Modelling Options for Networks with Delay: DDEs, DDFs, PDEs, and PIEs

Matthew Peet, S. Shivakumar, S. Wu, S. Weiland, A. Das, K. Gu et al.
Arizona State University
Tempe, AZ USA

Coupled Communications and Autonomy Challenges in Connected Autonomous Vehicles
NSF CPS PI Meeting 2019

November 21, 2019

Control of a Network of Vehicles with Delay

Consider the dynamics of a swarm of UAVs:

$$\dot{x}_i(t) = a_i x_i(t) + \sum_{j=1}^{N} a_{ij} x_j(t - \hat{\tau}_{ij}) + b_{1i} w(t - \bar{\tau}_i) + b_{2i} u(t - h_i)$$

$$z(t) = C_1 x(t) + D_{12} u(t) \qquad \text{Regulated Output}$$

$$y_i(t) = c_{2i} x_i(t - \tilde{\tau}_i) + d_{21i} w(t - \tilde{\tau}_i) \qquad \text{Sensed output}$$

Dynamics:

- a_i is the internal dynamics of UAV i
- a_{ij} is the effect of UAV j on UAV i.
- b_{1i} is the effect of noise on UAV i
- b_{2i} is the effect of the controller on UAV i
- c_{2i} is the measured output of from UAV i
- d_{21i} is the effect of noise on the sensor on UAV i
- C_1 is the output of states to minimize in the optimal control problem
- D_{12} is the actuator output to minimize in the optimal control problem

Delays:

- $\hat{\tau}_{ij}$ is the state delay from UAV j to UAV i
- h_i is the input delay from controller to reach UAV i
- $\bar{\tau}_i$ is the process delay (wind, tracking signal, et c.) for UAV i
- $\tilde{\tau}_i$ is the measurement delay from UAV i to controller

Optimal Control Form using Delay-Diff. Equations (DDEs)

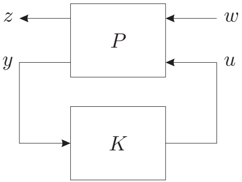
General Form of Optimal Control Problem using DDEs:

$$\dot{x}(t) = A_0 x(t) + B_1 w(t) + B_2 u(t) + \sum_{i=1}^{N} \left(A_i x(t-\tau_i) + B_{1i} w(t-\tau_i) + B_{2i} u(t-\tau_i) \right)$$

$$z(t) = C_{10} x(t) + D_{11} w(t) + D_{12} u(t) + \sum_{i=1}^{N} \left(C_{1i} x(t-\tau_i) + D_{11i} w(t-\tau_i) + D_{12i} u(t-\tau_i) \right)$$

$$y(t) = C_{20} x(t) + D_{21} w(t) + D_{22} u(t) + \sum_{i=1}^{N} \left(C_{2i} x(t-\tau_i) + D_{21i} w(t-\tau_i) + D_{22i} u(t-\tau_i) \right)$$

Equations: State Equation; Regulated Output; Sensed Output



Problem:

- No network Structure!

Problems:

- Utterly Intractable
- Can't represent some networks

For Simplicity, I leave off the distributed-delay terms.
EVERYTHING is infinite-dimensional - x and w and u

Optimal Control Form using Delay-Diff. Equations (DDEs)

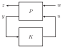
General Form of Optimal Control Problem using DDEs:

$$\dot{x}(t) = A_0 x(t) + B_1 w(t) + B_2 u(t) + \sum_{i=1}^{N} \Big(A_i x(t-\tau_i) + B_{1i} w(t-\tau_i) + B_{2i} u(t-\tau_i) \Big)$$

$$z(t) = C_{10} x(t) + D_{11} w(t) + D_{12} u(t) + \sum_{i=1}^{N} \Big(C_{1i} x(t-\tau_i) + D_{11i} w(t-\tau_i) + D_{12i} u(t-\tau_i) \Big)$$

$$y(t) = C_{20} x(t) + D_{21} w(t) + D_{22} u(t) + \sum_{i=1}^{N} \Big(C_{2i} x(t-\tau_i) + D_{21i} w(t-\tau_i) + D_{22i} u(t-\tau_i) \Big)$$

Equations: State Equation; Regulated Output; Sensed Output



Problem:
- No network Structure!

Problems:
- Utterly Intractable
- Can't represent some networks

For Simplicity, I leave off the distributed-delay terms.
EVERYTHING is infinite-dimensional - x and w and u

9(N+1) matrices needed to define system

A Network as a DDE

Network Model (neglecting state delays):

$$\dot{x}_i(t) = a_i x_i(t) + \sum_{j=1}^{N} a_{ij} x_j(t) + b_{1i} w(t-\tau_i) + b_{2i} u(t-\tau_{N+i})$$
$$z(t) = C_1 x(t) + D_{12} u(t)$$
$$y_i(t) = c_{2i} x_i(t-\tau_{2N+i}) + d_{21i} w(t-\tau_{2N+i}).$$

DDE Representation:

$$\dot{x}(t) = A_0 x(t) + \sum_{i=1}^{N} B_{1i} w(t-\tau_i) + \sum_{j=N+1}^{2N} B_{2i} u(t-\tau_i)$$
$$z(t) = C_{10} x(t) + D_{12} u(t)$$
$$y(t) = \sum_{i=2N+1}^{3N} C_{2i} x(t-\tau_i) + \sum_{i=2N+1}^{3N} D_{21i} w(t-\tau_i)$$

Problem: delayed information has dimension $3N(n_x \cdot N + n_w + n_u)$ where here N is # of UAVs.

Some Networks CANNOT be modelled using DDEs

Network model with input delay:

$$\dot{x}(t) = A_0 x(t) + B_1 w(t) + \sum_{i=1}^{N} B_{2i} u(t - \tau_i)$$
$$z(t) = C_1 x(t) + D_{12} u(t)$$
$$y(t) = C_2 x(t) + D_{21} w(t) + \sum_{i=1}^{N} D_{22i} u(t - \tau_i).$$

with STATIC FEEDBACK:

$$u(t) = F y(t)$$

Now, substituting $u(t) = F y(t)$ into the sensed output term, we obtain solutions of the form

$$\dot{x}(t) = A_0 x(t) + B_1 w(t) + \sum_{i} B_{2i} F y(t - \tau_i)$$
$$z(t) = C_1 x(t) + D_{12} F y(t)$$
$$y(t) = C_2 x(t) + D_{21} w(t) + \sum_{i=1}^{N} D_{22i} F y(t - \tau_i). \tag{1}$$

There is no DDE which satisfies Eqns. (1) due to the recursion in the output.

Advantages/Disadvantages of DDE formulation

Advantages:

- Well studied
  - State delay well-studied using LK functions
  - Input delay handled by Smith predictors
  - Padé approximations, LMI methods, SOS methods, etc.
- Always Well-Posed

Disadvantages?

- Lots of delay terms everywhere
- Implies lots of information is delayed
- Can't represent some models
- Many tools implicitly treat as a PDE

Optimal Control via Diff.-DiFFerence Equations (DDFs)

DDFs separate delayed information into low-dimensional channels

$$\begin{aligned}
\dot{x}(t) &= A_0 x(t) + B_1 w(t) + B_2 u(t) + B_v v(t)\\
z(t) &= C_1 x(t) + D_{11} w(t) + D_{12} u(t) + D_{1v} v(t)\\
y(t) &= C_2 x(t) + D_{21} w(t) + D_{22} u(t) + D_{2v} v(t)\\
r_i(t) &= C_{ri} x(t) + B_{r1i} w(t) + B_{r2i} u(t) + D_{rvi} v(t)
\end{aligned}$$

$$v(t) = \sum_{i=1}^{N} C_{vi} r_i(t-\tau_i) + \sum_{i=1}^{N} \int_{-\tau_i}^{0} C_{vdi}(s) r_i(t+s) ds$$

The information which is being delayed is stored in the information channels $r_i(t)$

- State in green is the infinite-dimensional part of the state
- Allows for lower-dimensional states
- Allows for simple difference equations (Discrete time) using D_{rvi}

Converting a DDE to a DDF

DDE Formulation:

$$\dot{x}(t) = A_0 x(t) + B_1 w(t) + B_2 u(t) + \sum_{i=1}^{N} \Big(A_i x(t-\tau_i) + B_{1i} w(t-\tau_i) + B_{2i} u(t-\tau_i) \Big)$$

$$z(t) = C_{10} x(t) + D_{11} w(t) + D_{12} u(t) + \sum_{i=1}^{N} \Big(C_{1i} x(t-\tau_i) + D_{11i} w(t-\tau_i) + D_{12i} u(t-\tau_i) \Big)$$

$$y(t) = C_{20} x(t) + D_{21} w(t) + D_{22} u(t) + \sum_{i=1}^{N} \Big(C_{2i} x(t-\tau_i) + D_{21i} w(t-\tau_i) + D_{22i} u(t-\tau_i) \Big)$$

DDF Formulation:

$$\dot{x}(t) = A_0 x(t) + B_1 w(t) + B_2 u(t) + B_v v(t)$$

$$z(t) = C_1 x(t) + D_{11} w(t) + D_{12} u(t) + D_{1v} v(t)$$

$$y(t) = C_2 x(t) + D_{21} w(t) + D_{22} u(t) + D_{2v} v(t)$$

$$r_i(t) = C_{ri} x(t) + B_{r1i} w(t) + B_{r2i} u(t) + D_{rvi} v(t)$$

$$v(t) = \sum_{i=1}^{N} C_{vi} r_i(t-\tau_i) + \sum_{i=1}^{N} \int_{-\tau_i}^{0} C_{vdi}(s) r_i(t+s) ds$$

Converting a DDE to a DDF

DDF Formulation:

$$\begin{aligned}
\dot{x}(t) &= A_0 x(t) + B_1 w(t) + B_2 u(t) + B_v v(t)\\
z(t) &= C_1 x(t) + D_{11} w(t) + D_{12} u(t) + D_{1v} v(t)\\
y(t) &= C_2 x(t) + D_{21} w(t) + D_{22} u(t) + D_{2v} v(t)\\
r_i(t) &= C_{ri} x(t) + B_{r1i} w(t) + B_{r2i} u(t) + D_{rvi} v(t)
\end{aligned}$$

$$v(t) = \sum_{i=1}^{K} C_{vi} r_i(t-\tau_i) + \sum_{i=1}^{K} \int_{-\tau_i}^{0} C_{vdi}(s) r_i(t+s) ds$$

In order of appearance (all other matrices unchanged):

$$B_v = \begin{bmatrix} I & 0 & 0 \end{bmatrix}, \qquad D_{1v} = \begin{bmatrix} 0 & I & 0 \end{bmatrix} \qquad D_{2v} = \begin{bmatrix} 0 & 0 & I \end{bmatrix}$$

$$C_{ri} = \begin{bmatrix} I \\ 0 \\ 0 \end{bmatrix}, \qquad B_{r1i} = \begin{bmatrix} 0 \\ I \\ 0 \end{bmatrix}, \qquad B_{r2i} = \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix}, \qquad D_{rvi} = 0$$

$$C_{vi} = \begin{bmatrix} A_i & B_{1i} & B_{2i} \\ C_{1i} & D_{11i} & D_{12i} \\ C_{2i} & D_{21i} & D_{22i} \end{bmatrix}, \qquad C_{vdi}(s) = \begin{bmatrix} A_{di}(s) & B_{1di}(s) & B_{2di}(s) \\ C_{1di}(s) & D_{11di}(s) & D_{12di}(s) \\ C_{2di}(s) & D_{21di}(s) & D_{22di}(s) \end{bmatrix}$$

Reverse Transformation (DDF to DDE) Not Possible

Standard Network Model using DDF Formulation

Network Model (neglecting state delays):

$$\dot{x}_i(t) = a_i x_i(t) + \sum_{j=1}^{N} a_{ij} x_j(t) + b_{1i} w(t-\tau_i) + b_{2i} u(t-\tau_{N+i})$$
$$z(t) = C_1 x(t) + D_{12} u(t)$$
$$y_i(t) = c_{2i} x_i(t-\tau_{2N+i}) + d_{21i} w(t-\tau_{2N+i}).$$

The DDF representation:

$$\dot{x}(t) = A_0 x(t) + \sum\nolimits_{i=1}^{2N} v_i(t)$$
$$z(t) = C_{10} x(t) + D_{12} u(t)$$
$$y(t) = \begin{bmatrix} 0 & 0 & I \end{bmatrix} v(t)$$

$$v(t) = \begin{bmatrix} r_1(t-\tau_1) \\ \vdots \\ r_{3N}(t-\tau_{3N}) \end{bmatrix} r_i(t) = \begin{cases} b_{1i} w(t) & i \in [1, N] \\ b_{2,i-N} u(t) & i \in [N+1, 2N] \\ c_{2,i-2N} x_{i-2N}(t) + d_{21,i-2N} w(t) & i \in [2N+1, 3N]. \end{cases}$$

The state-space dimension of the delayed component is $(2n_x + n_y)N$ (vs. $3N(n_x N + n_w + n_u)$ for the DDE). $y_i \in \mathbb{R}^{n_y}$, $x_i \in \mathbb{R}^{n_x}$.

Standard Network Model using DDF Formulation

Network Model (neglecting state delays):

$$\dot{x}_i(t) = a_i x_i(t) + \sum_{j=1}^{N} a_{ij} x_j(t) + b_{1i} w(t-\tau_i) + b_{2i} u(t-\tau_{N+i})$$

$$z(t) = C_1 x(t) + D_{12} u(t)$$

$$y_i(t) = c_{2i} x_i(t-\tau_{2N+i}) + d_{21i} w(t-\tau_{2N+i}).$$

The DDF representation:

$$\dot{x}(t) = A_0 x(t) + \sum_{i=1}^{2N} v_i(t)$$

$$z(t) = C_{10} x(t) + D_{12} u(t)$$

$$y(t) = \begin{bmatrix} 0 & 0 & I \end{bmatrix} v(t)$$

$$v(t) = \begin{bmatrix} r_1(t-\tau_1) \\ \vdots \\ r_{3N}(t-\tau_{3N}) \end{bmatrix} r_i(t) = \begin{cases} b_{1i} w(t) & i \in [1,N] \\ b_{2,i-N} u(t) & i \in [N+1,2N] \\ c_{2,i-2N} x_{i-2N}(t) + d_{21,i-2N} w(t) & i \in [2N+1,3N]. \end{cases}$$

The state-space dimension of the delayed component is $(2n_x + n_y)N$ (vs. $3N(n_x N + n_w + n_u)$ for the DDE). $y_i \in \mathbb{R}^{n_y}$, $x_i \in \mathbb{R}^{n_x}$.

Definition of r was chosen assuming dimension of x_i is less than that of w or u. Otherwise choose

$$r_i(t) = \begin{cases} w(t) & i \in [1,N] \\ u(t) & i \in [N+1,2N] \\ \begin{bmatrix} x_{i-2N}(t) \\ w(t) \end{bmatrix} & i \in [2N+1,3N]. \end{cases}$$

or

$$r_i(t) = \begin{cases} w(t) & i \in [1,N] \\ u(t) & i \in [N+1,2N] \\ c_{2,i-2N} x_{i-2N}(t) + d_{21,i-2N} w(t) & i \in [2N+1,3N]. \end{cases}$$

A Network which is a DDF but not a DDE

Static State Feedback Model:

$$\dot{x}(t) = A_0 x(t) + B_1 w(t) + \sum_i B_{2i} F y(t-\tau_i)$$

$$z(t) = C_1 x(t) + D_{12} F y(t)$$

$$y(t) = C_2 x(t) + D_{21} w(t) + \sum_{i=1}^{N} D_{22i} F y(t-\tau_i).$$

DDF Representation:

$$\dot{x}(t) = A_0 x(t) + B_1 w(t) + \sum\nolimits_{i=1}^{N} B_{2i} v_i(t)$$

$$z(t) = (C_1 + D_{12} F C_2) x(t) + D_{12} F D_{21} w(t) + D_{12} F D_{2v} \sum\nolimits_{i=1}^{N} D_{22i} v_i(t)$$

$$y(t) = C_2 x(t) + D_{21} w(t) + \sum\nolimits_{i=1}^{N} D_{22i} v_i(t)$$

$$r_i(t) = F C_2 x(t) + F D_{21} w(t) + F D_{22i} v_i(t)$$

$$v_i(t) = r_i(t-\tau_i)$$

Advantages/Disadvantages of DDF formulation

Advantages:

- Use of low dimensional channels
- Reduces computation complexity of all analysis and control algorithms
  - Padé approximations, LMI methods, SOS methods, etc.
- Can represent difference equations

Disadvantages:

- Relatively few analysis and controls techniques available
  - Literature is Sparse
- Well-posedness is not assumed

Optimal Control via ODE-PDE Formulation

Almost identical to DDF formulation

- Information channels are represented by PDEs of form $u_t = u_s$

$$\dot{x}(t) = A_0 x(t) + B_1 w(t) + B_2 u(t) + B_v v(t)$$
$$z(t) = C_1 x(t) + D_{11} w(t) + D_{12} u(t) + D_{1v} v(t)$$
$$y(t) = C_2 x(t) + D_{21} w(t) + D_{22} u(t) + D_{2v} v(t)$$
$$\dot{\phi}_i(t,s) = \frac{1}{\tau_i}\phi_{i,s}(t,s) \qquad \phi_i(t,0) = C_{ri} x(t) + B_{r1i} w(t) + B_{r2i} u(t) + D_{rvi} v(t)$$
$$v(t) = \sum_{i=1}^{N} C_{vi}\phi_i(t,-1) + \sum_{i=1}^{N} \int_{-1}^{0} \tau_i C_{vdi}(\tau_i s)\phi_i(t,s) ds$$

An extension of the DDF formulation

- $\phi_i(t)$ is same as $r_i(t)$ was in the DDF
- PDE state is in green.
- Coupled to ODE through Boundary Conditions.

Advantages/Disadvantages of ODE-PDE formulation

Class of Systems Considered: Includes the DDF class of systems

Advantages:

- Use of low dimensional channels
- Tools developed for PDEs can be applied
  - Discretization schemes
  - Backstepping methods for control
- More physical interpretation?

Disadvantages:

- Use of unbounded operators
  - Dirac operators
  - Differential operators

The Partial-Integral Equation (PIE) Formulation

$$\begin{aligned}\mathcal{T}\dot{\mathbf{x}}(t) + \mathcal{B}_{T_1}\dot{w}(t) + \mathcal{B}_{T_2}\dot{u}(t) &= \mathcal{A}\mathbf{x}(t) + \mathcal{B}_1 w(t) + \mathcal{B}_2 u(t)\\ z(t) &= \mathcal{C}_1\mathbf{x}(t) + \mathcal{D}_{11}w(t) + \mathcal{D}_{12}u(t),\\ y(t) &= \mathcal{C}_2\mathbf{x}(t) + \mathcal{D}_{21}w(t) + \mathcal{D}_{22}u(t),\end{aligned} \tag{2}$$

where $\mathcal{T}, \mathcal{A}, \mathcal{B}_i, \mathcal{C}_i, \mathcal{D}_{ij}$ are Partial Integral (4-PI) operators of the form

$$\left(\mathcal{P}\left\{\begin{smallmatrix} P, & Q_1 \\ Q_2, & \{R_i\}\end{smallmatrix}\right\} \underbrace{\begin{bmatrix} x \\ \mathbf{\Phi}\end{bmatrix}}_{\mathbf{x}}\right)(s) := \begin{bmatrix} Px + \int_{-1}^{0} Q_1(s)\mathbf{\Phi}(s)ds \\ Q_2(s)x + \left(\mathcal{P}_{\{R_i\}}\mathbf{\Phi}\right)(s)\end{bmatrix}$$

where $\mathcal{P}_{\{R_i\}}$ is a 3-PI operator of the form:

$$\left(\mathcal{P}_{\{R_i\}}\mathbf{\Phi}\right)(s) := R_0(s)\mathbf{\Phi}(s) + \int_{-1}^{s} R_1(s,\theta)\mathbf{\Phi}(\theta)d\theta + \int_{s}^{0} R_2(s,\theta)\mathbf{\Phi}(\theta)d\theta.$$

The state is in $\mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$

- Dimensions are the same as the ODE-PDE framework
- 4-PI operators are bounded, algebraic.
- Can be used as matrices in LMIs (yielding LOIs)

Linear Operator Inequalities (LOIs) in the PIE Formulation

PIE formulation of System:

All results on this page are for no input delays ($B_{T1} = 0$) and no process delays ($B_{T2} = 0$)

- Extension OK to input delays.

$$\begin{aligned}\mathcal{T}\dot{\mathbf{x}}(t) &= \mathcal{A}\mathbf{x}(t) + \mathcal{B}_1 w(t) + \mathcal{B}_2 u(t)\\ z(t) &= \mathcal{C}_1\mathbf{x}(t) + \mathcal{D}_{11} w(t) + \mathcal{D}_{12} u(t),\\ y(t) &= \mathcal{C}_2\mathbf{x}(t) + \mathcal{D}_{21} w(t) + \mathcal{D}_{22} u(t).\end{aligned}$$

KYP and H_∞-Gain

If there exists $\mathcal{P} = \mathcal{P}\left\{\begin{smallmatrix} P, & Q_1 \\ Q_2, & \{R_i\}\end{smallmatrix}\right\} \geq 0$ such that

$$\begin{bmatrix} -\gamma I & \mathcal{D}_{11}^* & \mathcal{B}_1^*\mathcal{P}\mathcal{T} \\ \mathcal{D}_1 & -\gamma I & \mathcal{C}_1 \\ \mathcal{T}^*\mathcal{P}\mathcal{B}_1 & \mathcal{C}_1^* & \mathcal{A}^*\mathcal{P}\mathcal{T} + \mathcal{T}^*\mathcal{P}\mathcal{A}\end{bmatrix} < 0$$

then $\|z\|_{L_2} \leq \gamma\|\omega\|_{L_2}$.

H_∞-Optimal Full State Feedback:

If there exist $\mathcal{P} = \mathcal{P}\left\{\begin{smallmatrix} P, & Q \\ Q^T, & \{R_i\}\end{smallmatrix}\right\} > 0$ and $\mathcal{Z} = \mathcal{P}\left\{\begin{smallmatrix} Z_1, & Z_2 \\ \emptyset, & \{\emptyset\}\end{smallmatrix}\right\}$ such that

$$\begin{bmatrix} -\gamma I & \mathcal{D}_{11} & (\mathcal{C}_1\mathcal{P} + \mathcal{D}_{12}\mathcal{Z})\mathcal{T}^* \\ *^* & -\gamma I & \mathcal{B}_1^* \\ *^* & *^* & (\mathcal{A}\mathcal{P} + \mathcal{B}_2\mathcal{Z})\mathcal{T}^* + \mathcal{T}(\mathcal{A}\mathcal{P} + \mathcal{B}_2\mathcal{Z})^*\end{bmatrix} < 0$$

then if $u(t) = \mathcal{Z}\mathcal{P}^{-1}\mathbf{x}(t)$, $\|z\|_{L_2} \leq \gamma\|\omega\|_{L_2}$.

H_∞-Optimal Estimator Design:

If there exist $\mathcal{P} = \mathcal{P}\left\{\begin{smallmatrix} P, & Q \\ Q^T, & \{R_i\}\end{smallmatrix}\right\} \geq 0$ and $\mathcal{Z} = \mathcal{P}\left\{\begin{smallmatrix} Z_1, & \emptyset \\ Z_2, & \{\emptyset\}\end{smallmatrix}\right\}$ such that

$$\begin{bmatrix} -\gamma I & -\mathcal{D}_{11}^* & -(\mathcal{P}\mathcal{B}_1 + \mathcal{Z}\mathcal{D}_{21})^*\mathcal{T} \\ *^* & -\gamma I & \mathcal{C}_1 \\ *^* & *^* & (\mathcal{P}\mathcal{A} + \mathcal{Z}\mathcal{C}_2)^*\mathcal{T} + \mathcal{T}^*(\mathcal{P}\mathcal{A} + \mathcal{Z}\mathcal{C}_2)\end{bmatrix} < 0$$

then if $\mathcal{L} = \mathcal{P}^{-1}\mathcal{Z}$, $\|\hat{z} - z\|_{L_2} \leq \gamma\|\omega\|_{L_2}$ where

$$\begin{aligned}\mathcal{T}\dot{\hat{\mathbf{x}}}(t) &= \mathcal{A}\hat{\mathbf{x}}(t) + \mathcal{L}(\hat{y}(t) - y(t))\\ \hat{y}(t) &= \mathcal{C}_2\hat{\mathbf{x}}(t) \quad \hat{z}(t) = \mathcal{C}_1\hat{\mathbf{x}}(t)\end{aligned}$$

PIETOOLS Code for solving Linear Operator Inequalities

PIE formulation of System:

```
pvar s,th,gam;
T = sosprogram([s,th],gam);
opvar A,B1,B2,C1,C2,D11,D12,D21,E;
A=..;B1=..;B2=..;C1=..;C2=..;
D11=..;D12=..;D21=..;E=..;
```

KYP and H_∞-Gain

```
[T,P] = sos_posopvar(T,dim,I,s,th);
D = [-gam*I   D11'      B1'*P*E;
     D11        -gam*I   C1;
     E'*P*B1  C1'       A'*P*E+E'*P*A];
T = sosopineq(T,D);
T = sossetobj(T,gam);
```

H_∞-Optimal Full State Feedback:

```
[T,P] = sos_posopvar(T,dim,I,s,th,deg);
[T,P] = sos_opvar(T,dim,I,s,th,deg);
M33 = (A*P+B2*Z)*E'+E*(A*P+B2*Z)'
M13 = (C1*P+D12*Z)*E'
D = [-gam*I   D11'      M13;
       D11    -gam*I     B1';
       M13'    B1       M33];
T = sosopineq(T,D);
T = sossetobj(T,gam);
T = sossolve(T);
```

H_∞-Optimal Estimator Design:

```
[T,P] = sos_posopvar(T,dim,I,s,th);
[T,P] = sos_opvar(T,dim,I,s,th,deg);
M33 = (P*A+Z*C2)'*E+E'*(P*A+Z*C2)
M13 = -B1'*P*E-D21'*Z'*E
D = [-gam*I   D11'      M13;
       D11    -gam*I     C1;
       M13'    C1'       M33];
T = sosopineq(T,D);
T = sossetobj(T,gam);
T = sossolve(T);
```

Advantages/Disadvantages of PIE formulation

Class of Systems Considered: Includes the DDF class of systems

Advantages:

- Actually Old (Barbasin type)
- Use of low dimensional channels
- Tools developed for ODEs can be applied
  - LMIs
  - Manipulation is easy

Disadvantages:

- Not many results in this area
  - Relatively New
  - Literature is Sparse
- Similar Structure to Singular Systems

A network control problem in the DDE formulation

$$\dot{x}(t) = A_0 x(t) + \sum_i A_i x(t-\tau_i) + B_1 w(t) + B_2 u(t), \quad y(t) = Cx(t) + D_1 w(t) + D_2 u(t)$$

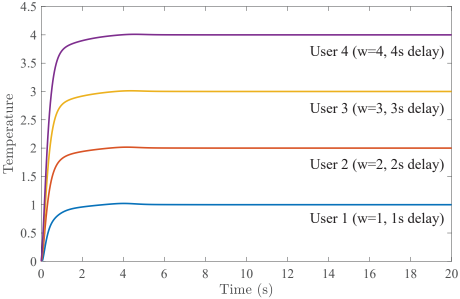
where

$$A_0 = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}, \quad A_i = \begin{bmatrix} 0 & 0 \\ 0 & \hat{A}_i \end{bmatrix}, \quad B_1 = \begin{bmatrix} -I \\ -\hat{\Gamma} + \mathrm{diag}(\alpha_1 \dots \alpha_K) \end{bmatrix}$$

$$\hat{A}_i(:,i) = \alpha_i \begin{bmatrix} \gamma_{i,1} & \dots & \gamma_{i,i-1} & -1 & \gamma_{i,i-1} & \dots & \gamma_{i,K} \end{bmatrix}^T$$

$$\hat{\Gamma}_{ij} = \alpha_j \gamma_{ij} = \begin{bmatrix} q_1 & \dots & q_K \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 \\ I \end{bmatrix}$$

$$C_0 = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}, \; C_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \; D_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \; D_2 = \begin{bmatrix} 0 \\ .1I \end{bmatrix}$$



Complexity: 8 states, 4 delays, 4 inputs, 4 disturbances, 8 regulated outputs

Results: A Matlab simulation of the step response of the closed-loop dynamics ($T_{2i}(t)$) with 4 users (w_i and τ_i as indicated) coupled with the controller with closed-loop gain of $.48$

DDF Formulation:

$$\dot{x}(t) = A_0 x(t) + B_1 w(t) + B_2 u(t) + B_v v(t)$$
$$z(t) = C_1 x(t) + D_{11} w(t) + D_{12} u(t) + D_{1v} v(t)$$
$$y(t) = C_2 x(t) + D_{21} w(t) + D_{22} u(t) + D_{2v} v(t)$$
$$r_i(t) = C_{ri} x(t) + B_{r1i} w(t) + B_{r2i} u(t) + D_{rvi} v(t)$$
$$v(t) = \sum_{i=1}^{K} C_{vi} r_i(t - \tau_i) + \sum_{i=1}^{K} \int_{-\tau_i}^{0} C_{vdi}(s) r_i(t+s) ds$$

PIE Formulation:

$$\begin{aligned} \mathcal{T}\dot{\mathbf{x}}(t) + \mathcal{B}_{T_1}\dot{w}(t) + \mathcal{B}_{T_2}\dot{u}(t) &= \mathcal{A}\mathbf{x}(t) + \mathcal{B}_1 w(t) + \mathcal{B}_2 u(t) \\ z(t) &= \mathcal{C}_1 \mathbf{x}(t) + \mathcal{D}_{11} w(t) + \mathcal{D}_{12} u(t), \\ y(t) &= \mathcal{C}_2 \mathbf{x}(t) + \mathcal{D}_{21} w(t) + \mathcal{D}_{22} u(t), \end{aligned} \tag{3}$$

Converting a DDF to a PIE

$$\mathcal{A} := \mathcal{P}\left\{\begin{smallmatrix} \mathbf{A}_0, & \mathbf{A} \\ 0, & \{I_\tau, 0, 0\}\end{smallmatrix}\right\}, \qquad \mathcal{T} := \mathcal{P}\left\{\begin{smallmatrix} I, & 0 \\ \mathbf{T}_0, & \{0, \mathbf{T}_a, \mathbf{T}_b\}\end{smallmatrix}\right\}$$

$$\mathcal{B}_1 := \mathcal{P}\left\{\begin{smallmatrix} \mathbf{B}_1, & \emptyset \\ 0, & \{\emptyset\}\end{smallmatrix}\right\}, \qquad \mathcal{B}_2 := \mathcal{P}\left\{\begin{smallmatrix} \mathbf{B}_2, & \emptyset \\ 0, & \{\emptyset\}\end{smallmatrix}\right\}, \qquad \mathcal{B}_{T_1} := \mathcal{P}\left\{\begin{smallmatrix} 0, & \emptyset \\ \mathbf{T}_1, & \{\emptyset\}\end{smallmatrix}\right\}, \qquad \mathcal{B}_{T_2} := \mathcal{P}\left\{\begin{smallmatrix} 0, & \emptyset \\ \mathbf{T}_2, & \{\emptyset\}\end{smallmatrix}\right\}$$

$$\mathcal{C}_1 := \mathcal{P}\left\{\begin{smallmatrix} \mathbf{C}_{10}, & \mathbf{C}_{11} \\ \emptyset, & \{\emptyset\}\end{smallmatrix}\right\}, \quad \mathcal{C}_2 := \mathcal{P}\left\{\begin{smallmatrix} \mathbf{C}_{20}, & \mathbf{C}_{21} \\ \emptyset, & \{\emptyset\}\end{smallmatrix}\right\}, \quad \mathcal{D}_{ij} := \mathcal{P}\left\{\begin{smallmatrix} \mathbf{D}_{ij}, & \emptyset \\ \emptyset, & \{\emptyset\}\end{smallmatrix}\right\}$$

$$\mathbf{T}_0 = \begin{bmatrix} C_{r1} + D_{rv1}C_{vx} \\ \vdots \\ C_{rK} + D_{rvK}C_{vx} \end{bmatrix}, \qquad \mathbf{T}_1 = \begin{bmatrix} B_{r11} + D_{rv1}D_{vw} \\ \vdots \\ B_{r1K} + D_{rvK}D_{vw} \end{bmatrix}, \qquad \mathbf{T}_2 = \begin{bmatrix} B_{r21} + D_{rv1}D_{vu} \\ \vdots \\ B_{r2K} + D_{rvK}D_{vu} \end{bmatrix}$$

$$\mathbf{T}_a(s) = \begin{bmatrix} D_{rv1}\begin{bmatrix} C_{I1}(s) & \cdots & C_{IK}(s)\end{bmatrix} \\ \vdots \\ D_{rvK}\begin{bmatrix} C_{I1}(s) & \cdots & C_{IK}(s)\end{bmatrix} \end{bmatrix}, \qquad \mathbf{T}_b = -I + \mathbf{T}_a(s), \quad I_\tau = \begin{bmatrix} \frac{1}{\tau_1}I & & \\ & \ddots & \\ & & \frac{1}{\tau_K}I \end{bmatrix},$$

$$\mathbf{A}_0 = A_0 + B_vC_{vx}, \; \mathbf{A}(s) = B_v\begin{bmatrix} C_{I1}(s) & \cdots & C_{IK}(s)\end{bmatrix}, \; \mathbf{B}_1 = B_1 + B_vD_{vw}, \; \mathbf{B}_2 = B_2 + B_vD_{vu},$$

$$\mathbf{C}_{10} = C_1 + D_{1v}C_{vx}, \qquad \mathbf{C}_{11} = D_{1v}\begin{bmatrix} C_{I1}(s) & \cdots & C_{IK}(s)\end{bmatrix},$$

$$\mathbf{C}_{20} = C_2 + D_{2v}C_{vx}, \qquad \mathbf{C}_{21} = D_{2v}\begin{bmatrix} C_{I1}(s) & \cdots & C_{IK}(s)\end{bmatrix},$$

$$\mathbf{D}_{11} = (D_{11} + D_{1v}D_{vw}), \; \mathbf{D}_{12} = (D_{12} + D_{1v}D_{vu}), \; \mathbf{D}_{21} = (D_{21} + D_{2v}D_{vw}), \; \mathbf{D}_{22} = (D_{22} + D_{2v}D_{vu})$$

$$\hat{C}_{vi} = C_{vi} + \int_{-1}^{0} \tau_i C_{vdi}(\tau_i s)ds, \qquad D_I = \left(I - \left(\sum_{i=1}^{K} \hat{C}_{vi}D_{rvi}\right)\right)^{-1}, \quad C_{vx} = D_I\left(\sum_{i=1}^{K} \hat{C}_{vi}C_{ri}\right)$$

$$D_{vw} = D_I\left(\sum_{i=1}^{K} \hat{C}_{vi}B_{r1i}\right), \; D_{vu} = D_I\left(\sum_{i=1}^{K} \hat{C}_{vi}B_{r2i}\right), \; C_{Ii}(s) = -D_I\left(C_{vi} + \tau_i\int_{-1}^{s} C_{vdi}(\tau_i\eta)d\eta\right)$$

Conclusion: What is the best way to represent a network?

The Last Slide (Thanks to NSF CNS-1739990)

Depends on your goal

- Control? Exploration? Consensus?

DDEs:

- Convenient for small or delay-free networks
- Can use Padé approximation
- Smith Predictors, SOS
- Lots of literature
- No Difference equations

DDFs:

- Can use Padé approximation
- Not much literature
- Can be used for large networks and lots of delays
- Can combine discrete/continuous time
- OK for static feedback

ODE-PDEs:

- All the advantages of DDFs
- Good for intuition
- Good if you know how to use backstepping
- Can apply PDE discretization schemes

PIEs:

- Can use intuition from ODEs
- Can use PIETOOLS
- good for optimal control
- MAY be good for simulation (no BC's)

Discussion?

More Options?

Illustration of H_∞ Gain Analysis

Example 1:

$$\dot{x}(t) = \begin{bmatrix} -2 & 0 \\ 0 & -.9 \end{bmatrix} x(t) + \begin{bmatrix} -1 & 0 \\ -1 & -1 \end{bmatrix} x(t-\tau) + \begin{bmatrix} -.5 \\ 1 \end{bmatrix} w(t), \quad y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t)$$

d	1	2	3	Padé	[Fridman 2001]	[Shaked 1998]
$\gamma_{\min}$	.2373	.2365	.2365	.2364	.32	2

Example 2: Stable for $\tau \in [.100173, 1.71785]$:

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -2 & .1 \end{bmatrix} x(t) + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} x(t-\tau) + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} w(t)$$

$$y(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} x(t)$$

We plot bounds for the H_∞ norm as the delay varies within this interval. As expected, the H_∞ norm approaches infinity quickly as we approach the limits of the stable region.

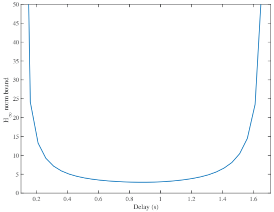


Figure: Calculated H_∞ norm bound vs. delay for Ex. 2

The Inverse of a 4-PI Operator is a 4-PI Operator!

Result from Keqin Gu

How to find (Note $R_1 = R_2$)

$$\mathcal{K} = \mathcal{P}\left\{\begin{smallmatrix} Z_1, & Z_2 \\ \emptyset, & \{\emptyset\} \end{smallmatrix}\right\}\mathcal{P}\left\{\begin{smallmatrix} P, & Q \\ Q^T, & \{S,\, R,\, R\} \end{smallmatrix}\right\}^{-1}?$$

Assume Q and R are polynomial

Extract Polynomial Coefficients: $Q(s) = HZ(s)$ and $R(s,\theta) = Z(s)^T \Gamma Z(\theta)$.

Then $\mathcal{P}\left\{\begin{smallmatrix} P, & Q \\ Q^T, & \{S,\, R,\, R\} \end{smallmatrix}\right\}^{-1} = \mathcal{P}\left\{\begin{smallmatrix} \hat{P}, & \hat{Q} \\ \hat{Q}^T, & \{\hat{S},\, \hat{R},\, \hat{R}\} \end{smallmatrix}\right\}$ where

$$\hat{P} = \left(I - \hat{H} V H^T\right) P^{-1}, \qquad \hat{Q}(s) = \frac{1}{\tau}\hat{H} Z(s) S(s)^{-1}$$

$$\hat{S}(s) = \frac{1}{\tau^2} S(s)^{-1} \qquad \hat{R}(s,\theta) = \frac{1}{\tau} S(s)^{-1} Z(s)^T \hat{\Gamma} Z(\theta) S(\theta)^{-1},$$

where

$$\hat{H} = P^{-1} H \left(V H^T P^{-1} H - I - V\Gamma\right)^{-1}$$

$$\hat{\Gamma} = -(\hat{H}^T H + \Gamma)(I + V\Gamma)^{-1},$$

$$V = \int_{-\tau}^{0} Z(s) S(s)^{-1} Z(s)^T ds$$

Boring Numerical Controller Synthesis Examples

$$\dot{x}(t) = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} -1 & -1 \\ 0 & -.9 \end{bmatrix} x(t-\tau) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} w(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$

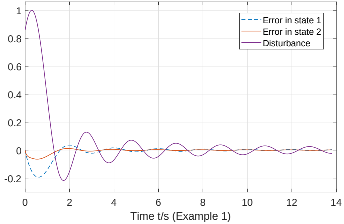
$$y(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ .1 \end{bmatrix} u(t)$$

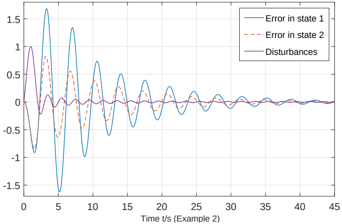
d	1	2	3	Padé	Fridman 2003	Li 1997
$\gamma_{\min}(\tau = .999)$	.10001	.10001	.10001	.1000	.22844	1.8822
$\gamma_{\min}(\tau = 2)$	1.43	1.36	1.341	1.340	∞	∞
CPU sec	.478	.879	2.48	2.78	N/A	N/A

$$\dot{x}(t) = \begin{bmatrix} 2 & 1 \\ 0 & -1 \end{bmatrix} x(t) + \begin{bmatrix} -1 & 0 \\ -1 & 1 \end{bmatrix} x(t-\tau) + \begin{bmatrix} -.5 \\ 1 \end{bmatrix} w(t) + \begin{bmatrix} 3 \\ 1 \end{bmatrix} u(t)$$

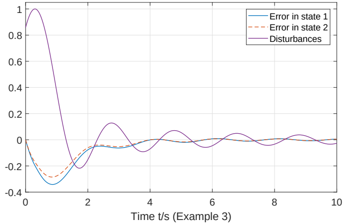
$$y(t) = \begin{bmatrix} 1 & -.5 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$

d	1	2	3	Padé
$\gamma_{\min}(\tau = .3)$	.3953	.3953	.3953	.3953
CPU sec	.655	1.248	2.72	2.91





γ_{min}	Example 1			Example 2			Example 3		
	d=1	d=2	d=4	d=1	d=2	d=4	d=1	d=2	d=4
using simplified estimator	0.2371	0.23651	0.23608	7.2111			0.2264		
using generalized estimator	0.2357			7.2111			0.2264		
Padé	0.2357			7.2107			0.2264		



Systems with Input Delays (Control at the Boundary)

- When there is only one input delay, we may alternatively design an estimator using a delayed output (which then becomes a predictor) and is stable in closed-loop using the separation principle
- Control at the boundary is slightly more complex than in-domain control.

$$\mathbf{x}_p = \mathcal{T}\mathbf{x}_f + \mathcal{B}_{T2}u(t) \qquad u(t) = \mathcal{K}\mathbf{x}_f(t)$$

Replace $\mathcal{T} \to \mathcal{T} + \mathcal{B}_{T2}\mathcal{K}$

$$\begin{bmatrix} -\gamma I & \mathcal{D}_1 & (\mathcal{C}\mathcal{P} + \mathcal{D}_2\mathcal{Z})(\mathcal{T} + \mathcal{B}_{T2}\mathcal{K})^* \\ \mathcal{D}_1^T & -\gamma I & \mathcal{B}_1^* \\ (\mathcal{T} + \mathcal{B}_{T2}\mathcal{K})(\mathcal{C}\mathcal{P} + \mathcal{D}_2\mathcal{Z})^* & \mathcal{B}_1 & (\mathcal{A}\mathcal{P} + \mathcal{B}_2\mathcal{Z})(\mathcal{T} + \mathcal{B}_{T2}\mathcal{K})^* + (\mathcal{T} + \mathcal{B}_{T2}\mathcal{K})(\mathcal{A}\mathcal{P} + \mathcal{B}_2\mathcal{Z})^* \end{bmatrix} < 0$$