IDeA Labs
Information & Decision Algorithms Laboratories
Brigham Young University

www.idealabs.byu.edu

Business

Life Sciences

Engineering

Organizations

Passive Reconstruction of Non-Target-Specific Discrete-Time LTI Systems

Vasu Chetty
Brigham Young University

American Control Conference
Boston, MA
July 6, 2016

Outline

- System Representations
- Network Reconstruction
  - Target Specificity
  - Active vs. Passive Reconstruction
- Main Result
  - Dynamical Structure Function Representations
  - Reconstruction Process
  - Numerical Example
- Conclusions and Future Work

Outline

- System Representations
- Network Reconstruction
  - Target Specificity
  - Active vs. Passive Reconstruction
- Main Result
  - Dynamical Structure Function Representations
  - Reconstruction Process
  - Numerical Example
- Conclusions and Future Work

System Representations

- There are many ways to represent a system

System Representations

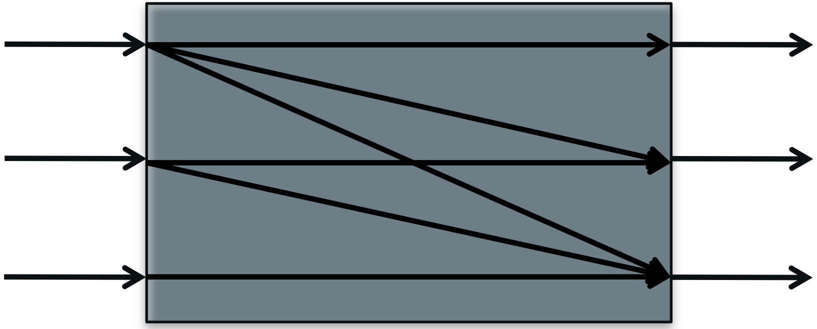
- There are many ways to represent a system
- Different representations of the same system detail different notions of structure

System Representations

- There are many ways to represent a system
- Different representations of the same system detail different notions of structure
- Three common system representations:
  - Transfer function
  - State space model
  - Dynamical structure function (linear dynamical graphs)

System Representations

- Transfer function:

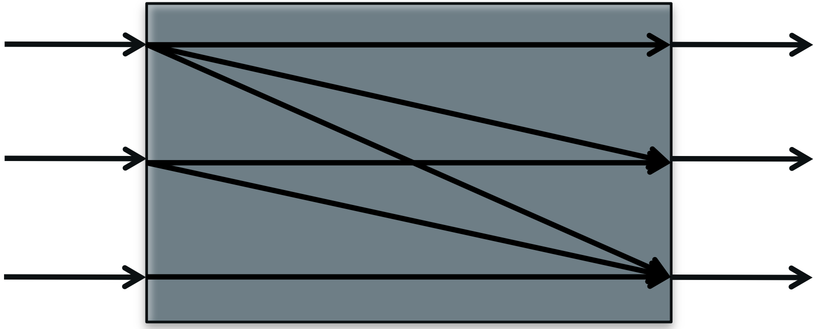


Closed-loop paths

$$\begin{bmatrix} Y_1(z) \\ Y_2(z) \\ Y_3(z) \end{bmatrix} = \begin{bmatrix} \frac{2}{2z+1} & 0 & 0 \\ \frac{-2}{2z^2+z} & \frac{1}{z} & 0 \\ \frac{4}{4z^3-z} & \frac{-2}{2z^2-z} & \frac{2}{2z-1} \end{bmatrix} \begin{bmatrix} U_1(z) \\ U_2(z) \\ U_3(z) \end{bmatrix}$$

System Representations

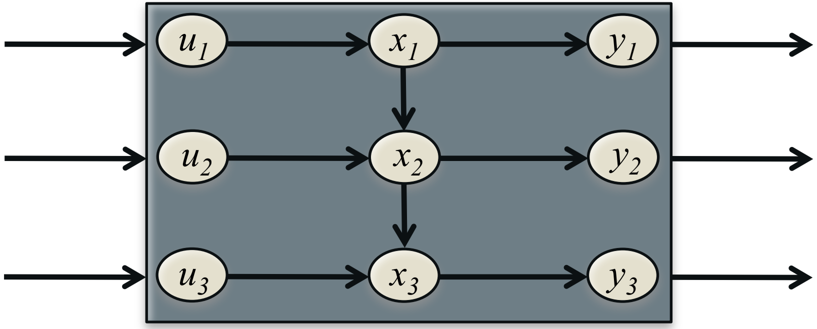
- Transfer function:



Closed-loop paths

$$\begin{bmatrix} Y_1(z) \\ Y_2(z) \\ Y_3(z) \end{bmatrix} = \begin{bmatrix} \frac{2}{2z+1} & 0 & 0 \\ \frac{-2}{2z^2+z} & \frac{1}{z} & 0 \\ \frac{4}{4z^3-z} & \frac{-2}{2z^2-z} & \frac{2}{2z-1} \end{bmatrix} \begin{bmatrix} U_1(z) \\ U_2(z) \\ U_3(z) \end{bmatrix}$$

- State space model:

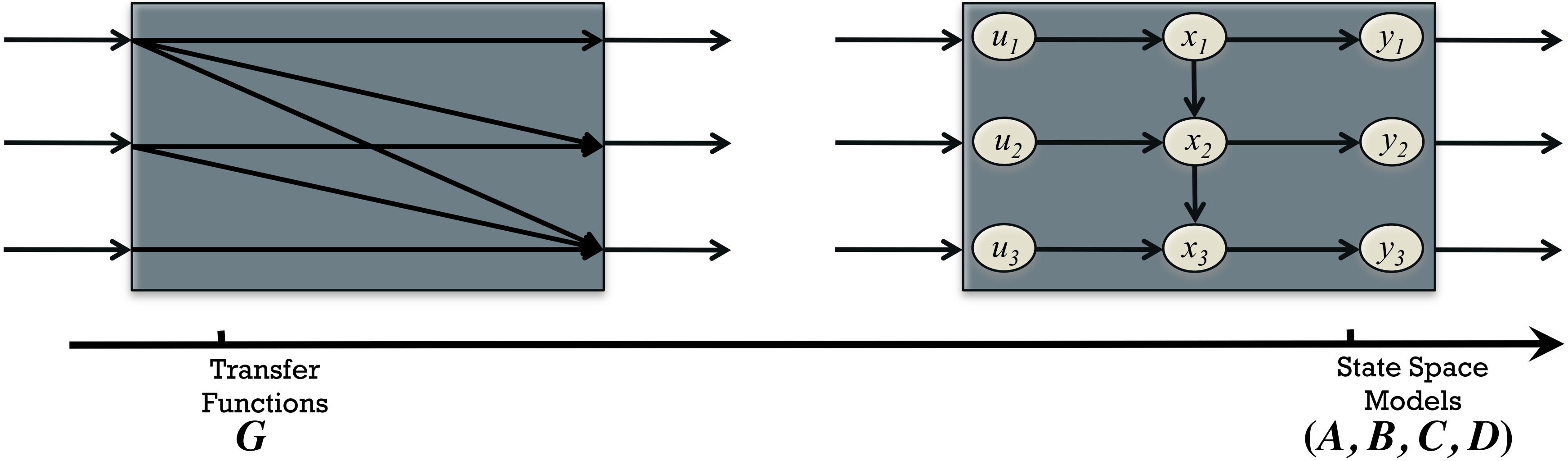


Computational Structure

$$\begin{bmatrix} x_1[k+1] \\ x_2[k+1] \\ x_3[k+1] \end{bmatrix} = \begin{bmatrix} -.5 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & .5 \end{bmatrix} \begin{bmatrix} x_1[k] \\ x_2[k] \\ x_3[k] \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1[k] \\ u_2[k] \\ u_3[k] \end{bmatrix}$$

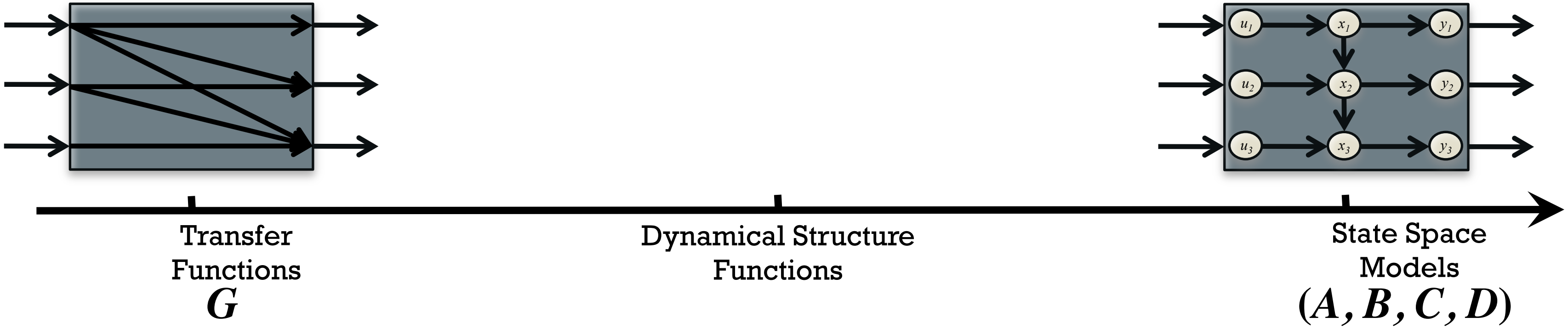
$$\begin{bmatrix} y_1[k] \\ y_2[k] \\ y_3[k] \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1[k] \\ x_2[k] \\ x_3[k] \end{bmatrix}$$

System Representations



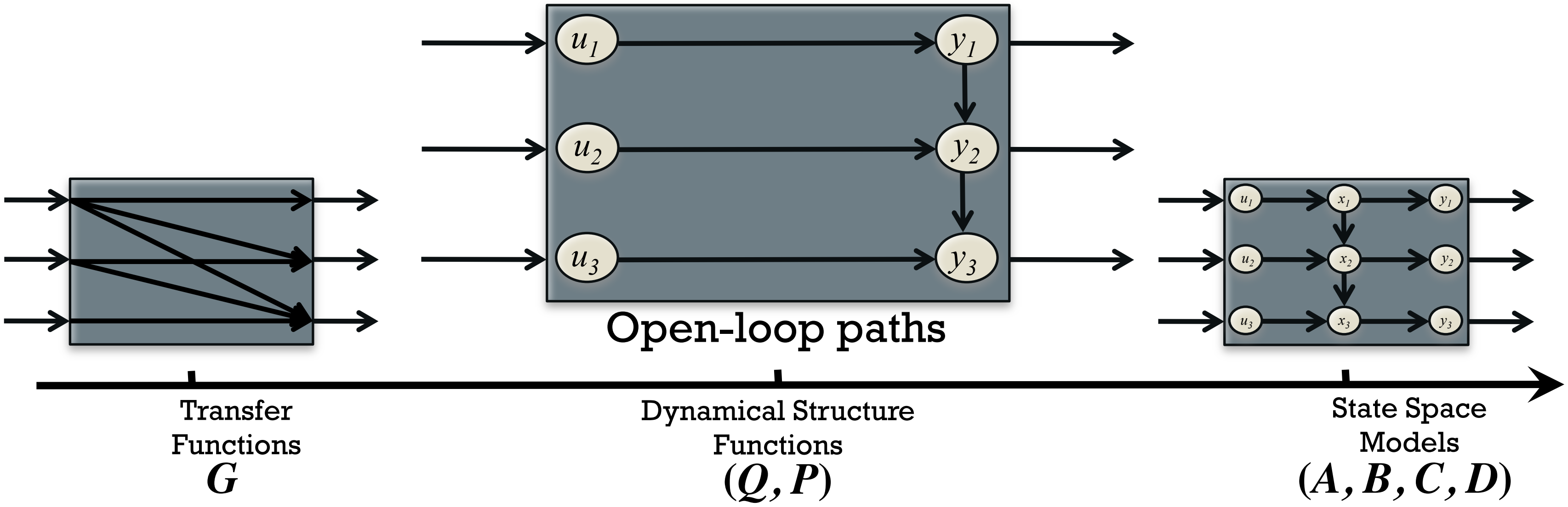
Structural Informativity

System Representations



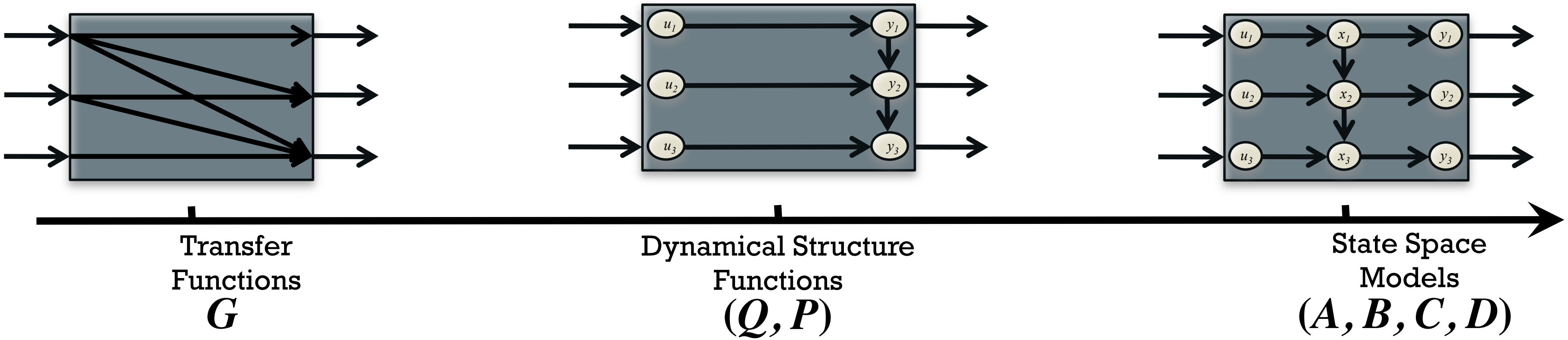
Structural Informativity

System Representations



$$\begin{bmatrix} Y_1(z) \\ Y_2(z) \\ Y_3(z) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ \frac{-1}{z} & 0 & 0 \\ 0 & \frac{-2}{2z-1} & 0 \end{bmatrix} \begin{bmatrix} Y_1(z) \\ Y_2(z) \\ Y_3(z) \end{bmatrix} + \begin{bmatrix} \frac{2}{2z+1} & 0 & 0 \\ 0 & \frac{1}{z} & 0 \\ 0 & 0 & \frac{2}{2z-1} \end{bmatrix} \begin{bmatrix} U_1(z) \\ U_2(z) \\ U_3(z) \end{bmatrix}$$

System Representations

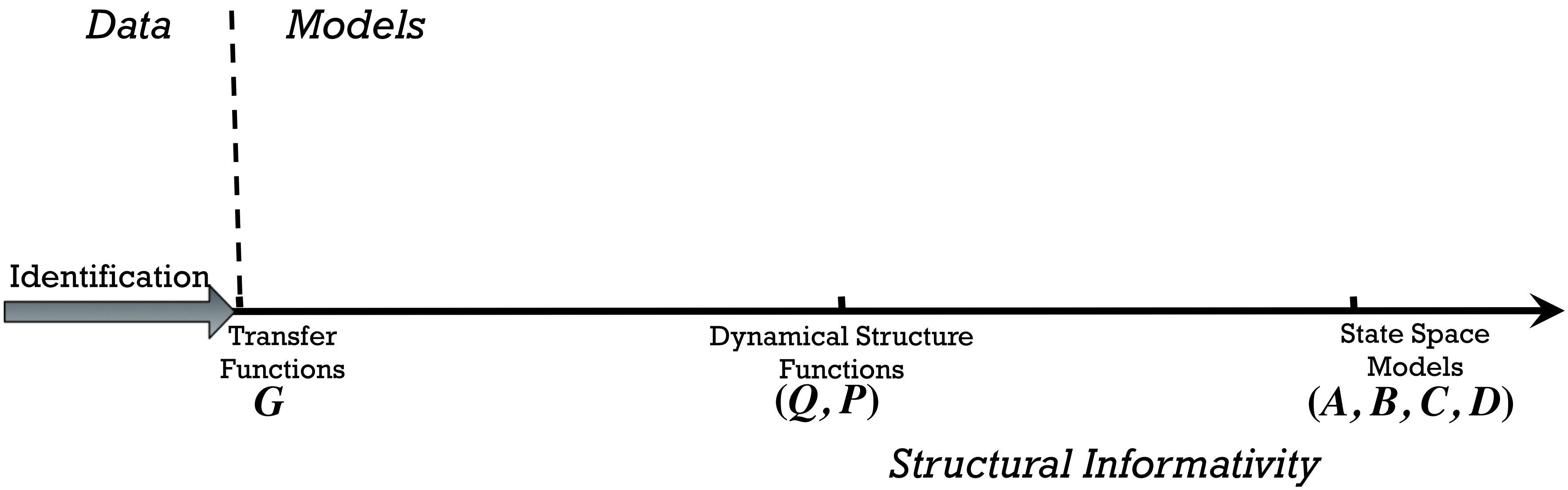


Structural Informativity

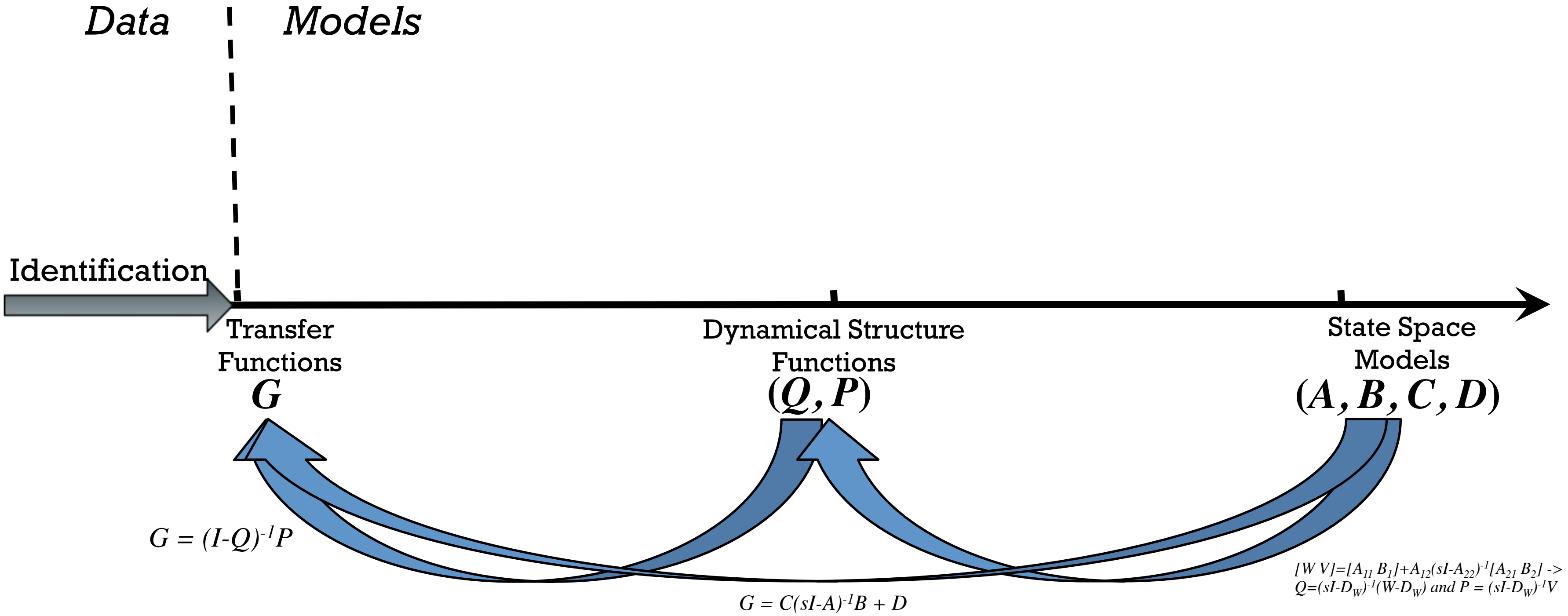
Outline

- System Representations
- Network Reconstruction
  - Target Specificity
  - Active vs. Passive Reconstruction
- Main Result
  - Dynamical Structure Function Representations
  - Reconstruction Process
  - Numerical Example
- Conclusions and Future Work

System Identification

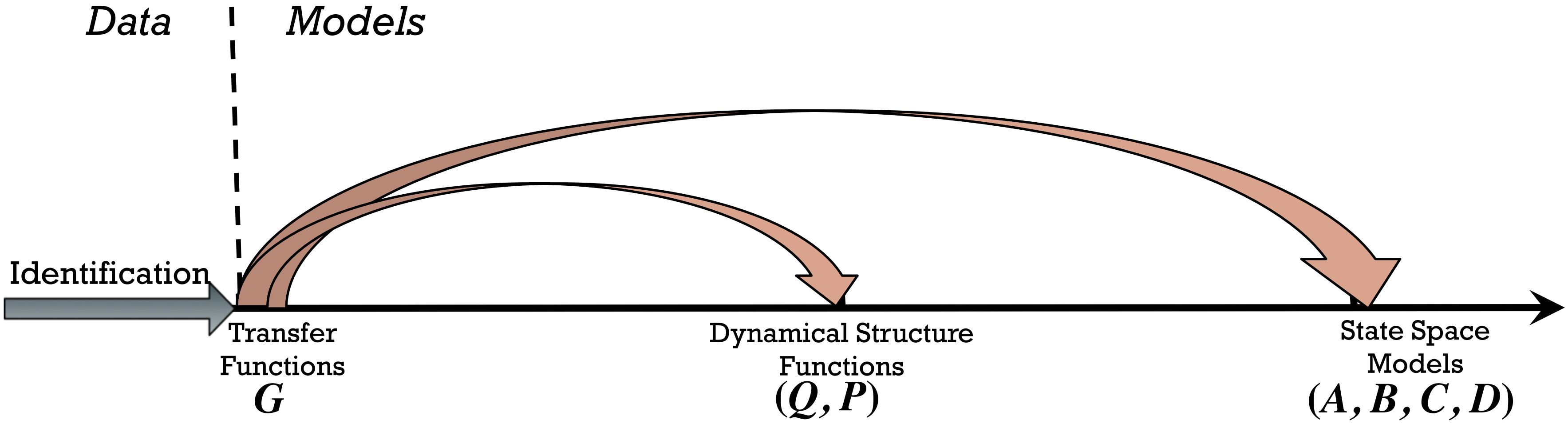


Derivations of Less Informative Models



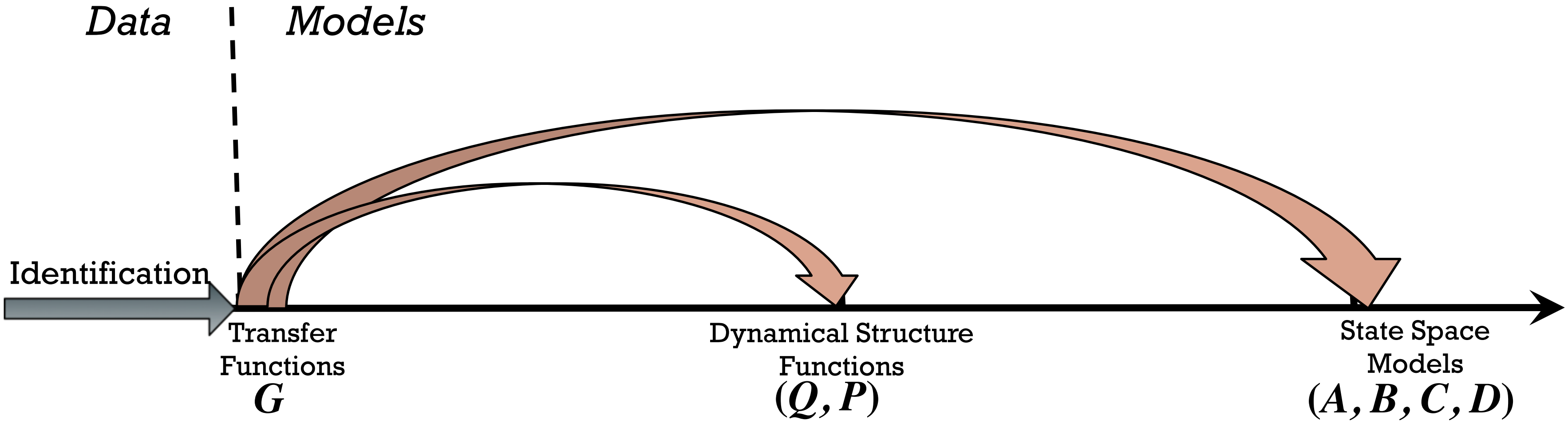
- Well-posed problems

Network Reconstruction



- NOT well-posed problems

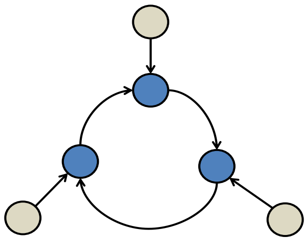
Network Reconstruction



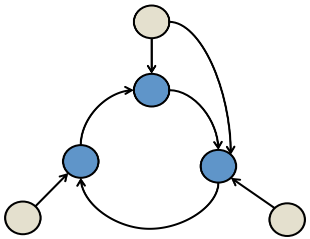
- NOT well-posed problems
- Extra information beyond input-output data is required

Target Specificity

- Target specificity refers to a property of the system in which:
  - Each input affects other measured outputs only through a particular measured output associated with that input, and
  - Every measured output is associated with a distinct input which affects it first



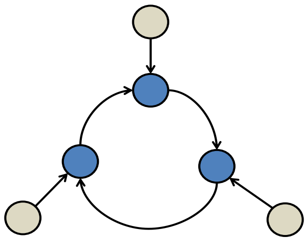
Target-specific



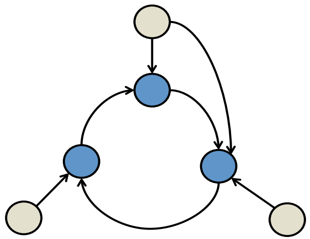
Non-target-specific

Target Specificity

- Target specificity refers to a property of the system in which:
  - Each input affects other measured outputs only through a particular measured output associated with that input, and
  - Every measured output is associated with a distinct input which affects it first
- Many reconstruction algorithms assume target specificity, since it is sufficient for reconstruction



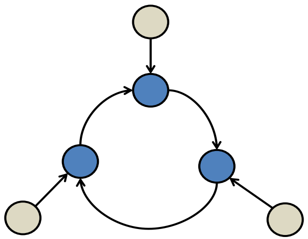
Target-specific



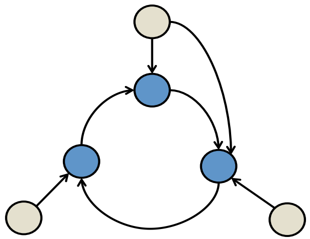
Non-target-specific

Target Specificity

- Target specificity refers to a property of the system in which:
  - Each input affects other measured outputs only through a particular measured output associated with that input, and
  - Every measured output is associated with a distinct input which affects it first
- Many reconstruction algorithms assume target specificity, since it is sufficient for reconstruction
- Target specificity is not necessary for network reconstruction



Target-specific



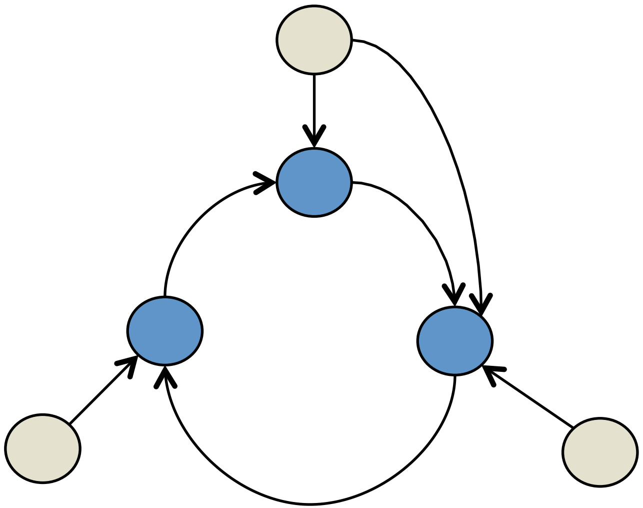
Non-target-specific

Active Reconstruction

- Network reconstruction where:
  - Experiments performed by individually perturbing each input

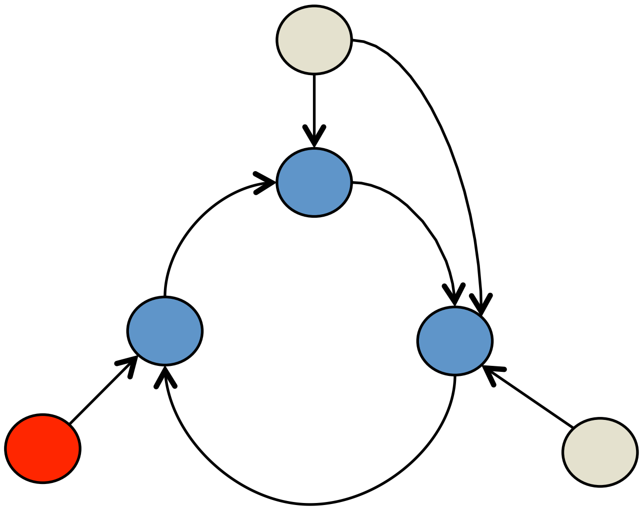
+ Active Reconstruction

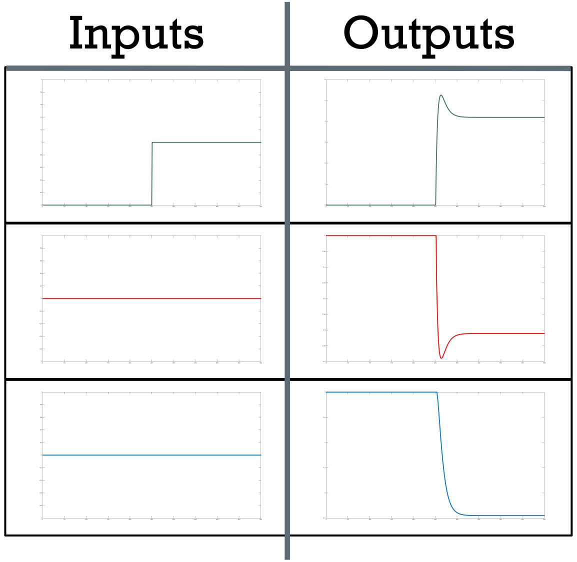
- Network reconstruction where:
  - Experiments performed by individually perturbing each input



Active Reconstruction

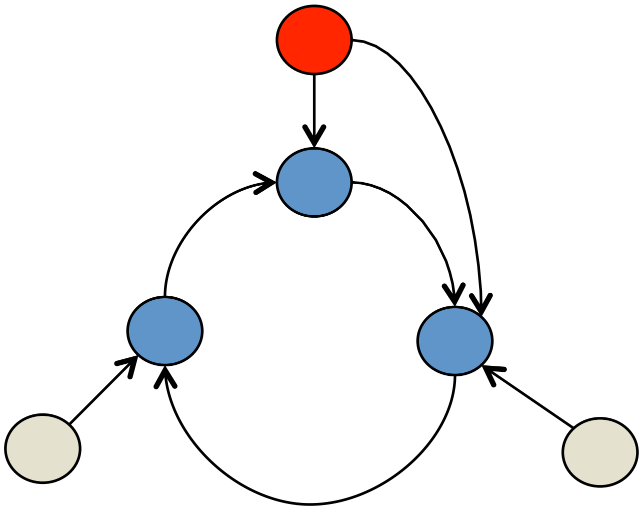
- Network reconstruction where:
  - Experiments performed by individually perturbing each input

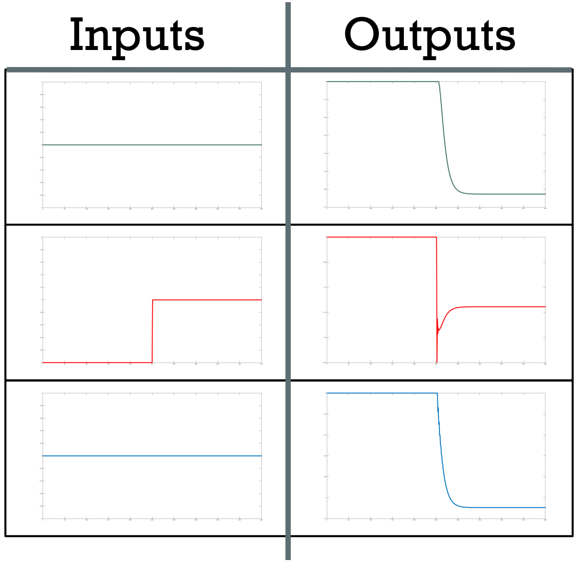




Active Reconstruction

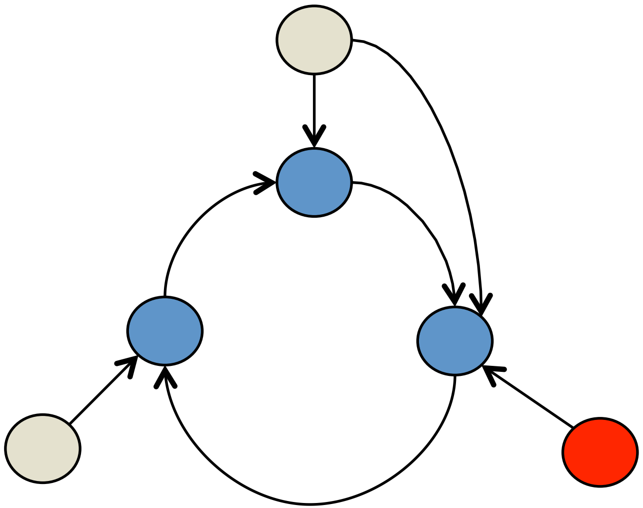
- Network reconstruction where:
  - Experiments performed by individually perturbing each input

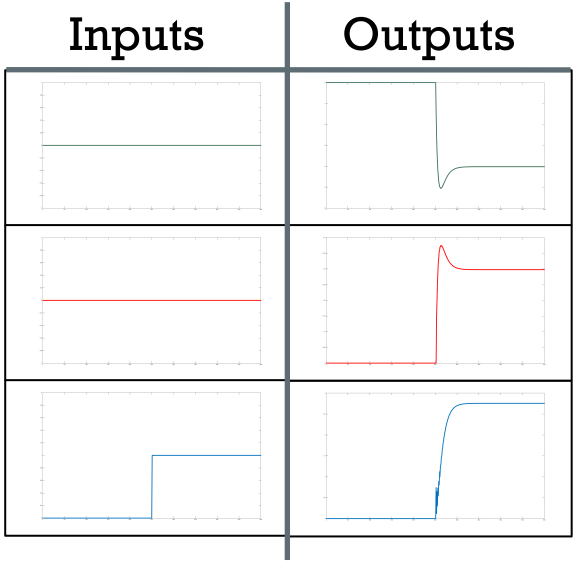




+ Active Reconstruction

- Network reconstruction where:
  - Experiments performed by individually perturbing each input



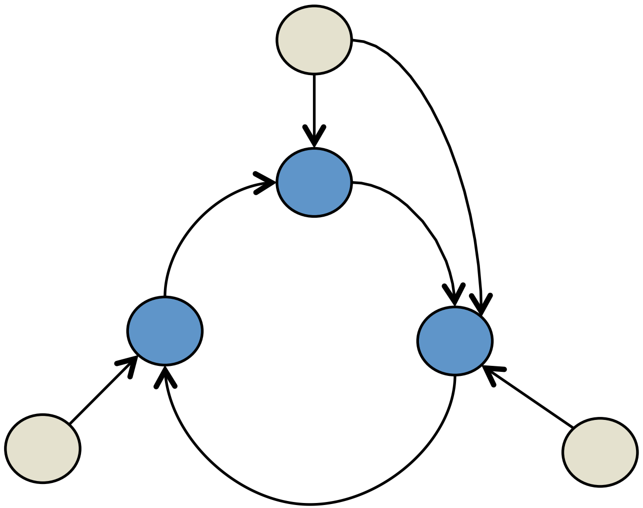


Passive Reconstruction

- Network reconstruction where:
  - All inputs perturbed simultaneously, usually by noise

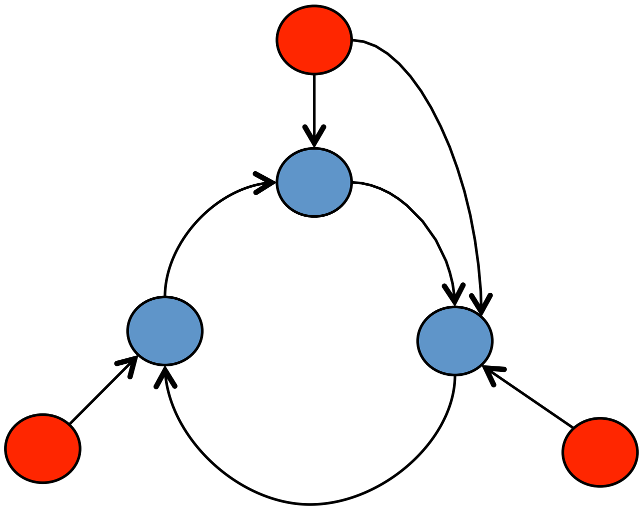
Passive Reconstruction

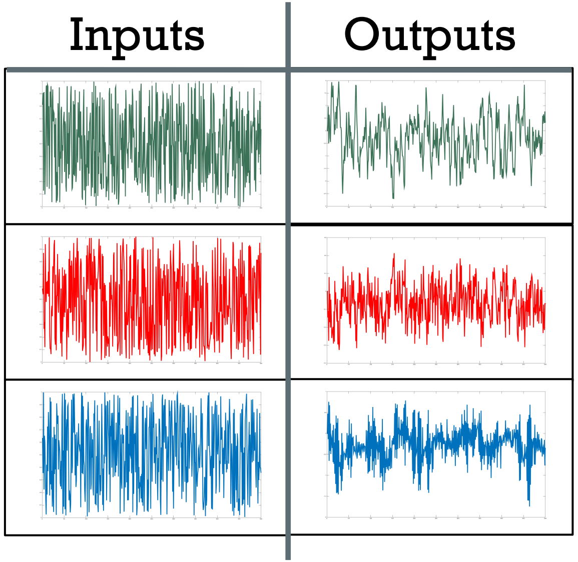
- Network reconstruction where:
  - All inputs perturbed simultaneously, usually by noise



Passive Reconstruction

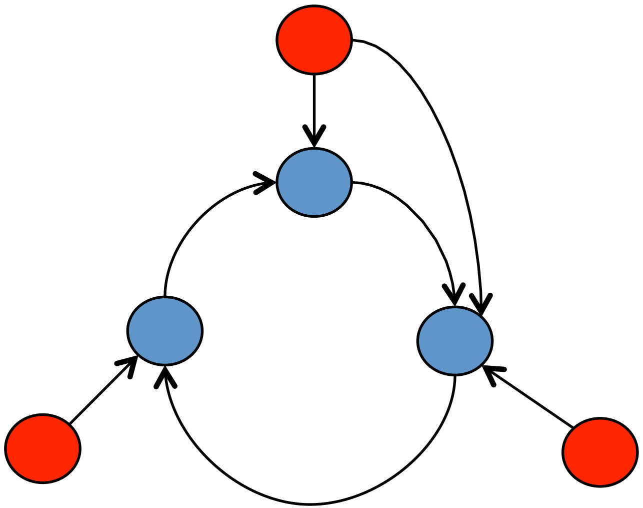
- Network reconstruction where:
  - All inputs perturbed simultaneously, usually by noise

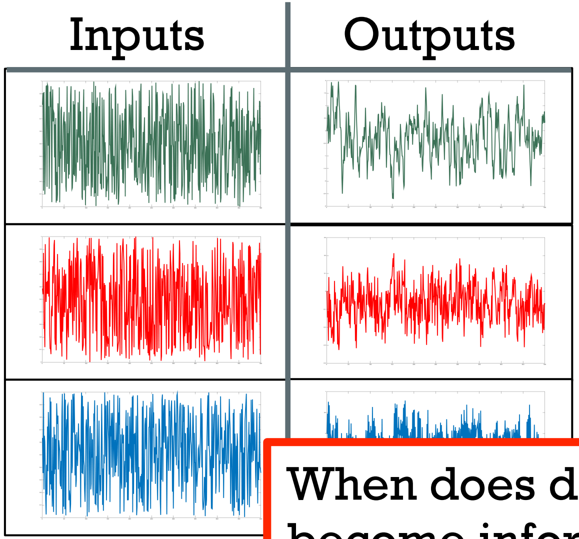




Passive Reconstruction

- Network reconstruction where:
  - All inputs perturbed simultaneously, usually by noise





When does data become informative?

Outline

- System Representations
- Network Reconstruction
  - Target Specificity
  - Active vs. Passive Reconstruction
- Main Result
  - Dynamical Structure Function Representations
  - Reconstruction Process
  - Numerical Example
- Conclusions and Future Work

Main Result

- Overview:
  - Time domain representation
  - Necessary and sufficient conditions in time domain
  - Network reconstruction in time domain

+ Frequency Domain Representation

- Dynamical structure function defined in the frequency domain

$$Y(z) = Q(z)Y(z) + P(z)U(z)$$

Time Domain: Convolution Representation

- Take the inverse Z-transform of

$$Y(z) = Q(z)Y(z) + P(z)U(z)$$

to get the convolution representation

$$y[k] = Q[k] * y[k] + P[k] * u[k]$$

which can be rewritten as

$$\overline{y}_r = \overline{Q}_r \overline{y}_r + \overline{P}_r \overline{u}_r$$

Time Domain: Toeplitz Representation

$$\bar{y}_r = \bar{Q}_r \bar{y}_r + \bar{P}_r \bar{u}_r$$

where

$$\bar{Q}_r = \begin{bmatrix} 0 & \cdots & \cdots & \cdots \\ Q[1] & \ddots & & \\ Q[2] & Q[1] & \ddots & \\ \vdots & \ddots & \ddots & \ddots \\ Q[r] & & \ddots & \ddots \end{bmatrix}$$

$$\bar{P}_r = \begin{bmatrix} 0 & \cdots & \cdots & \cdots \\ P[1] & \ddots & & \\ P[2] & P[1] & \ddots & \\ \vdots & \ddots & \ddots & \ddots \\ P[r] & & \ddots & \ddots \end{bmatrix}$$

$$\bar{y}_r = \begin{bmatrix} y[1]^T & y[2]^T & \ldots & y[r]^T \end{bmatrix}^T$$

$$\bar{u}_r = \begin{bmatrix} u[1]^T & u[2]^T & \ldots & u[r]^T \end{bmatrix}^T$$

Time Domain: Toeplitz Representation

$$\bar{y}_r = \bar{Q}_r \bar{y}_r + \bar{P}_r \bar{u}_r$$

where

$$\bar{Q}_r = \begin{bmatrix} 0 & \cdots & \cdots & \cdots \\ Q[1] & \ddots & & \\ Q[2] & Q[1] & \ddots & \\ \vdots & \ddots & \ddots & \ddots \\ Q[r] & & \ddots & \ddots \end{bmatrix} \qquad \bar{P}_r = \begin{bmatrix} 0 & \cdots & \cdots & \cdots \\ P[1] & \ddots & & \\ P[2] & P[1] & \ddots & \\ \vdots & \ddots & \ddots & \ddots \\ P[r] & & \ddots & \ddots \end{bmatrix}$$

$$\bar{y}_r = \begin{bmatrix} y[1]^T & y[2]^T & \ldots & y[r]^T \end{bmatrix}^T$$

$$\bar{u}_r = \begin{bmatrix} u[1]^T & u[2]^T & \ldots & u[r]^T \end{bmatrix}^T$$

Stability: $r \to \infty \Rightarrow Q[r] \to 0 \wedge P[r] \to 0$

Reconstruction Process

- Learn the Toeplitz Representation
- Calculate the Convolution Representation
- Determine the Frequency Domain Dynamical Structure Function

+ Learning the Toeplitz Representation

$$\bar{y}_r = \bar{Q}_r \bar{y}_r + \bar{P}_r \bar{u}_r$$

Learning the Toeplitz Representation

$$\bar{y}_r = \bar{Q}_r \bar{y}_r + \bar{P}_r \bar{u}_r$$

$$\bar{y}_r = \begin{bmatrix} \bar{Q}_r & \bar{P}_r \end{bmatrix} \begin{bmatrix} \bar{y}_r \\ \bar{u}_r \end{bmatrix}$$

+ Learning the Toeplitz Representation

$$\bar{y}_r = \bar{Q}_r \bar{y}_r + \bar{P}_r \bar{u}_r$$

$$\bar{y}_r = \begin{bmatrix} \bar{Q}_r & \bar{P}_r \end{bmatrix} \begin{bmatrix} \bar{y}_r \\ \bar{u}_r \end{bmatrix}$$

$$\bar{y}_r^T = \begin{bmatrix} \bar{y}_r^T & \bar{u}_r^T \end{bmatrix} \begin{bmatrix} \bar{Q}_r^T \\ \bar{P}_r^T \end{bmatrix}$$

Learning the Toeplitz Representation

$$\bar{y}_r = \bar{Q}_r \bar{y}_r + \bar{P}_r \bar{u}_r$$

$$\bar{y}_r = \begin{bmatrix} \bar{Q}_r & \bar{P}_r \end{bmatrix} \begin{bmatrix} \bar{y}_r \\ \bar{u}_r \end{bmatrix}$$

$$\bar{y}_r^T = \begin{bmatrix} \bar{y}_r^T & \bar{u}_r^T \end{bmatrix} \begin{bmatrix} \bar{Q}_r^T \\ \bar{P}_r^T \end{bmatrix}$$

$$\begin{bmatrix} y_1^T & \dots & y_r^T \end{bmatrix} = \begin{bmatrix} y_1^T & \dots & y_r^T & u_1^T & \dots & u_r^T \end{bmatrix} \begin{bmatrix} 0 & Q[1]^T & \dots & Q[r]^T \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & Q[1]^T \\ 0 & P[1]^T & \dots & P[r]^T \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & P[1]^T \end{bmatrix}$$

+ Learning the Toeplitz Representation

$$\hat{y} = \hat{L}\vec{\hat{x}}$$

Learning the Toeplitz Representation

$$\hat{y} = \hat{L}\vec{\hat{x}}$$

$$\hat{y} = \begin{bmatrix} y_1^T & \ldots & y_1^T & y_f^T & \ldots & y_f^T \end{bmatrix}^T$$

$$\hat{x} = \begin{bmatrix} Q[1]^T & \ldots & Q[r]^T & P[1]^T & \ldots & P[r]^T \end{bmatrix}$$

$$\hat{L} = \begin{bmatrix} y_1^T & 0 & 0 & \ldots & 0 & 0 & 0 & u_1^T & 0 & 0 & \ldots & 0 & 0 & 0 \\ 0 & \ddots & 0 & \ldots & 0 & \ddots & 0 & 0 & \ddots & 0 & \ldots & 0 & \ddots & 0 \\ 0 & 0 & y_1^T & \ldots & 0 & 0 & 0 & 0 & 0 & u_1^T & \ldots & 0 & 0 & 0 \\ & \vdots & & & & \vdots & & & \vdots & & & & \vdots & \\ y_f^T & 0 & 0 & \ldots & y_1^T & 0 & 0 & u_1^T & 0 & 0 & \ldots & u_1^T & 0 & 0 \\ 0 & \ddots & 0 & \ldots & 0 & \ddots & 0 & 0 & \ddots & 0 & \ldots & 0 & \ddots & 0 \\ 0 & 0 & y_f^T & \ldots & 0 & 0 & y_1^T & 0 & 0 & u_1^T & \ldots & 0 & 0 & u_1^T \end{bmatrix}$$

Learning the Toeplitz Representation

$$\hat{y} = \hat{L}\vec{\hat{x}}$$

$$\hat{y} = \begin{bmatrix} y_1^T & \dots & y_1^T & y_f^T & \dots & y_f^T \end{bmatrix}^T$$

$$\hat{x} = \begin{bmatrix} Q[1]^T & \dots & Q[r]^T & P[1]^T & \dots & P[r]^T \end{bmatrix}$$

$$\hat{L} = \begin{bmatrix} y_1^T & 0 & 0 & \dots & 0 & 0 & 0 & u_1^T & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & \ddots & 0 & \dots & 0 & \ddots & 0 & 0 & \ddots & 0 & \dots & 0 & \ddots & 0 \\ 0 & 0 & y_1^T & \dots & 0 & 0 & 0 & 0 & 0 & u_1^T & \dots & 0 & 0 & 0 \\ & \vdots & & & & \vdots & & & \vdots & & & & \vdots & \\ y_f^T & 0 & 0 & \dots & y_1^T & 0 & 0 & u_1^T & 0 & 0 & \dots & u_1^T & 0 & 0 \\ 0 & \ddots & 0 & \dots & 0 & \ddots & 0 & 0 & \ddots & 0 & \dots & 0 & \ddots & 0 \\ 0 & 0 & y_f^T & \dots & 0 & 0 & y_1^T & 0 & 0 & u_1^T & \dots & 0 & 0 & u_1^T \end{bmatrix}$$

$$S = \begin{bmatrix} s_{11} & \dots & s_{1n} \\ \vdots & \ddots & \vdots \\ s_{n1} & \dots & s_{nn} \end{bmatrix}$$

$$\vec{S} = \begin{bmatrix} s_{11} \\ \vdots \\ s_{n1} \\ \vdots \\ s_{1n} \\ \vdots \\ s_{nn} \end{bmatrix}$$

Learning the Toeplitz Representation

$$\hat{y} = \hat{L}\vec{\hat{x}}$$

Does not have full column rank, requires a priori information to ensure a unique solution

Learning the Toeplitz Representation

- Create $\bar{T} = \begin{bmatrix} T_1 & T_2 \\ T_3 & T_4 \end{bmatrix}$

+ Learning the Toeplitz Representation

- Create $\bar{T} = \begin{bmatrix} T_1 & T_2 \\ T_3 & T_4 \end{bmatrix}$

- where
  - T_1 is a priori information about how to reduce $Q(z)$ given $Q(z)$
  - T_2 is a priori information about how to reduce $Q(z)$ given $P(z)$
  - T_3 is a priori information about how to reduce $P(z)$ given $Q(z)$
  - T_4 is a priori information about how to reduce $P(z)$ given $P(z)$

Learning the Toeplitz Representation

- Create $\bar{T} = \begin{bmatrix} T_1 & T_2 \\ T_3 & T_4 \end{bmatrix}$

- where
  - T_1 is a priori information about how to reduce $Q(z)$ given $Q(z)$

$$x = T_1\alpha \Rightarrow \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = T_1 q_1 \Rightarrow T_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Learning the Toeplitz Representation

- Create $\bar{T} = \begin{bmatrix} T_1 & T_2 \\ T_3 & T_4 \end{bmatrix}$

- Vectorizing the unknowns requires

$$\hat{T} = \begin{bmatrix} T_1 & \dots & 0 & T_2 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & T_1 & 0 & \dots & T_2 \\ T_3 & \dots & 0 & T_4 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & T_3 & 0 & \dots & T_4 \end{bmatrix}$$

+ Learning the Toeplitz Representation

$$\vec{\hat{x}} = \hat{T}\hat{\alpha}$$

$$\hat{y} = \hat{L}\vec{\hat{x}}$$

$$\hat{y} = \hat{L}\hat{T}\hat{\alpha}$$

$$\hat{y} = \hat{M}\hat{\alpha}$$

Learning the Toeplitz Representation

- Theorem 1: Given a system characterized by the transfer function $G(z)$, its dynamical structure function $(Q(z), P(z))$ can be identified in the time domain if and only if
  - $\hat{M} = \hat{L}\hat{T}$ is injective, i.e. $\hat{y} \in \Re(\hat{M})$ and $rank(\hat{M}) = kr$
  - r chosen sufficiently large.

Calculating the Convolution Representation

- Theorem 2: … the entries of [the convolution representation of the dynamical structure function ($Q(z)$, $P(z)$)] have the form:

$$a\delta_{t,0} + \sum_{i=0}^{w} b_i (c_i)^t$$

where w is the number of delays in the corresponding link.

+ Determining the Dynamical Structure Function

- Given the convolution representation of the system, we can take its Z-transform to get the dynamical structure function

Numerical Example

- Consider the following state space system:

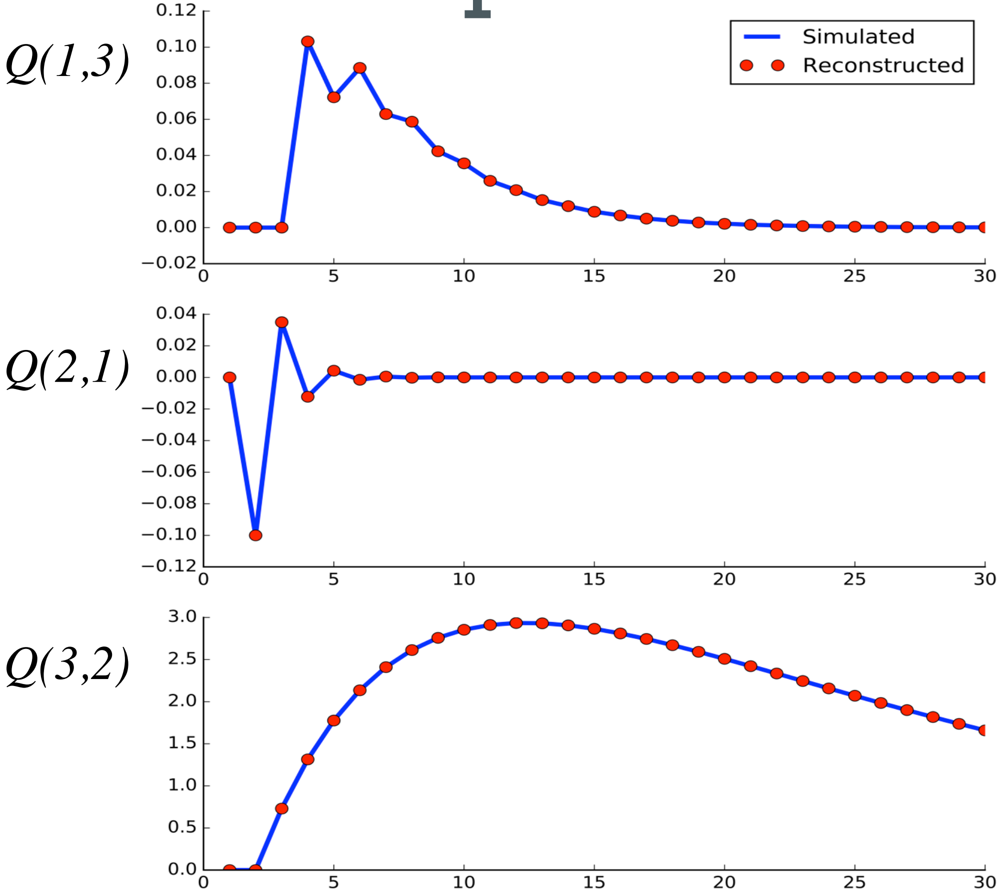
$$x[k+1] = \begin{bmatrix} .75 & 0 & 0 & 0 & 0 & 1.2 \\ -.1 & -.35 & 0 & 0 & 0 & 0 \\ 0 & 0 & .85 & -1 & 0 & 0 \\ 0 & -.73 & 0 & .95 & 0 & 0 \\ 0 & 0 & .43 & 0 & -.6 & 0 \\ 0 & 0 & 0 & 0 & .2 & .55 \end{bmatrix} x[k] + \begin{bmatrix} 1.4 & 0 & -1.4 \\ 0 & -.25 & 0 \\ 0 & 0 & .75 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} u[k]$$

$$y[k] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} x[k]$$

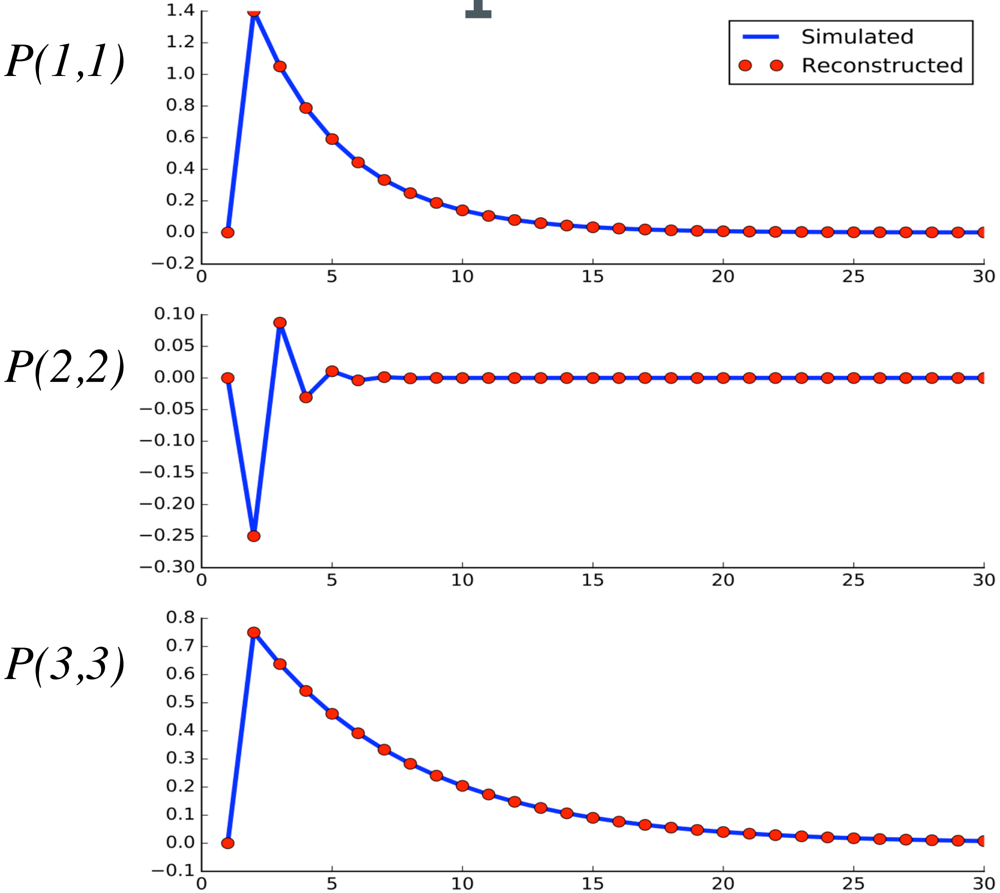
Numerical Example

- Simulate with $r = 600$
- Solve for $\bar{Q}_r$ and $\bar{P}_r$
- Assume structure of $P(z)$ known a priori

Numerical Example



+ Numerical Example



Numerical Example

- Given the Toeplitz representation ($\bar{Q}_r$, $\bar{P}_r$), find convolution representation using a custom curve fit in MATLAB with:

$$a_k \delta_{t,0} + \sum_{i=0}^{w_k} b_i (c_i)^t$$

- This yields:

$$Q(k)[1,3] = .5096(.75)^k - .1108(-.6)^k - .8158(.55)^k + .417\delta$$

$$Q(k)[2,1] = .2839(-.3556)^k - .2839\delta$$

$$Q(k)[3,2] = 7.684(.95)^k - 8.588(.85)^k + .904\delta$$

$$P(k)[1,1] = 1.867(.75)^k - 1.867\delta$$

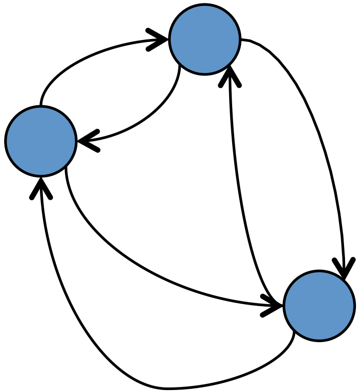
$$P(k)[1,3] = -P(k)[1,1]$$

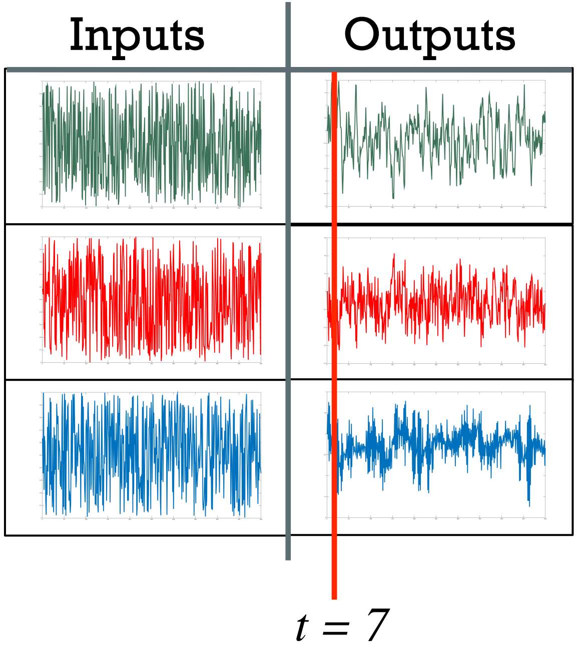
$$P(k)[2,2] = .7143(-.35)^k - .7143\delta$$

$$P(k)[3,3] = .8824(.85)^k - .8824\delta$$

Numerical Example

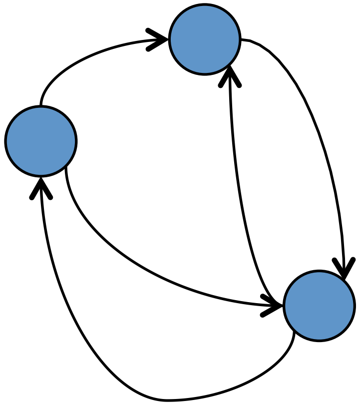
$$Q(z)=\begin{bmatrix} 0 & \dfrac{-.46z-.05}{z^2+0.23z+.014} & \dfrac{-.14z^2+.03z}{z^2+.01z-1.01} \\ \dfrac{-7.95z^2-.74z-1.41}{z^2+.84z+.17} & 0 & -.67 \\ \dfrac{.02z^2+1.67z+1.26}{z^2-.19z-.55} & \dfrac{15983.5z^2-14616.1z-2752}{z^2-.9z-.17} & 0 \end{bmatrix}$$

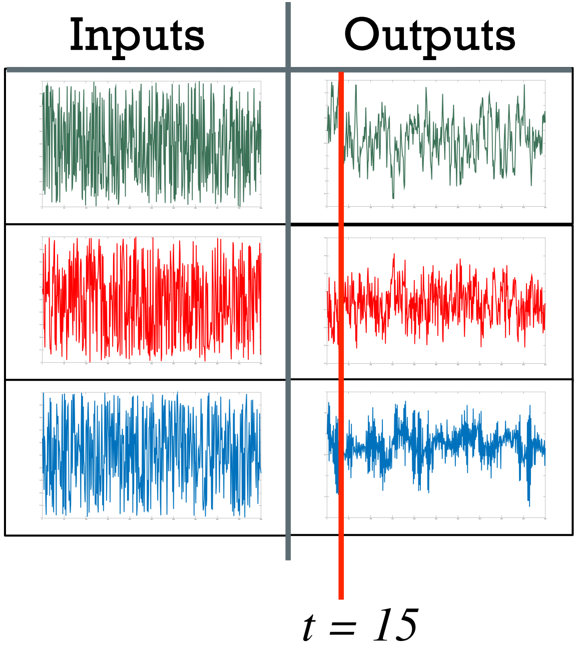




Numerical Example

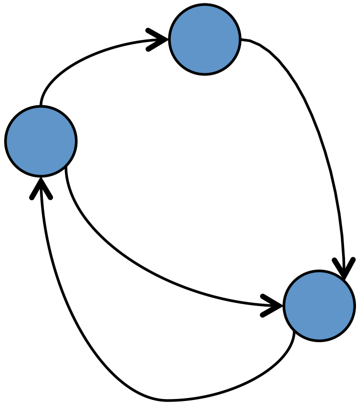
$$Q(z) = \begin{bmatrix} 0 & 0 & \frac{.35z^2 - .39z - .1}{z^3 - .675z^2 - .4z + .265} \\ \frac{.01z - .01}{z + .35} & 0 & -.67 \\ \frac{.19z^2 + .25z + .22}{z^2 - .08z - .75} & \frac{0.45z^3 + 3.97z^2 + 4.2z}{z^3 - .87z^2 - .67z + .59} & 0 \end{bmatrix}$$

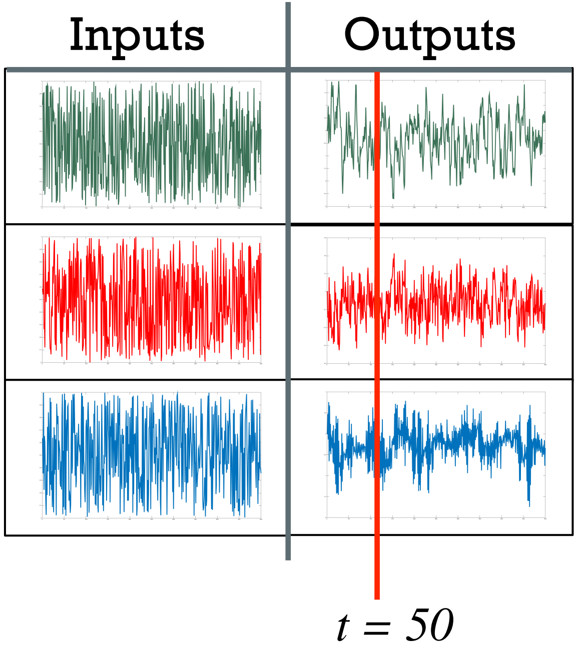




Numerical Example

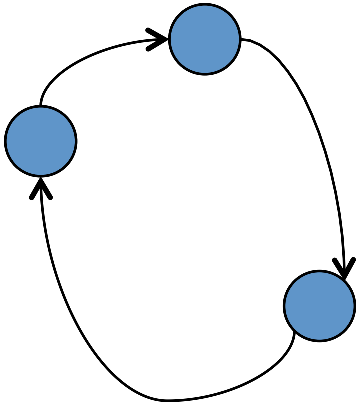
$$Q(z) = \begin{bmatrix} 0 & 0 & \frac{.48z^2 - .08z - .1}{z^3 - 1.9z^2 + 1.2z - .25} \\ \frac{-.01z - .1}{z + .35} & 0 & 0 \\ \frac{.12}{z - .85} & \frac{.02z + .72}{z^2 - 1.8z + .8} & 0 \end{bmatrix}$$

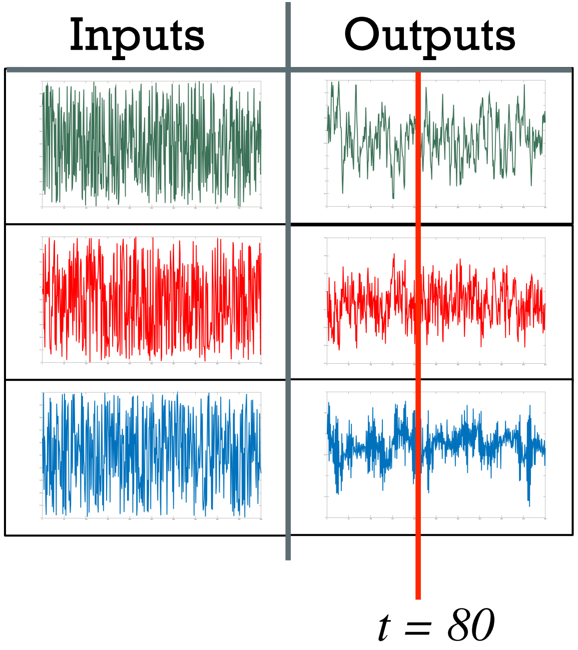




Numerical Example

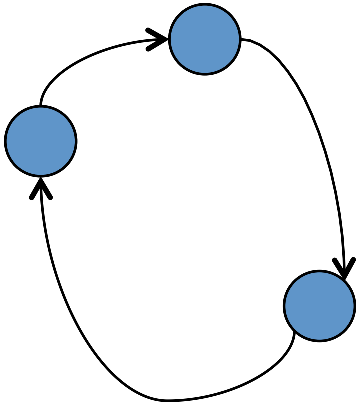
$$Q(z) = \begin{bmatrix} 0 & 0 & \frac{41.28}{(4z-3)(5z+3)(20z-11)} \\ \frac{-2}{20z+7} & 0 & 0 \\ 0 & \frac{-.01z+.75}{z^2-1.8z+.8} & 0 \end{bmatrix}$$

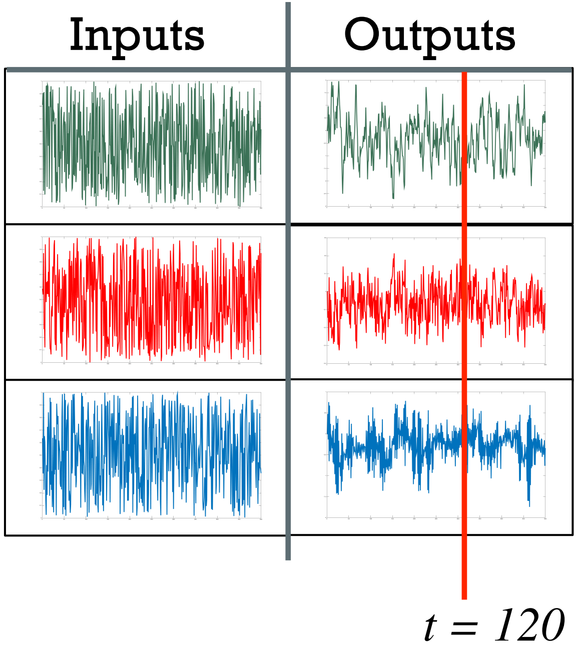




Numerical Example

$$Q(z) = \begin{bmatrix} 0 & 0 & \dfrac{41.28}{(4z-3)(5z+3)(20z-11)} \\ \dfrac{-2}{20z+7} & 0 & 0 \\ 0 & \dfrac{292}{(20z-17)(20z-19)} & 0 \end{bmatrix}$$





Numerical Example

State space model

$$x[k+1]=\begin{bmatrix} .75 & 0 & 0 & 0 & 0 & 1.2 \\ -.1 & -.35 & 0 & 0 & 0 & 0 \\ 0 & 0 & .85 & -1 & 0 & 0 \\ 0 & -.73 & 0 & .95 & 0 & 0 \\ 0 & 0 & .43 & 0 & -.6 & 0 \\ 0 & 0 & 0 & 0 & .2 & .55 \end{bmatrix} x[k]+\begin{bmatrix} 1.4 & 0 & -1.4 \\ 0 & -.25 & 0 \\ 0 & 0 & .75 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} u[k]$$

$$y[k]=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} x[k]$$

Convolution Representation

$$Q(k)[1,3]=.5096(.75)^k-.1108(-.6)^k-.8158(.55)^k+.417\delta$$

$$Q(k)[2,1]=.2839(-.3556)^k-.2839\delta$$

$$Q(k)[3,2]=7.684(.95)^k-8.588(.85)^k+.904\delta$$

$$P(k)[1,1]=1.867(.75)^k-1.867\delta$$

$$P(k)[1,3]=-P(k)[1,1]$$

$$P(k)[2,2]=.7143(-.35)^k-.7143\delta$$

$$P(k)[3,3]=.8824(.85)^k-.8824\delta$$

Unique Derivation

$$Q(z)=\begin{bmatrix} 0 & 0 & \dfrac{41.28}{(4z-3)(5z+3)(20z-11)} \\ \dfrac{-2}{20z+7} & 0 & 0 \\ 0 & \dfrac{292}{(20z-17)(20z-19)} & 0 \end{bmatrix}$$

$$P(z)=\begin{bmatrix} \dfrac{5.6}{4z-3} & 0 & \dfrac{-5.6}{4z-3} \\ 0 & \dfrac{-5}{20z+7} & 0 \\ 0 & 0 & \dfrac{15}{20z-17} \end{bmatrix}$$

Inverse Z-transform

+ Outline

- System Representations
- Network Reconstruction
  - Target Specificity
  - Active vs. Passive Reconstruction
- Main Result
  - Dynamical Structure Function Representations
  - Reconstruction Process
  - Numerical Example
- Conclusions and Future Work

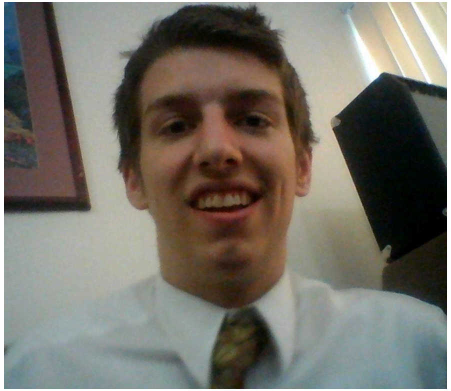
Conclusions

- Developed a time domain representation of the dynamical structure function
- Developed necessary and sufficient conditions for reconstruction of the time domain dynamical structure function
- Developed a methodology for reconstructing a dynamical structure function from passive measurements

Future Work

- Many open problems, including:
  - What is the best way to choose r?
  - Can we reconstruct in the time domain without measuring inputs?
  - Characterizing the inputs to ensure informative data
  - Time domain reconstruction with measurement or process noise

Other Authors



Joel Eliason

Sean Warnick

Thank You