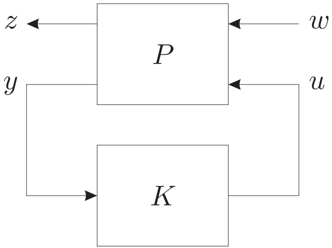
SOS for Systems with Multiple Delays Part 2. H_∞-Optimal Estimation

Matthew M. Peet and K. Gu
Arizona State University
Tempe, AZ USA

American Control Conference 2019
Philadelphia, PA

July 11, 2019



$$\dot{x}(t) = Ax(t) + B_1 w(t) + B_2 u(t),$$
$$z(t) = C_1 x(t) + D_{11} w(t) + D_{12} u(t)$$
$$y(t) = C_2 x(t) + D_{21} w(t) + D_{22} u(t)$$

- z is regulated output
- y is measured output
- w is disturbance
- u is actuation

H_∞-Optimal Full State Feedback:

There exist $P > 0$ and Z such that

$$\begin{bmatrix} PA^T + AP + Z^T B_2^T + B_2 Z & *^T & *^T \\ B_1^T & -\gamma I & *^T \\ C_1 P + D_{12} Z & D_{11} & -\gamma I \end{bmatrix} < 0$$

Then if $u(t) = ZP^{-1}x(t)$, $\|z\|_{L_2} \le \gamma \|w\|_{L_2}$.

H_∞-Optimal Estimator Design:

There exist $P > 0$ and Z such that

$$\begin{bmatrix} PA + ZC_2 + (PA + ZC_2)^T & *^T & *^T \\ -(PB + ZD_{21})^T & -\gamma I & *^T \\ C_1 & D_{11} & -\gamma I \end{bmatrix} < 0$$

Then if $u = 0$ and

$$\dot{\hat{x}}(t) = A\hat{x}(t) + P^{-1}Z(C_2\hat{x}(t) - y(t))$$
$$z_e(t) = C_1(\hat{x}(t) - x(t))$$

we have $\|z_e\|_{L_2} \le \gamma \|w\|_{L_2}$.

What to do about Time-Delay Systems?

Nominal Form:

$$\dot{x}(t) = A_0 x(t) + \sum_{i=1}^{K} A_i x(t-\tau_i) + Bw(t)$$

$$z(t) = C_{10} x(t) + \sum_{i=1}^{K} C_{1i} x(t-\tau_i) + D_1 w(t) \qquad \text{Regulated Output}$$

$$y(t) = C_{20} x(t) + \sum_{i=1}^{K} C_{2i} x(t-\tau_i) + D_2 w(t) \qquad \text{Sensed Output}$$

- $w(t) \in \mathbb{R}^r$ is disturbance
- $y(t) \in \mathbb{R}^q$ are sensor measurements
- $x(t) \in \mathbb{R}^n$ is the state
- $z(t) \in \mathbb{R}^p$ is regulated output

We Solve:

- H_∞-Optimal Observer Synthesis

Big Picture Goal: Treat the Time-Delay System like an ODE!

$$\mathcal{T}\dot{\mathbf{x}}(t) = \mathcal{A}\mathbf{x}(t) + \mathcal{B}w(t) \qquad \mathbf{x}(t) = \begin{bmatrix} x(t) \\ x_s(t+s) \end{bmatrix}$$

$$y(t) = \mathcal{C}_2\mathbf{x}(t) + \mathcal{D}_2 w(t) \quad z(t) = \mathcal{C}_1\mathbf{x}(t) + \mathcal{D}_1 w(t)$$

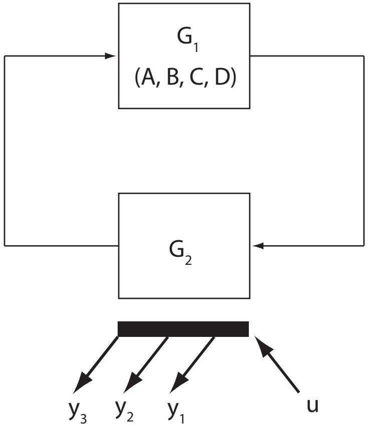
For this we need an **Algebraic Representation**!

A linear time-delay system is the interconnection of an ODE and a simple transport PDE with point actuation and point observation.

ODE: The system G_1

$$\dot{x}_1(t) = Ax_1(t) + Bu_1(t)$$
$$u_2(t) = Cx_1(t) + Du_1(t)$$

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \left[\begin{array}{c|c} A_0 & \begin{bmatrix} A_1 & \cdots & A_n \end{bmatrix} \\ \hline I & 0 \end{array}\right]$$



PDE: The system G_2

$$\frac{\partial}{\partial t}\phi(t,s) = \frac{\partial}{\partial s}\phi(t,s) \quad \phi(t,0) = u_2(t),$$

$$u_1(t) = \begin{bmatrix} \phi(-\tau_1) \\ \vdots \\ \phi(-\tau_K) \end{bmatrix}$$

Of course, the solution is just $x_2(t,s) = u_2(t-s)$.

Step 1: The ODE-PDE Representation (with BC's)

The Following Systems are Equivalent:

Standard TDS Form:

$$\dot{x}(t) = A_0 x(t) + \sum_{i=1}^{K} A_i x(t-\tau_i) + Bw(t)$$

$$z(t) = C_{10} x(t) + \sum_{i=1}^{K} C_{1i} x(t-\tau_i) + D_1 w(t)$$

$$y(t) = C_{20} x(t) + \sum_{i=1}^{K} C_{2i} x(t-\tau_i) + D_2 w(t)$$

Coupled ODE-PDE Form: (Denote $\phi_{i,s}(t,s) = \frac{\partial}{\partial s}\phi_i(t,s)$)

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\phi}_i(t,s) \end{bmatrix} = \begin{bmatrix} A_0 x(t) + \sum_{i=1}^{K} A_i \phi_i(t,-1) \\ \frac{1}{\tau_i}\phi_{i,s}(t,s) \end{bmatrix} + \begin{bmatrix} Bw(t) \\ 0 \end{bmatrix}, \qquad \phi_i(t,0) = x(t)$$

$$z(t) = C_{10} x(t) + \sum_{i=1}^{K} C_{1i} \phi_i(t,-1) + D_1 w(t)$$

$$y(t) = C_{20} x(t) + \sum_{i=1}^{K} C_{2i} \phi_i(t,-1) + D_2 w(t)$$

Problem: How to represent the Boundary Condition $\phi_i(t,0) = x(t)$???

Step 2: The Partial Integral Equation (PIE) Representation

Fundamental Theorem of Calculus:

$$\phi(s) = \phi(0) - \int_s^0 \phi_s(\eta) d\eta$$

Hence (since $\phi(t,0) = x(t)$)

$$\phi(t,-1) = x(t) - \int_{-1}^0 \phi_s(t,\eta) d\eta \qquad \text{and} \qquad \phi(t,s) = x(t) - \int_s^0 \phi_s(t,\eta) d\eta$$

Partial Integral Equation (PIE) Form of a TDS (No BCs):

$$\begin{bmatrix} \dot{x}(t) \\ \dot{x}(t) - \int_s^0 \dot{\phi}_{i,s}(t,\eta) d\eta \end{bmatrix} = \begin{bmatrix} (A_0 + \sum_{i=1}^K A_i) x(t) - \int_{-1}^0 \sum_{i=1}^K A_i \phi_{i,s}(t,\eta) d\eta \\ \phi_{i,s}(t,s) \end{bmatrix} + \begin{bmatrix} Bw(t) \\ 0 \end{bmatrix}$$

$$z(t) = \left(C_{10} + \sum_{i=1}^K C_{1i} \right) x(t) - \int_{-1}^0 \sum_{i=1}^K C_{1i} \phi_{i,s}(t,\eta) d\eta + D_1 w(t)$$

$$y(t) = \left(C_{20} + \sum_{i=1}^K C_{2i} \right) x(t) - \int_{-1}^0 \sum_{i=1}^K C_{2i} \phi_{i,s}(t,\eta) d\eta + D_2 w(t)$$

Step 2: Define the **New State Variable:** Φ

PIE Form of a TDS, Simplified:

$$\begin{bmatrix} \dot{x}(t) \\ \mathbf{1}_K \dot{x}(t) - \int_s^0 \dot{\mathbf{\Phi}}(t,\eta)d\eta \end{bmatrix} = \begin{bmatrix} \mathbf{A}_0 x(t) + \int_{-1}^0 \mathbf{A}\mathbf{\Phi}(t,\eta)d\eta \\ I_\tau \mathbf{\Phi}(t,s) \end{bmatrix} + \begin{bmatrix} Bw(t) \\ 0 \end{bmatrix}$$

$$z(t) = \mathbf{C}_{10}x(t) + \int_{-1}^0 \mathbf{C}_{11}\mathbf{\Phi}(t,\eta)d\eta + D_1 w(t)$$

$$y(t) = \mathbf{C}_{20}x(t) + \int_{-1}^0 \mathbf{C}_{21}\mathbf{\Phi}(t,\eta)d\eta + D_2 w(t)$$

---

where

$$\mathbf{\Phi} = \begin{bmatrix} \phi_{1,s} \\ \vdots \\ \phi_{K,s} \end{bmatrix}, \qquad I_\tau = \begin{bmatrix} \frac{1}{\tau_1} I & & \\ & \ddots & \\ & & \frac{1}{\tau_K} I \end{bmatrix}, \qquad \mathbf{1}_K = \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix},$$

$$\mathbf{A}_0 = A_0 + \sum_{i=1}^K A_i, \qquad \mathbf{A} = -\begin{bmatrix} A_1 & \cdots & A_K \end{bmatrix},$$

$$\mathbf{C}_{20} = C_{20} + \sum_{i=1}^K C_{2i}, \qquad \mathbf{C}_{10} = C_{10} + \sum_{i=1}^K C_{1i}$$

$$\mathbf{C}_{21} = -\begin{bmatrix} C_{21} & \cdots & C_{2K} \end{bmatrix}, \qquad \mathbf{C}_{11} = -\begin{bmatrix} C_{11} & \cdots & C_{1K} \end{bmatrix},$$

Step 3: Express Dynamics using 4-PI Operators

Definition of a 4-PI Operator $\left(\mathcal{P}\left\{\begin{smallmatrix} P, & Q_1 \\ Q_2, & \{R_i\}\end{smallmatrix}\right\}\right)$: $\mathbb{R} \times L_2 \to \mathbb{R} \times L_2$

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

4-PI Operators include a 3-PI Operator, Defined as:

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

Clean PIE Representation of a TDS:

Define the fundamental State: $\mathbf{x}(t) = \begin{bmatrix} x(t) \\ \mathbf{\Phi}(t,\cdot)\end{bmatrix}$.

$$\begin{aligned} \mathcal{T}\dot{\mathbf{x}}(t) &= \mathcal{A}\mathbf{x}(t) + \mathcal{B}w(t) \\ z(t) &= \mathcal{C}_1\mathbf{x}(t) + \mathcal{D}_1 w(t), \qquad y(t) = \mathcal{C}_2\mathbf{x}(t) + \mathcal{D}_2 w(t)\end{aligned}$$

$\mathcal{T} := \mathcal{P}\left\{\begin{smallmatrix} I, & 0 \\ \mathbf{1}_K, & \{0,0,-I\}\end{smallmatrix}\right\}$ $\quad\mathcal{A} := \mathcal{P}\left\{\begin{smallmatrix} \mathbf{A}_0, & \mathbf{A} \\ 0, & \{I_\tau, 0, 0\}\end{smallmatrix}\right\}$, $\quad\mathcal{C}_1 := \mathcal{P}\left\{\begin{smallmatrix} \mathbf{C}_{10}, & \mathbf{C}_{11} \\ \emptyset, & \{\emptyset\}\end{smallmatrix}\right\}$, $\quad\mathcal{C}_2 := \mathcal{P}\left\{\begin{smallmatrix} \mathbf{C}_{20}, & \mathbf{C}_{21} \\ \emptyset, & \{\emptyset\}\end{smallmatrix}\right\}$,

$\mathcal{B} := \mathcal{P}\left\{\begin{smallmatrix} B, & \emptyset \\ 0, & \{\emptyset\}\end{smallmatrix}\right\}$, $\quad\mathcal{D}_1 := \mathcal{P}\left\{\begin{smallmatrix} D_1, & \emptyset \\ \emptyset, & \{\emptyset\}\end{smallmatrix}\right\}$, $\quad\mathcal{D}_2 := \mathcal{P}\left\{\begin{smallmatrix} D_2, & \emptyset \\ \emptyset, & \{\emptyset\}\end{smallmatrix}\right\}$

4-PI Operators also define Complete Quadratic Lyapunov Krasovskii Functionals

The Complete-Quadratic L-K Functional:

$$\begin{aligned} V(\phi) =& \phi(0)^T P \phi(0) + \sum_{i=1}^{K} \int_{-\tau_i}^{0} \phi(0)^T Q_i(s) \phi(s) ds + \sum_{i=1}^{K} \int_{-\tau_i}^{0} \phi(s)^T Q_i(s)^T \phi(0) ds \\ &+ \sum_{i=1}^{K} \int_{-\tau_i}^{0} \phi(s)^T S_i(s) \phi(s) + \sum_{i,j=1}^{K} \int_{-\tau_i}^{0} \int_{-\tau_j}^{0} \phi(s)^T R_{ij}(s,\theta) \phi(\theta) d\theta \end{aligned}$$

Define $a_i = \frac{\tau_i}{\tau_K}$, $\hat{P} = P$ and

$$\hat{Q}(s) := \begin{bmatrix} \sqrt{a_1} Q_1(a_1 s) & \cdots & \sqrt{a_K} Q_K(a_K s) \end{bmatrix}, \quad \hat{S}(s) := \begin{bmatrix} S_1(a_1 s) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & S_K(a_K s) \end{bmatrix}$$

$$\hat{R}(s,\theta) := \begin{bmatrix} \sqrt{a_1 a_1} R_{11}(s a_1, \theta a_1) & \cdots & \sqrt{a_1 a_K} R_{1K}(s a_1, \theta a_K) \\ \vdots & \cdots & \vdots \\ \sqrt{a_K a_1} R_{K1}(s a_K, \theta a_1) & \cdots & \sqrt{a_K a_K} R_{KK}(s a_K, \theta a_K) \end{bmatrix}.$$

Then $V(\phi) \geq 0$ **IF AND ONLY IF** $\mathcal{P}\left\{ {}_{\hat{Q}_2, \{\hat{S}, \hat{R}, \hat{R}\}}^{P, \; \hat{Q}_1} \right\} \geq 0$

4-PI Operators have a well-define Matlab structure

A general operator on $\mathcal{P}\left\{\begin{smallmatrix} P, & Q_1 \\ Q_2, & \{R_i\}\end{smallmatrix}\right\} : \mathbb{R}^p \times L_2^q[a,b] \to \mathbb{R}^m \times L_2^n[a,b]$

$$\left(\mathcal{P}\left\{\begin{smallmatrix} P, & Q_1 \\ Q_2, & \{R_i\}\end{smallmatrix}\right\} \begin{bmatrix} x \\ \mathbf{x} \end{bmatrix}\right)(s) := \begin{bmatrix} Px + \int_a^b Q_1(s)\mathbf{x}(s)ds \\ Q_2(s)x + \left(\mathcal{P}_{\{R_i\}}\mathbf{x}\right)(s) \end{bmatrix}.$$

MATLAB structure has following elements.

1. `opvar P`: declares P to be a 4-PI operator object.
2. `P.P`: a $m \times p$ matrix
3. `P.Q1, P.Q2`: $m \times q$ and $n \times p$ matrix valued polynomials in s, respectively
4. `P.R`: a 3-PIE structure containing R_0, R_1, and R_2
5. `P.R.R0` : $n \times q$ matrix valued polynomial in s
6. `P.R.R1, P.R.R2` : $n \times q$ matrix valued polynomials in s and θ
7. `P.dim`: $\begin{bmatrix} m & p \\ n & q \end{bmatrix}$.
8. `P.I`: [a,b].
9. `P.var1`: s (default)
10. `P.var2`: th (default)

3-PI $\mathcal{P}_{\{N_i\}}$ Operators Form an Algebra

Property 1: Composition

$$\mathcal{P}_{\{R_i\}} = \mathcal{P}_{\{B_i\}}\mathcal{P}_{\{N_i\}}$$

where

$$\begin{aligned}
R_0(s) &= B_0(s)N_0(s)\\
R_1(s,\theta) &= B_0(s)N_1(s,\theta) + B_1(s,\theta)N_0(\theta) + \int_a^\theta B_1(s,\xi)N_2(\xi,\theta)d\xi\\
&\qquad + \int_\theta^s B_1(s,\xi)N_1(\xi,\theta)d\xi + \int_s^b B_2(s,\xi)N_1(\xi,\theta)d\xi\\
R_2(s,\theta) &= B_0(s)N_2(s,\theta) + B_2(s,\theta)N_0(\theta) + \int_a^s B_1(s,\xi)N_2(\xi,\theta)d\xi\\
&\qquad + \int_s^\theta B_2(s,\xi)N_2(\xi,\theta)d\xi + \int_\theta^b B_2(s,\xi)N_1(\xi,\theta)d\xi
\end{aligned}$$

Triple Notation:

$$\{R_i\} = \{B_i\} \times \{N_i\}$$

Matlab Implementation:

$$\{N_i\} = \{T_i\} \times \{R_i\} \quad \rightarrow \quad \mathcal{P}_{\{N_i\}} = \mathcal{P}_{\{T_i\}}\mathcal{P}_{\{R_i\}}$$

```
opvar T R
T.R.R0=...; T.R.R1=...; T.R.R2=...; T.dim=[0 0;m n]; T.I=[-1,0]
R.R.R0=...; R.R.R1=...; R.R.R2=...; R.dim=[0 0;n q]; R.I=[-1,0]
N=T*R
```

4-PI Operators Form an Algebra

$$\mathcal{P}\left\{\begin{smallmatrix} L, & M_1 \\ M_2, & \{N_i\}\end{smallmatrix}\right\}\mathcal{P}\left\{\begin{smallmatrix} P, & Q_1 \\ Q_2, & \{R_i\}\end{smallmatrix}\right\}\begin{bmatrix} x \\ \mathbf{\Phi}\end{bmatrix}$$

$$= \begin{bmatrix} \left(LP + \int_a^b M_1(\nu)Q_2(\nu)d\nu\right)x + \int_a^b LQ_1(\nu)\mathbf{\Phi}(\nu)d\nu + \int_a^b M_1(\nu)\left(\mathcal{P}_{\{R_i\}}\mathbf{\Phi}\right)(\nu)d\nu \\ \left(M_2(s)P + \left(\mathcal{P}_{\{N_i\}}Q_2\right)(s)\right)x + M_2(s)\int_a^b Q_1(\nu)\mathbf{\Phi}(\nu)d\nu + \left(P_{\{N_i\}}\mathcal{P}_{\{R_i\}}\mathbf{\Phi}\right)(s)\end{bmatrix}$$

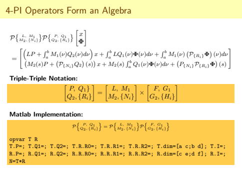
Triple-Triple Notation:

$$\begin{bmatrix} P, & Q_1\} \\ Q_2, & \{R_i\}\end{bmatrix} = \begin{bmatrix} L, & M_1 \\ M_2, & \{N_i\}\end{bmatrix} \times \begin{bmatrix} F, & G_1 \\ G_2, & \{H_i\}\end{bmatrix}$$

Matlab Implementation:

$$\mathcal{P}\left\{\begin{smallmatrix} P, & Q_1 \\ Q_2, & \{R_i\}\end{smallmatrix}\right\} = \mathcal{P}\left\{\begin{smallmatrix} L, & M_1 \\ M_2, & \{N_i\}\end{smallmatrix}\right\}\mathcal{P}\left\{\begin{smallmatrix} F, & G_1 \\ G_2, & \{H_i\}\end{smallmatrix}\right\}$$

```
opvar T R
T.P=; T.Q1=; T.Q2=; T.R.R0=; T.R.R1=; T.R.R2=; T.dim=[a c;b d]; T.I=;
R.P=; R.Q1=; R.Q2=; R.R.R0=; R.R.R1=; R.R.R2=; R.dim=[c e;d f]; R.I=;
N=T*R
```



- The composition property is surprising and non-trivial.
- Two integrations can be expressed using a single integral.
- Two derivatives can NOT be expressed using a single derivative.

Transpose/Adjoint in the 4-PI $\mathcal{P}\left\{\begin{smallmatrix} P, & Q_1 \\ Q_2, & \{R_i\}\end{smallmatrix}\right\}$ Operator Algebra

Property 2: Transpose/Adjoint

$$\left\langle \mathbf{x}, \mathcal{P}\left\{\begin{smallmatrix} \hat{P}, & \hat{Q}_1 \\ \hat{Q}_2, & \{\hat{R}_i\}\end{smallmatrix}\right\}\mathbf{y}\right\rangle_{\mathbb{R}^n\times L_2} = \left\langle \mathcal{P}\left\{\begin{smallmatrix} P, & Q_1 \\ Q_2, & \{R_i\}\end{smallmatrix}\right\}\mathbf{x}, \mathbf{y}\right\rangle_{\mathbb{R}^n\times L_2}$$

where

$$\hat{P} = P^T, \quad \hat{Q}_1(s) = Q_2(s)^T, \quad \hat{Q}_2(s) = Q_1(s)^T,$$
$$\hat{R}_0(s) = R_0(s)^T, \quad \hat{R}_1(s,\eta) = R_2(\eta,s)^T, \quad \hat{R}_2(s,\eta) = R_1(\eta,s)^T$$

Property 3: Addition

$$\mathcal{P}\left\{\begin{smallmatrix} P+L, & Q_1+M_1 \\ Q_2+M_2, & \{R_i+N_i\}\end{smallmatrix}\right\} = \mathcal{P}\left\{\begin{smallmatrix} P, & Q_1 \\ Q_2, & \{R_i\}\end{smallmatrix}\right\} + \mathcal{P}\left\{\begin{smallmatrix} L, & M_1 \\ M_2, & \{N_i\}\end{smallmatrix}\right\}$$

Matlab Implementation:

```
opvar T
T.P=...; T.Q1=...;T.Q2=...; T.R.R0=...; T.R.R1=...; T.R.R2=...;
T.dim=[p q;m n]; T.I=[-tau,0]; a=2;
R.P=...; R.Q1=...;R.Q2=...; R.R.R0=...; R.R.R1=...; R.R.R2=...;
R.dim=[p q;m n];
N=T';
N=T+R;
```

Transpose/Adjoint in the 4-PI $\mathcal{P}\left\{\begin{smallmatrix} P, & Q_1 \\ Q_2, & \{R_i\}\end{smallmatrix}\right\}$ Operator Algebra

Property 2: Transpose/Adjoint

$$\langle \mathbf{x}, \mathcal{P}\left\{\begin{smallmatrix} \hat{P}, & \hat{Q}_1 \\ \hat{Q}_2, & \{\hat{R}_i\}\end{smallmatrix}\right\}\mathbf{y}\rangle_{\mathbb{R}^n \times L_2} = \langle \mathcal{P}\left\{\begin{smallmatrix} P, & Q_1 \\ Q_2, & \{R_i\}\end{smallmatrix}\right\}\mathbf{x}, \mathbf{y}\rangle_{\mathbb{R}^n \times L_2}$$

where

$$\hat{P} = P^T, \quad \hat{Q}_1(s) = Q_2(s)^T, \quad \hat{Q}_2(s) = Q_1(s)^T,$$
$$\hat{R}_0(s) = R_0(s)^T, \quad \hat{R}_1(s,\eta) = R_2(\eta,s)^T, \quad \hat{R}_2(s,\eta) = R_1(\eta,s)^T$$

Property 3: Addition

$$\mathcal{P}\left\{\begin{smallmatrix} P+L, & Q_1+M_1 \\ Q_2+M_2, & \{R_i+N_i\}\end{smallmatrix}\right\} = \mathcal{P}\left\{\begin{smallmatrix} P, & Q_1 \\ Q_2, & \{R_i\}\end{smallmatrix}\right\} + \mathcal{P}\left\{\begin{smallmatrix} L, & M_1 \\ M_2, & \{N_i\}\end{smallmatrix}\right\}$$

Matlab Implementation:

```
opvar T
T.P=...; T.Q1=...;T.Q2=...; T.R.R0=...; T.R.R1=...; T.R.R2=...;
T.dim=[p q;m n]; T.I=[-tau,0]; a=2;
R.P=...; R.Q1=...;R.Q2=...; R.R.R0=...; R.R.R1=...; R.R.R2=...;
R.dim=[p q;m n];
N=T';
N=T+R;
```

- Note that N.dim will be [q p; n m].
- The inner product on $\mathbb{R}^n \times L_2$ is

$$\langle x, y\rangle_{\mathbb{R}\times L_2} = x_1^T y_1 + \int_{-\tau}^{0} x_2(s)^T y_2(s) ds$$

PIE Dynamics:

$$\mathcal{T}\dot{\mathbf{x}}_f(t) = \mathcal{A}\mathbf{x}(t)$$

We now propose a Lyapunov function of the form

$$V(\mathbf{x}) = \langle \mathcal{T}\mathbf{x}, \mathcal{P}\mathcal{T}\mathbf{x}\rangle = \langle \mathbf{x}_p, \mathcal{P}\mathbf{x}_p\rangle$$

The time-derivative of the Lyapunov function is

$$\begin{aligned}\dot{V}(\mathbf{x}(t)) &= \langle \mathcal{T}\mathbf{x}, \mathcal{P}\mathcal{T}\dot{\mathbf{x}}\rangle + \langle \mathcal{T}\dot{\mathbf{x}}, \mathcal{P}\mathcal{T}\mathbf{x}\rangle \\ &= \langle \mathcal{T}\mathbf{x}, \mathcal{P}\mathcal{A}\mathbf{x}\rangle + \langle \mathcal{A}\mathbf{x}, \mathcal{P}\mathcal{T}\mathbf{x}\rangle \\ &= \langle \mathbf{x}, \mathcal{T}^*\mathcal{P}\mathcal{A}\mathbf{x}\rangle + \langle \mathbf{x}, \mathcal{A}^*\mathcal{P}\mathcal{T}\mathbf{x}\rangle \\ &= \langle \mathbf{x}, (\mathcal{T}^*\mathcal{P}\mathcal{A} + \mathcal{A}^*\mathcal{P}\mathcal{T})\,\mathbf{x}\rangle\end{aligned}$$

Stability Condition: $\mathcal{P} > 0$ and

$$\mathcal{T}^*\mathcal{P}\mathcal{A} + \mathcal{A}^*\mathcal{P}\mathcal{T} \le 0$$

LMI Equivalent:
Descriptor Systems:

$$E\dot{x}(t) = Ax(t)$$

$$V(x) = x^T E^T P E x$$

$$\begin{aligned}\dot{V}(x_p) &= \dot{x}^T E^T P E x \\ &\quad + x^T E^T P E \dot{x} \\ &= x^T (E^T P A + A^T P E) x\end{aligned}$$

$$E^T P A + A^T P E < 0$$

An LMI for Positivity of 4-PI Operators

Positivity is an LMI constraint on the coefficients of polynomials $\begin{bmatrix} P, & Q_1 \\ Q_2, & \{R_i\} \end{bmatrix}$.

Theorem 1.

Suppose

$$\begin{bmatrix} P, & Q_1 \\ Q_2, & \{R_i\} \end{bmatrix} = \begin{bmatrix} I, & 0 \\ 0, & \{Z_i\} \end{bmatrix}^* \times \begin{bmatrix} P_1, & P_2 \\ P_2^T, & \{P_3, 0, 0\} \end{bmatrix} \times \begin{bmatrix} I, & 0 \\ 0, & \{Z_i\} \end{bmatrix}$$

where $P = \begin{bmatrix} P_1 & P_2 \\ P_2^T & P_3 \end{bmatrix} \geq 0$ *and*

$$\{Z_i\} := \left\{ \begin{bmatrix} \sqrt{g(s)} Z_{d1}(s) \\ \\ \end{bmatrix}, \begin{bmatrix} \sqrt{g(s)} Z_{d2}(s,\theta) \end{bmatrix}, \begin{bmatrix} \\ \sqrt{g(s)} Z_{d2}(s,\theta) \end{bmatrix} \right\}.$$

where $g(s) = s(-1-s)$ *or* $g = 1$ *and* Z_d *is the vector of monomials. Then* $\mathcal{P}\left\{\begin{smallmatrix} P, & Q_1 \\ Q_2, & \{R_i\} \end{smallmatrix}\right\} \geq 0$.

The LMI Tests Existence of a 4-PI Square Root

Matlab Implementation:

```
[prog, P] = sosposop_RL2RL(prog,[nR nL],X,s,th,[d1 d2]);
```

implies $\mathcal{P}\left\{\begin{smallmatrix} P.P, & P.Q1 \\ P.Q2, & \{P.R\} \end{smallmatrix}\right\} \geq 0$

An LMI for Positivity of 4-PI Operators

Positivity is an LMI constraint on the coefficients of polynomials $\begin{bmatrix} P, & Q_1 \\ Q_2, & \{R_i\} \end{bmatrix}$.

Theorem 1.

Suppose

$$\begin{bmatrix} P, & Q_1 \\ Q_2, & \{R_i\} \end{bmatrix} = \begin{bmatrix} I, & 0 \\ 0, & \{Z_i\} \end{bmatrix}^* \times \begin{bmatrix} P_1, & P_2 \\ P_2^T, & \{P_3, 0, 0\} \end{bmatrix} \times \begin{bmatrix} I, & 0 \\ 0, & \{Z_i\} \end{bmatrix}$$

where $P = \begin{bmatrix} P_1 & P_2 \\ P_2^T & P_3 \end{bmatrix} \geq 0$ *and*

$$\{Z_i\} := \left\{ \begin{bmatrix} \sqrt{g(s)} Z_{d1}(s) & & \\ & & \end{bmatrix}, \begin{bmatrix} & \sqrt{g(s)} Z_{d2}(s,\theta) & \end{bmatrix}, \begin{bmatrix} & & \\ & & \sqrt{g(s)} Z_{d2}(s,\theta) \end{bmatrix} \right\}.$$

where $g(s) = s(-1-s)$ *or* $g = 1$ *and* Z_d *is the vector of monomials. Then* $\mathcal{P}\left\{\begin{smallmatrix} P, & Q_1 \\ Q_2, & \{R_i\} \end{smallmatrix}\right\} \geq 0$.

The LMI Tests Existence of a 4-PI Square Root

Matlab Implementation:

```
[prog, P] = sosposop_RL2RL(prog,[nR nL],X,s,th,[d1 d2]);
```

implies $\mathcal{P}\left\{\begin{smallmatrix} P.P, & P.Q1 \\ P.Q2, & \{P.R\} \end{smallmatrix}\right\} \geq 0$

Positivity of a 4-PIE operator represents the most general form of inequality

- All existing inequalities for LMI methods for linear TDS are special cases
  - Each inequality corresponds to a specific choice of P.
- Jensen's Inequality is a special Case

$$\int_a^b f^2(x)dx - \int_a^b \int_a^b f(x)f(y)dxdy \geq 0 \qquad \leftrightarrows \qquad \mathcal{P}\left\{ \begin{smallmatrix} 0, & 0 \\ 0, & \{I, -I, -I\} \end{smallmatrix} \right\} \geq 0$$

- Wirtinger's inequality is a special case.
- Poincare's inequality is a special case (If we include the $\mathcal{T}$ operator).
- Bessel's inequality is a special case.

Matlab Toolbox Implementation (Stability Analysis)

Almost Complete Matlab Code:

```
pvar s th; opvar A T
A=...
T=...;
X=[-tau,0];
prog = sosprogram([s th])
[prog, P] = sosposop_RL2RL(prog,[nR nL],X,s,th,[d1 d2]);
[prog, N] = sosposop_RL2RL_noR0(prog,[nR nL],X,s,th,[d1 d2]);
[prog, gN] = sosposop_RL2RL_noR0_PS(prog,[nR nL],X,s,th,[d1 d2]);
[prog] = sosopeq(prog,A'*P*T+T'*P*A+N+gN)
prog = sossolve(prog,pars)
```

Stability Conditions:

$\mathcal{P} > 0$ and

$$\mathcal{T}^*\mathcal{P}\mathcal{A} + \mathcal{A}^*\mathcal{P}\mathcal{T} \leq 0$$

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

d	1	2	3	4	limit
$\tau_{\max}$	1.33	1.371	1.3717	1.3718	1.372
CPU sec	2.13	6.29	24.45	79.0	

$$\dot{x}(t) = -\sum_{i=1}^{K} \frac{x(t-i/K)}{K}$$

$K \downarrow n \rightarrow$	1	2	3	5	10
1	.366	.094	.158	.686	12.8
2	.112	.295	1.260	10.83	61.05
3	.177	1.311	6.86	96.85	5223
5	.895	13.05	124.7	2014	200950
10	13.09	59.5	5077	200231	NA

Table: CPU sec indexed by # of states (n) and # of delays (K)

Complexity Scaling Results: Viable when $nK < 50$

The KYP Lemma using 4-PI Operators

$$\mathcal{T}\dot{\mathbf{x}}(t) = \mathcal{A}\mathbf{x}(t) + \mathcal{B}w(t)$$
$$z(t) = \mathcal{C}_1\mathbf{x}(t) + \mathcal{D}_1 w(t), \qquad y(t) = \mathcal{C}_2\mathbf{x}(t) + \mathcal{D}_2 w(t)$$

Theorem 2 (KYP and H_∞-Gain).

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

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

on $\mathbb{R}^{r+p+n} \times L_2^{nK}[-1,0]$, *where* $\mathcal{T}, \mathcal{A}, \mathcal{B}, \mathcal{C}_1, \mathcal{D}_1$ *are as defined previously. Then* $\|z\|_{L_2} \leq \gamma \|\omega\|_{L_2}$.

Proof **Choose Lyapunov function as**

$$V(\mathbf{x}) = \langle \mathcal{T}\mathbf{x}, \mathcal{P}\left\{\begin{smallmatrix} P, & Q_1 \\ Q_2, & \{R_i\}\end{smallmatrix}\right\}\mathcal{T}\mathbf{x}\rangle$$

Then $\dot{V}(\mathbf{x}(t)) - \gamma w^T(t)w(t) - \gamma \upsilon(t)^T\upsilon(t) + \langle z(t), \upsilon(t)\rangle + \langle \upsilon(t), z(t)\rangle < 0$,
where $\upsilon(t) = \frac{1}{\gamma}z(t)$, hence $\|z\|_{L_2} \leq \gamma\|\omega\|_{L_2}$.

Matlab Implementation of the 4-PI KYP Lemma

Almost Complete Matlab Code:

```
pvar s th gam; opvar T A B C1 D1;
A=...;B=...;C1=...;D1=...;T=...;
X=[-tau,0];
prog = sosprogram([s;th],gam)
[prog, P] = sosposopvar(prog,[n n],X,s,th,[d1 d2]);
D=[-gam*eye(nw) D'         B'*P*T;
   D         -gam*eye(ny) C1;
   T'*P*B   C1'           T'*P*A+A'*P*T];
```

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

```
[prog, N] = sosposopvar_noR0(prog,D.dim(:,2),X,s,th,[d1 d2]);
[prog, gN] = sosposopvar_noR0_PS(prog,D.dim(:,2),X,s,th,[d1 d2]);
prog = sosopeq(prog,D+N+gN);
prog = sossetobj(prog, gamma); prog = sossolve(prog);
```

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.

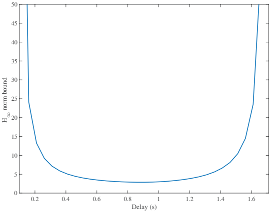


Figure: Calculated H_∞ norm bound vs. delay for Ex. 2

H_∞-Optimal Observer Synthesis

Nominal System using 4-PIE Operators:

$$\mathcal{T}\dot{\mathbf{x}}(t) = \mathcal{A}\mathbf{x}(t) + \mathcal{B}w(t)$$
$$y(t) = \mathcal{C}_2\mathbf{x}(t) + \mathcal{D}_2 w(t), \qquad z(t) = \mathcal{C}_1\mathbf{x}(t) + \mathcal{D}_1 w(t)$$

Structure of the Observer:

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

where the observer gains are

$$\mathcal{L} := \mathcal{P}\left\{\begin{smallmatrix} L_1, & \emptyset \\ L_2, & \{\emptyset\} \end{smallmatrix}\right\}$$

Note: Observer corrects estimate of both current state and history

---

Implementation of the Observer in original states (A PDE!, not a TDS):

$$\begin{bmatrix} \dot{\hat{x}}(t) \\ \dot{\hat{\phi}}_i(t,s) \end{bmatrix} = \begin{bmatrix} A_0\hat{x}(t) + \sum_i A_i\hat{\phi}_i(t,-1) \\ \frac{1}{\tau_i}\hat{\phi}_{i,s}(t,s) \end{bmatrix} + \begin{bmatrix} L_1(\hat{y}(t) - y(t)) \\ L_{2i}(s)(\hat{y}(t) - y(t)) \end{bmatrix}$$

$$\hat{y}(t) = C_{20}\hat{x}(t) + \sum_{i=1}^{K} C_{2i}\hat{\phi}_i(t,-1) \quad \hat{\phi}_i(t,0) = \hat{x}(t)$$

An LOI for H_∞-Optimal Observer Design

Define $e(t) = \hat{\mathbf{x}}(t) - \mathbf{x}(t)$. The closed-loop error dynamics are

$$\mathcal{T}\dot{\mathbf{e}}(t) = (\mathcal{A} + \mathcal{L}\mathcal{C}_2)\mathbf{e}(t) - (\mathcal{B} + \mathcal{L}\mathcal{D}_2)w(t)$$
$$z_e(t) = \mathcal{C}_1\mathbf{e}(t) - \mathcal{D}_1 w(t)$$

Theorem 3.

Suppose there exist operators $\mathcal{P} = \mathcal{P}\left\{\begin{smallmatrix} P, & Q \\ Q^T, & \{R_i\}\end{smallmatrix}\right\} \geq 0 : \mathbb{R}^n \times L_2^n \to \mathbb{R}^n \times L_2^n$ *and* $\mathcal{Z} = \mathcal{P}\left\{\begin{smallmatrix} Z_1, & \emptyset \\ Z_2, & \{\emptyset\}\end{smallmatrix}\right\} : \mathbb{R}^q \to \mathbb{R}^n \times L_2^n$ *such that*

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

on $\mathbb{R}^{r+p+n} \times L_2^n[-\tau, 0]$*, where* $\mathcal{T}, \mathcal{A}, \mathcal{B}, \mathcal{C}_1, \mathcal{D}_1, \mathcal{C}_2, \mathcal{D}_2$ *are as defined previously. Then if* $\mathcal{L} = \mathcal{P}^{-1}Z$*, solutions satisfy* $\|z_e\|_{L_2} \leq \gamma\|\omega\|_{L_2}$.

Structure of $\mathcal{L}$: Inverse of a 4-PI operator is a 4-PI operator (if $R_1 = R_2$)

$$\mathcal{P}\left\{\begin{smallmatrix} P, & Q \\ Q^T, & \{R_i\}\end{smallmatrix}\right\}^{-1} = \mathcal{P}\left\{\begin{smallmatrix} \hat{P}, & \hat{Q} \\ \hat{Q}^T, & \{\hat{R}_i\}\end{smallmatrix}\right\} \quad \Rightarrow \quad \mathcal{L} := \mathcal{P}\left\{\begin{smallmatrix} \hat{P}, & \hat{Q} \\ \hat{Q}^T, & \{\hat{R}_i\}\end{smallmatrix}\right\}\mathcal{P}\left\{\begin{smallmatrix} Z_1, & \emptyset \\ Z_2, & \{\emptyset\}\end{smallmatrix}\right\} = \mathcal{P}\left\{\begin{smallmatrix} L_1, & \emptyset \\ L_2, & \{\emptyset\}\end{smallmatrix}\right\}$$

Proof **Choose Lyapunov function as**

$$V(\mathbf{e}) = \langle \mathcal{T}\mathbf{e}, \mathcal{P}\{{}_{Q^T,\ \{R_i\}}^{P,\ \ Q}\}\mathcal{T}\mathbf{e}\rangle$$

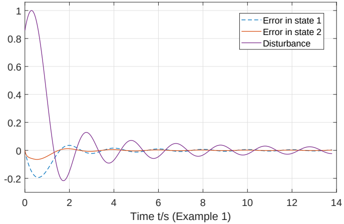
Define $v_e = \frac{1}{\gamma} z_e$. Then

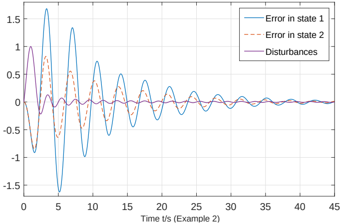
$$\dot{V}(\mathbf{e}) - \gamma w^T w + \frac{1}{\gamma} z_e^T z_e =$$

$$\left\langle \begin{bmatrix} w \\ v_e \\ \mathbf{e}_f \end{bmatrix}, \begin{bmatrix} -\gamma I & -\mathcal{D}_1^* & -(\mathcal{P}\mathcal{B}+\mathcal{Z}\mathcal{D}_2)^*\mathcal{T} \\ -\mathcal{D}_1 & -\gamma I & \mathcal{C}_1 \\ -\mathcal{T}^*(\mathcal{P}\mathcal{B}+\mathcal{Z}\mathcal{D}_2) & \mathcal{C}_1^* & (\mathcal{P}\mathcal{A}+\mathcal{Z}\mathcal{C}_2)^*\mathcal{T}+\mathcal{T}^*(\mathcal{P}\mathcal{A}+\mathcal{Z}\mathcal{C}_2) \end{bmatrix} \begin{bmatrix} w \\ v_e \\ \mathbf{x}_f \end{bmatrix} \right\rangle < 0$$

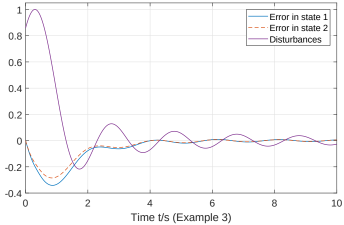
Almost Complete Matlab Code:

```
pvar s th gam; opvar T A B C1 C2 D1 D2;
A=...;B=...;C1=...;C2=...;D1=...;D2=...;T=...;X=[-tau,0];
prog = sosprogram([s;th],gam)
[prog, P] = sos_posopvar(prog,[n n],X,s,th,[d1 d2]);
[prog, Z] = sos_opvar(prog,[n q;n 0],X,s,th,[d1 d2]);
D=[-gam*eye(nw)      D'              -(P*B+Z*D2)'*T;
   D               -gam*eye(ny)      C1;
   -T'*(P*B+Z*D2)    C1'             T'*(P*A+Z*C2)+(P*A+Z*C2)*T];
[prog, N] = sosposop_RL2RL_noR0(prog,D.dim(:,2),X,s,th,[d1 d2]);
[prog, gN] = sosposop_RL2RL_noR0_PS(prog,D.dim(:,2),X,s,th,[d1 d2]);
prog = sosopeq(prog,D+N+gN);prog = sossetobj(prog, gamma); prog =
sossolve(prog);
```





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		



The Last Slide (Thanks to NSF CNS-1739990)

$\mathcal{P}\left\{\begin{smallmatrix} P, & Q_1 \\ Q_2, & \{R\} \end{smallmatrix}\right\}$ Operators extend LMI techniques to PDEs and Delay Systems.

- $A^T P + PA < 0$ becomes

$$\mathcal{A}^*\mathcal{P}\mathcal{T} + \mathcal{T}^*\mathcal{P}\mathcal{A} \le 0$$

Conclusions:

PROs:

- Computationally Efficient
- A more rational treatment of boundary conditions.
- No Conservatism (Almost N+S)
- Easily Extended to New Problems
  - e.g. Input Delay
  - e.g. Sampled Data Systems
- A very Nice Parser

CONs:

- Operator Theory
- Descriptor Systems

Extensions:

- IQCs for Nonlinearity
  - H_2 Gain

Solvable (in order of difficulty)

- Optimal Dynamic Output Feedback
- Inversion of the $\mathcal{P}\left\{\begin{smallmatrix} P, & Q_1 \\ Q_2, & \{R\} \end{smallmatrix}\right\}$ Operator
  - When $R_1 \neq R_2$

The VERY Last Slide

Everything Here is a TOOL!

Good Luck
Be Productive

With Luck, you won't need luck