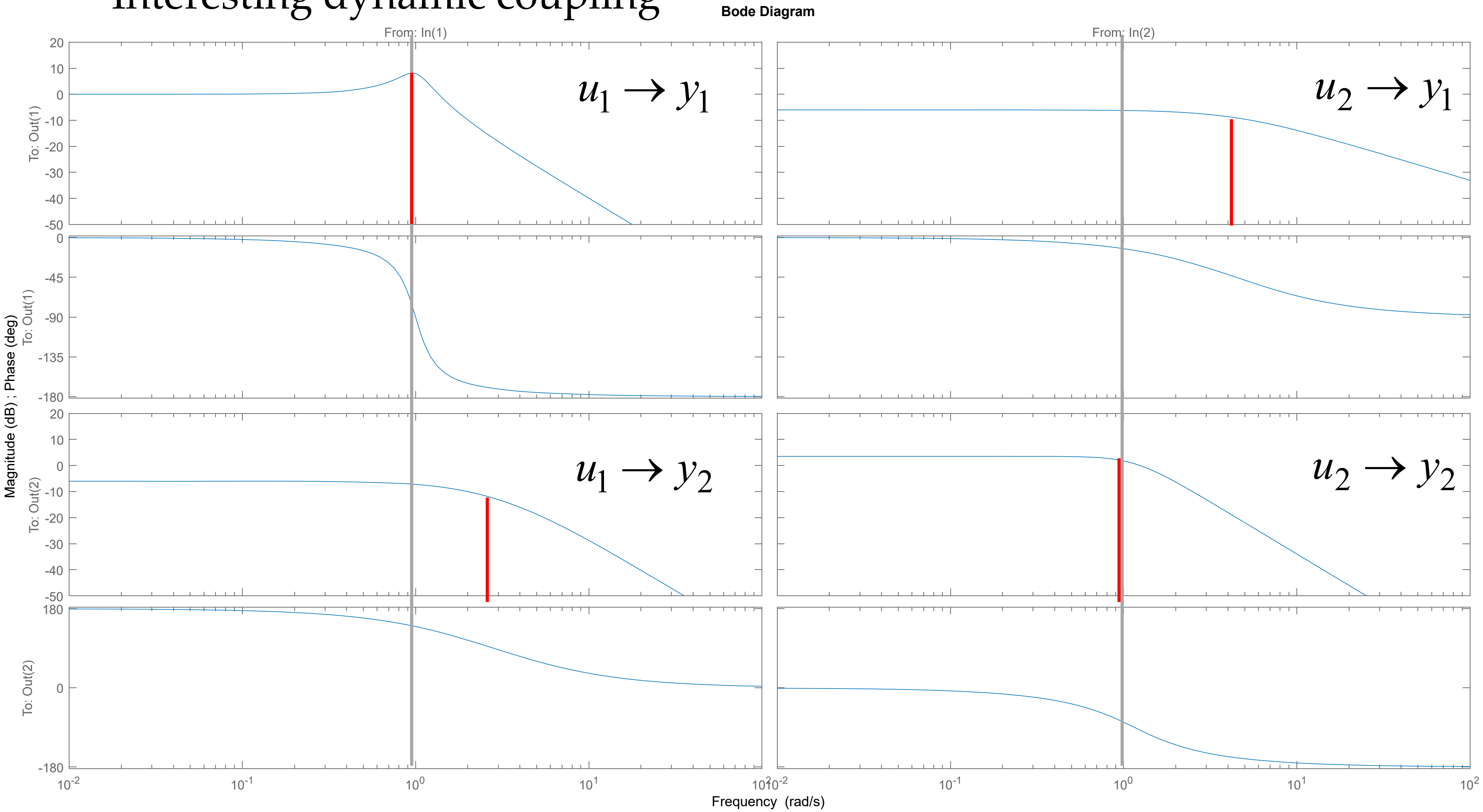
- Mild steady-state coupling

$$G(0) = \begin{bmatrix} 1 & 0.5 \\ -0.5 & 1.5 \end{bmatrix}$$

[1] J. Rawlings, D. Mayne, M. Diehl. "Model Predictive Control: Theory, Computation, and Design," Nob Hill Publishing, 2nd Edition, 2019.

Example – Unstable Nash [1]

- Interesting dynamic coupling



- Coupling (off-diagonal) is faster than control pairings (diagonal)

[1] J. Rawlings, D. Mayne, M. Diehl. "Model Predictive Control: Theory, Computation, and Design," Nob Hill Publishing, 2nd Edition, 2019.

Example – Unstable Nash [1]

- Define transfer function dynamics

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \underbrace{\begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix}}_{G(s)} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix}$$

```
%% Example 1 - Unstable Nash equilibrium
G11 = tf([1],[1 0.4 1]);
G12 = tf([0.5],[0.225 1]);
G21 = tf([-0.5],[0.125 0.75 1]);
G22 = tf([2],[1 1.6 4/3]);

G = [G11 G12; G21 G22];

figure;
bode(G)
```

- Convert to discrete-time state-space with $\Delta t = 0.2s$

$$x_1^+ = A_1 x_1 + B_{11} u_1 + B_{12} u_2$$
$$y_1 = C_1 x_1$$

$$x_2^+ = A_2 x_2 + B_{22} u_2 + B_{21} u_1$$
$$y_2 = C_2 x_2$$

```
%% State-space realization
sys11c = ss(G11);
sys12c = ss(G12);
sys21c = ss(G21);
sys22c = ss(G22);

dt = 0.2;

sys11 = c2d(sys11c,dt);
sys12 = c2d(sys12c,dt);
sys21 = c2d(sys21c,dt);
sys22 = c2d(sys22c,dt);

A1 = blkdiag(sys11.A,sys12.A);
B11 = [sys11.B;zeros(size(sys12.A,1),size(sys11.B,2))];
B12 = [zeros(size(sys11.A,1),size(sys12.B,2));sys12.B;];
C1 = [sys11.C sys12.C];
A2 = blkdiag(sys22.A,sys21.A);
B22 = [sys22.B;zeros(size(sys21.A,1),size(sys22.B,2))];
B21 = [zeros(size(sys22.A,1),size(sys21.B,2));sys21.B;];
C2 = [sys22.C sys21.C];
```

[1] J. Rawlings, D. Mayne, M. Diehl. "Model Predictive Control: Theory, Computation, and Design," Nob Hill Publishing, 2nd Edition, 2019.

Example – Unstable Nash [1]

- Define cost functions

$$V_1\left(x_1(0),U_1,U_2\right)=\sum_{k=0}^{N-1}\ell_1\left(x_1(k),u_1(k)\right)+V_{1f}\left(x_1(N)\right)$$

$$V_2\left(x_2(0),U_1,U_2\right)=\sum_{k=0}^{N-1}\ell_2\left(x_2(k),u_2(k)\right)+V_{2f}\left(x_2(N)\right)$$

```
%% Cost function design
Q_bar = 1;
R = 0.01;

Q1 = C1'*Q_bar*C1;
Q2 = C2'*Q_bar*C2;
R1 = R;
R2 = R;

P1f = dlyap(A1',Q1);
P2f = dlyap(A2',Q2);

N = 30;
```

$$\ell_i\left(x_i(k),u_i(k)\right)=\frac{1}{2}x_i^T(k)Q_i x_i(k)+\frac{1}{2}u_i^T(k)R_i u_i(k) \qquad Q_i=C_i^T\bar{Q}_i C_i$$

$$V_{i,f}\left(x_i(N)\right)=\frac{1}{2}x_i^T(N)P_{i,f}x_i(N) \qquad A_i^T P_{i,f}A_i-P_{i,f}=-Q_{i,f}$$

[1] J. Rawlings, D. Mayne, M. Diehl. "Model Predictive Control: Theory, Computation, and Design," Nob Hill Publishing, 2nd Edition, 2019.

Example – Unstable Nash [1]

- Centralized MPC formulation

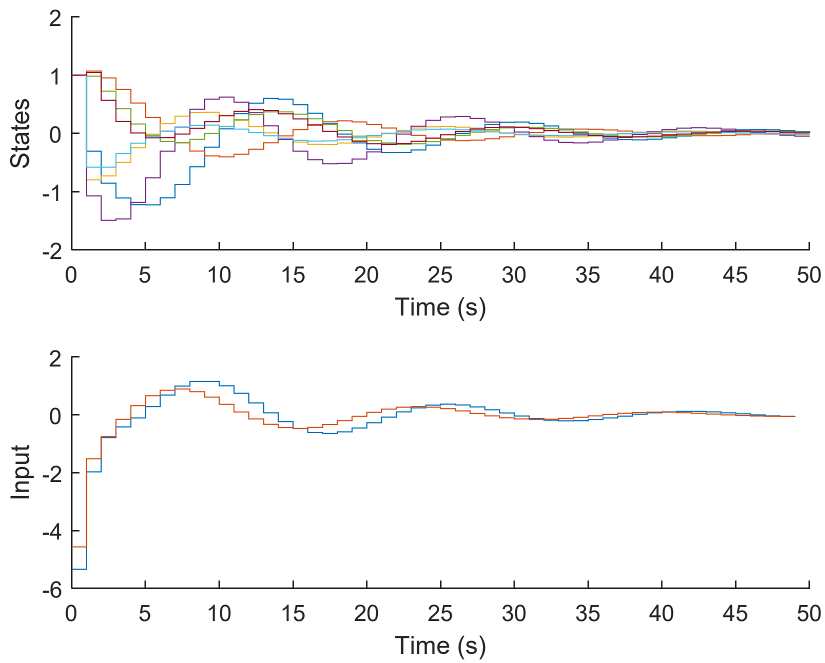
```
%% Centralized MPC
n1 = 3; m1 = 1;
n2 = 4; m2 = 1;

u1_ = sdpvar(repmat(m1,1,N),repmat(1,1,N));
x1_ = sdpvar(repmat(n1,1,N+1),repmat(1,1,N+1));
u2_ = sdpvar(repmat(m2,1,N),repmat(1,1,N));
x2_ = sdpvar(repmat(n2,1,N+1),repmat(1,1,N+1));

constraints = [];
objective = 0;
for k = 1:N
    objective = objective + 0.5*x1_{k}'*Q1*x1_{k} + 0.5*u1_{k}'*R1*u1_{k};
    objective = objective + 0.5*x2_{k}'*Q2*x2_{k} + 0.5*u2_{k}'*R2*u2_{k};
    constraints = [constraints, x1_{k+1} == A1*x1_{k} + B11*u1_{k} + B12*u2_{k}];
    constraints = [constraints, x2_{k+1} == A2*x2_{k} + B21*u1_{k} + B22*u2_{k}];
end
objective = objective + 0.5*x1_{N+1}'*P1f*x1_{N+1};
objective = objective + 0.5*x2_{N+1}'*P2f*x2_{N+1};

controller = optimizer(constraints,objective,sdpsettings('solver','gurobi'),{x1_{1}, x2_{1}},[u1_{1}, u2_{1}]);
```

[1] J. Rawlings, D. Mayne, M. Diehl. "Model Predictive Control: Theory, Computation, and Design," Nob Hill Publishing, 2nd Edition, 2019.

Example – Unstable Nash [1]

- Centralized MPC simulation

```
%%
x0 = ones(n1+n2,1);
k_sim = 50;
x_sim = [x0];
u_sim = [];
J_sim = 0;
for i = 1:k_sim
    [u,diagnostics] = controller{{x_sim(1:n1,i),x_sim(n1+1:end,i)}};
    u_sim = [u_sim u'];
    x_sim = [x_sim blkdiag(A1,A2)*x_sim(:,end) + [B11 B12;B21 B22]*u_sim(:,end)];
    J_sim = J_sim + 0.5*x_sim(:,end-1)'*blkdiag(Q1,Q2)*x_sim(:,end-1) + 0.5*u_sim(:,end)'*blkdiag(R1,R2)*u_sim(:,end);
end
```



$$J_{sim}^{cent} = 2.97$$

[1] J. Rawlings, D. Mayne, M. Diehl. "Model Predictive Control: Theory, Computation, and Design," Nob Hill Publishing, 2nd Edition, 2019.

Example – Unstable Nash [1]

- Decentralized MPC formulation

```
%% Decentralized MPC
% Controller 1
u1_ = sdpvar(repmat(m1,1,N),repmat(1,1,N));
x1_ = sdpvar(repmat(n1,1,N+1),repmat(1,1,N+1));

constraints = [];
objective = 0;
for k = 1:N
    objective = objective + 0.5*x1_{k}'*Q1*x1_{k} + 0.5*u1_{k}'*R1*u1_{k};
    constraints = [constraints, x1_{k+1} == A1*x1_{k} + B11*u1_{k}];
end
objective = objective + 0.5*x1_{N+1}'*P1f*x1_{N+1};

controller1 = optimizer(constraints,objective,sdpsettings('solver','gurobi'),x1_{1},u1_{1});

% Controller 2
u2_ = sdpvar(repmat(m2,1,N),repmat(1,1,N));
x2_ = sdpvar(repmat(n2,1,N+1),repmat(1,1,N+1));

constraints = [];
objective = 0;
for k = 1:N
    objective = objective + 0.5*x2_{k}'*Q2*x2_{k} + 0.5*u2_{k}'*R2*u2_{k};
    constraints = [constraints, x2_{k+1} == A2*x2_{k} + B22*u2_{k}];
end
objective = objective + 0.5*x2_{N+1}'*P2f*x2_{N+1};

controller2 = optimizer(constraints,objective,sdpsettings('solver','gurobi'),x2_{1},u2_{1});
```

Local cost

Neglect effects of other inputs

[1] J. Rawlings, D. Mayne, M. Diehl. "Model Predictive Control: Theory, Computation, and Design," Nob Hill Publishing, 2nd Edition, 2019.

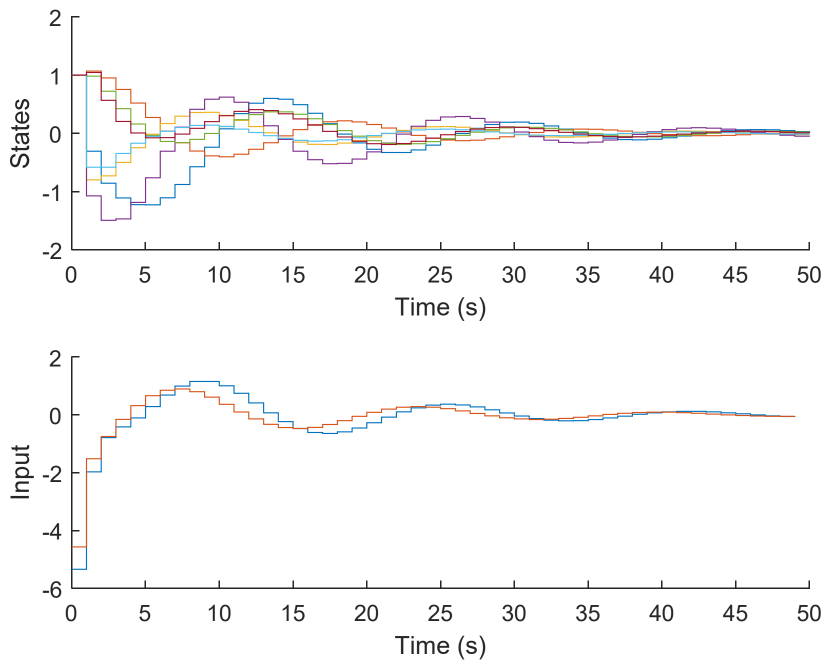
Example – Unstable Nash [1]

- Decentralized MPC simulation
    - Main part of simulation (call controllers)

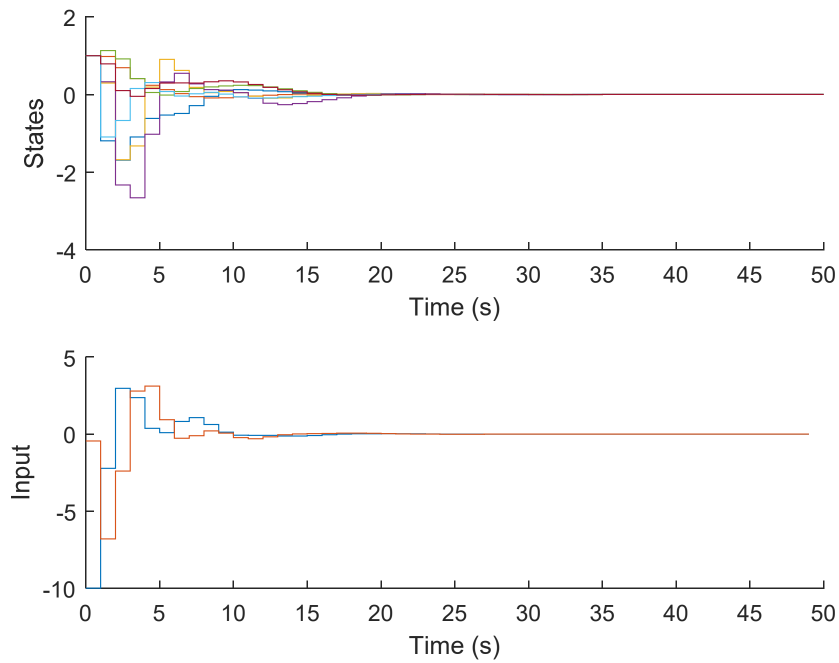
```
[u1] = controller1{x_sim(1:n1,i)};
[u2] = controller2{x_sim(1+n1:end,i)};
u_sim = [u_sim [u1;u2]];
```

Only requires local information

Centralized $J_{sim}^{cent} = 2.97$



Decentralized $J_{sim}^{decent} = 6.92$



[1] J. Rawlings, D. Mayne, M. Diehl. "Model Predictive Control: Theory, Computation, and Design," Nob Hill Publishing, 2nd Edition, 2019.

Example – Unstable Nash [1]

- Noncooperative Distributed MPC formulation

```
%% Noncooperative Distributed MPC
% Controller 1
u1_ = sdpvar(repmat(m1,1,N),repmat(1,1,N));
x1_ = sdpvar(repmat(n1,1,N+1),repmat(1,1,N+1));
u2_ = sdpvar(repmat(m2,1,N),repmat(1,1,N));

constraints = [];
objective = 0;
for k = 1:N
    objective = objective + 0.5*x1_{k}'*Q1*x1_{k} + 0.5*u1_{k}'*R1*u1_{k};
    constraints = [constraints, x1_{k+1} == A1*x1_{k} + B11*u1_{k} + B12*u2_{k}];
end
objective = objective + 0.5*x1_{N+1}'*P1f*x1_{N+1};

controller1 = optimizer(constraints,objective,sdpsettings('solver','gurobi'),{x1_{1},u2_{:}},u1_);

% Controller 2
u2_ = sdpvar(repmat(m2,1,N),repmat(1,1,N));
x2_ = sdpvar(repmat(n2,1,N+1),repmat(1,1,N+1));
u1_ = sdpvar(repmat(m1,1,N),repmat(1,1,N));

constraints = [];
objective = 0;
for k = 1:N
    objective = objective + 0.5*x2_{k}'*Q2*x2_{k} + 0.5*u2_{k}'*R2*u2_{k};
    constraints = [constraints, x2_{k+1} == A2*x2_{k} + B22*u2_{k} + B21*u1_{k}];
end
objective = objective + 0.5*x2_{N+1}'*P2f*x2_{N+1};

controller2 = optimizer(constraints,objective,sdpsettings('solver','gurobi'),{x2_{1},u1_{:}},u2_);
```

Local cost

Includes other inputs as known disturbances

[1] J. Rawlings, D. Mayne, M. Diehl. "Model Predictive Control: Theory, Computation, and Design," Nob Hill Publishing, 2nd Edition, 2019.

Example – Unstable Nash [1]

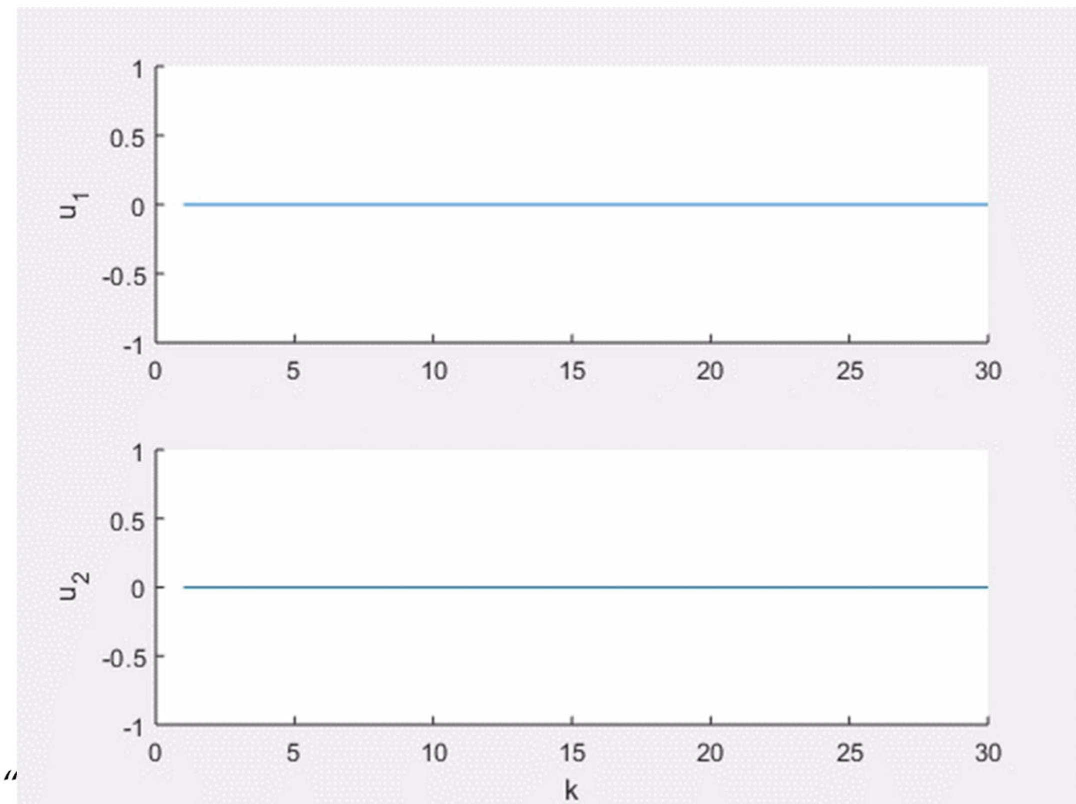
- Noncooperative Distributed MPC simulation
  - Communication iterations at time step 0

```
u1p = mat2cell(zeros(m1,N),1,ones(1,N));
u2p = mat2cell(zeros(m1,N),1,ones(1,N));
i = 1;
for p = 1:20
    [u1] = controller1{{x_sim(1:n1,i),u2p{:}}};
    [u2] = controller2{{x_sim(1+n1:end,i),u1p{:}}};
    u1p = u1;
    u2p = u2;
end
```

Initial guess at input traj.

Iteratively communicate planned input traj.

Goes unstable



Unstable Nash equilibrium

[1] J. Rawlings, D. Mayne, M. Diehl. “ … gn,” Nob Hill Publishing, 2nd Edition, 2019.

Example – Unstable Nash [1]

- Cooperative Distributed MPC formulation

```
%% Cooperative Distributed MPC
% Controller 1
u1_ = sdpvar(repmat(m1,1,N),repmat(1,1,N));
x1_ = sdpvar(repmat(n1,1,N+1),repmat(1,1,N+1));
u2_ = sdpvar(repmat(m2,1,N),repmat(1,1,N));
x2_ = sdpvar(repmat(n2,1,N+1),repmat(1,1,N+1));

constraints = [];
objective = 0;
for k = 1:N
    objective = objective + 0.5*x1_{k}'*Q1*x1_{k} + 0.5*u1_{k}'*R1*u1_{k};
    objective = objective + 0.5*x2_{k}'*Q2*x2_{k} + 0.5*u2_{k}'*R2*u2_{k};
    constraints = [constraints, x1_{k+1} == A1*x1_{k} + B11*u1_{k} + B12*u2_{k}];
    constraints = [constraints, x2_{k+1} == A2*x2_{k} + B21*u1_{k} + B22*u2_{k}];
end
objective = objective + 0.5*x1_{N+1}'*P1f*x1_{N+1};
objective = objective + 0.5*x2_{N+1}'*P2f*x2_{N+1};

controller1 = optimizer(constraints,objective,sdpsettings('solver','gurobi'),{x1_{1},x2_{1},u2_{:}},u1_);
```

Full system cost

Includes other inputs as known disturbances

- Controller 2 is similar

[1] J. Rawlings, D. Mayne, M. Diehl. "Model Predictive Control: Theory, Computation, and Design," Nob Hill Publishing, 2nd Edition, 2019.

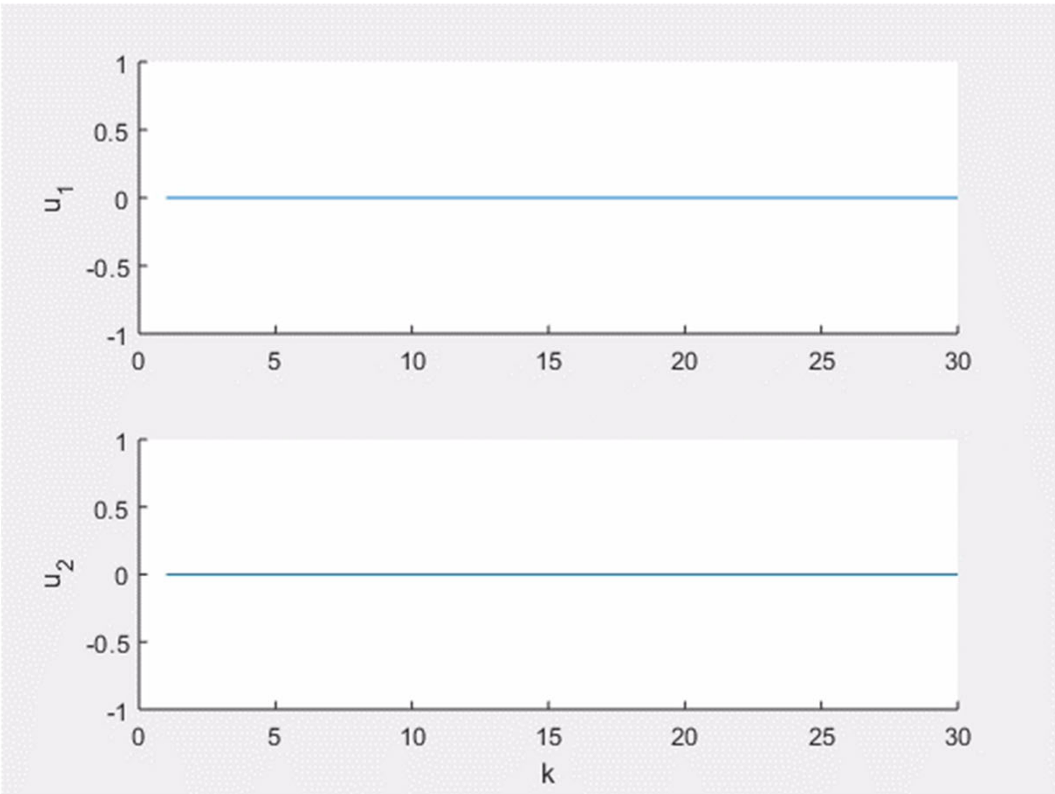
Example – Unstable Nash [1]

- Cooperative Distributed MPC simulation
  - Communication iterations at time step 0

```
u1p = mat2cell(zeros(m1,N),1,ones(1,N));
u2p = mat2cell(zeros(m1,N),1,ones(1,N));
i = 1;
for p = 1:20
    [u1] = controller1{{x_sim(1:n1,i),x_sim(1+n1:end,i),u2p{:}}};
    [u2] = controller2{{x_sim(1+n1:end,i),x_sim(1:n1,i),u1p{:}}};
    u1p = u1;
    u2p = u2;
end
```

Converges

Stable Nash equilibrium



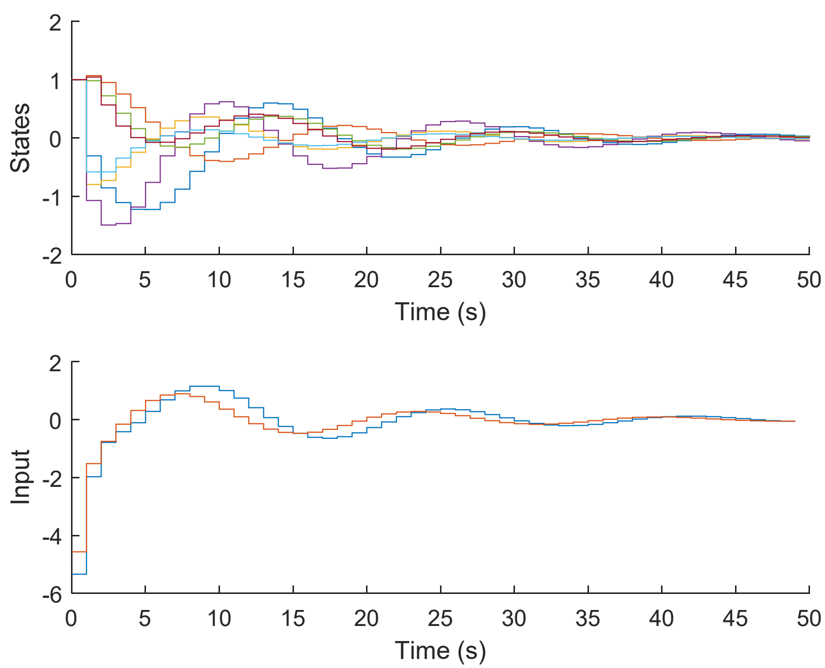
[1] J. Rawlings, D. Mayne, M. Diehl. "Model Predictive Control: Theory, Computation, and Design," Nob Hill Publishing, 2nd Edition, 2019.

Example – Unstable Nash [1]

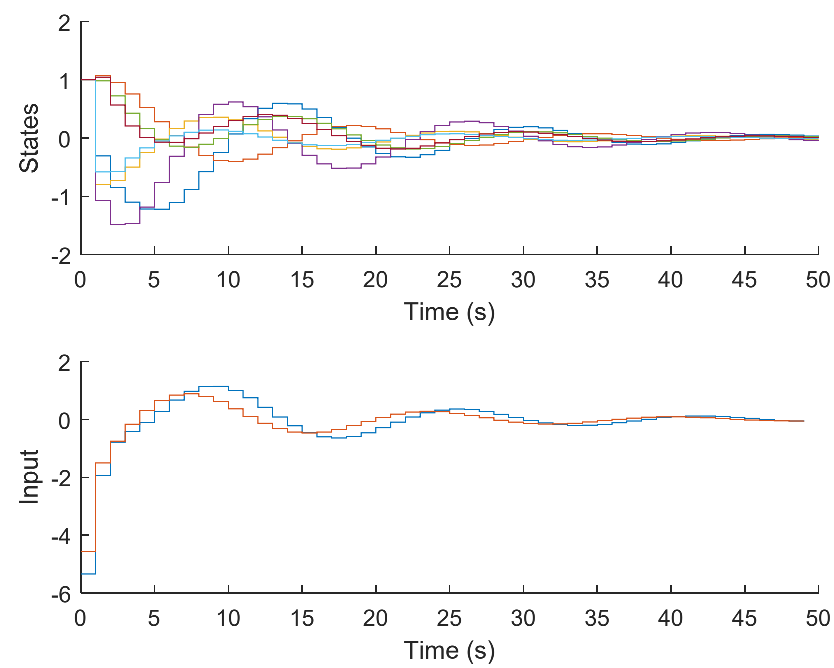
- Cooperative Distributed MPC simulation
  - Main part of simulation (call controllers)

```
for p = 1:20
    [u1] = controller1{{x_sim(1:n1,i),x_sim(1+n1:end,i),u2p{:}}};
    [u2] = controller2{{x_sim(1+n1:end,i),x_sim(1:n1,i),u1p{:}}};
    u1p = u1;
    u2p = u2;
end
u_sim = [u_sim cell2mat([u1(1);u2(1)])];
```

Centralized $J_{sim}^{cent} = 2.97$



Cooperative $J_{sim}^{coop} = 2.97$



[1] J. Rawlings, D. Mayne, M. Diehl. "Model Predictive Control: Theory, Computation, and Design," Nob Hill Publishing, 2nd Edition, 2019.