Data Driven Koopman Operator Theoretic Framework for Nonlinear System and Control Applications

Amit Surana

suranaa@utrc.utc.com, 860-610-7690

United Technologies Research Center,

East Hartford, CT, 06118

UTC IASE Seminar Series, University of Connecticut

3rd Oct, 2016

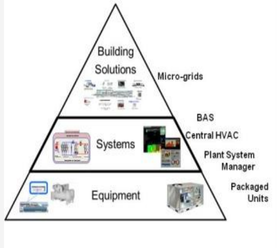
United Technologies
Research Center

Data Driven Applications @UTC

UTC Applications

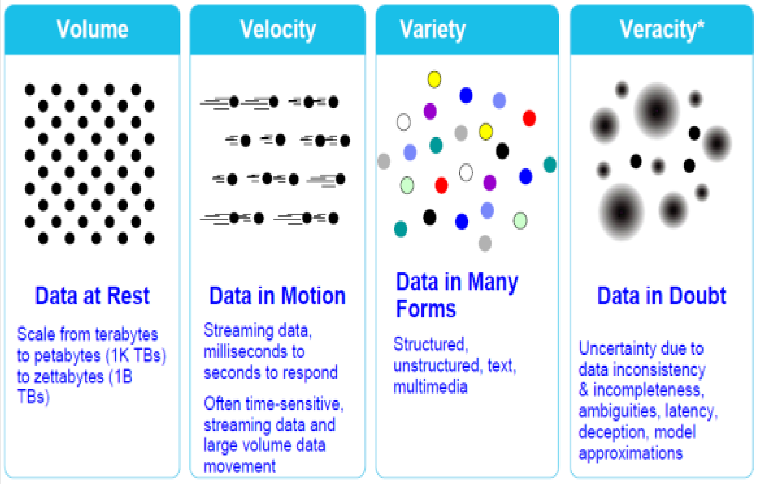
Pratt &Whitney

Climate, Control & Security



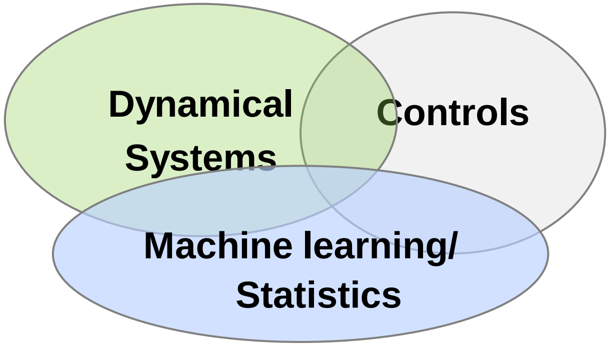
United Technologies Aerospace Systems

- ▪Service Technologies
- ▪Aerospace prognostics and health management
- ▪Building energy diagnostics and prognostics
- ▪Building security
- ▪Autonomy Intelligence Systems



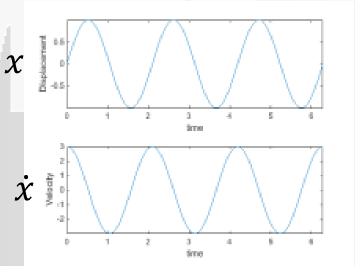
Volume	Velocity	Variety	Veracity*
Data at Rest	Data in Motion	Data in Many Forms	Data in Doubt
Scale from terabytes to petabytes (1K TBs) to zettabytes (1B TBs)	Streaming data, milliseconds to seconds to respond Often time-sensitive, streaming data and large volume data movement	Structured, unstructured, text, multimedia	Uncertainty due to data inconsistency & incompleteness, ambiguities, latency, deception, model approximations

Interdisciplinary approach

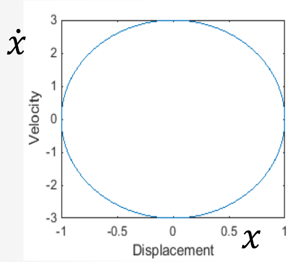


Dynamical Systems Framework

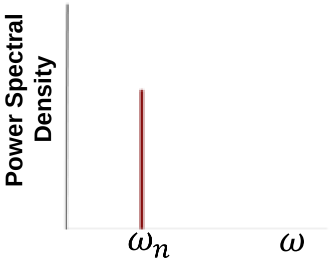
State-vs-time (Newton)



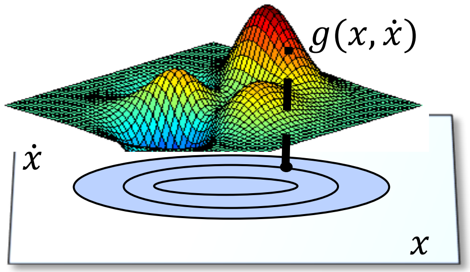
State-vs-state (Poincare)



Spectral (Wiener)



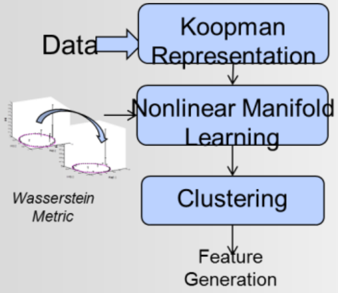
Dynamics of observables (Koopman)



Provides unifying approach*

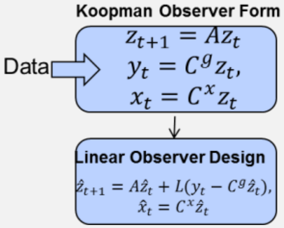
Machine Learning for Time Series Data

- *Forecasting/Anomaly detection*
- *Automatic feature generation*
- *Causal inference*



Data Driven System & Controls

- Nonlinear stability
- System Id
- *Estimation/control*

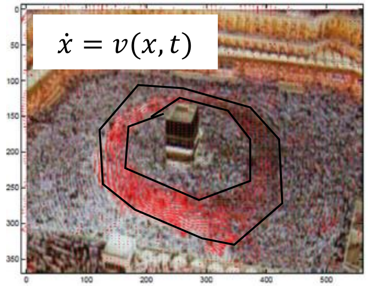


$$z_{t+1} = Az_t$$
$$y_t = C^g z_t,$$
$$x_t = C^x z_t$$

$$\hat{z}_{t+1} = A\hat{z}_t + L(y_t - C^g \hat{z}_t),$$
$$\hat{x}_t = C^x \hat{z}_t$$

Signal Processing

- Background subtraction
- *Crowd video analysis*



**Spectral Operator Theory Methods in Applied Dynamical Systems, I. Mezic, 2016*

Koopman Based Nonlinear Systems & Control Applications

Multi Input Multi Output System

$$\dot{x} = f(x) + \sum_{l=1}^{m} g_l(x) u_i$$

$$y = h(x)$$

$$x \in R^d, u \in R^m, y \in R^p$$

Transformation →

- Bilinear
- Liptchitz
- Observer forms
- Feedback linearization

…

Explore ***data driven Koopman operator framework*** to facilitate nonlinear estimator/controller design

United Technologies Research Center

Koopman Operator Theoretic Framework

Geometric

Dynamics: $\dot{x} = f(x),$

$x \in M \subset R^d$

Flow Map: $\Phi(t, x_0)$

Operator Theoretic: Infinite dimensional but linear

$Function\ Space$: $F = \{\psi : \psi : M \rightarrow C\}$

$Koopman\ (semi)group, U^t : F \rightarrow F$

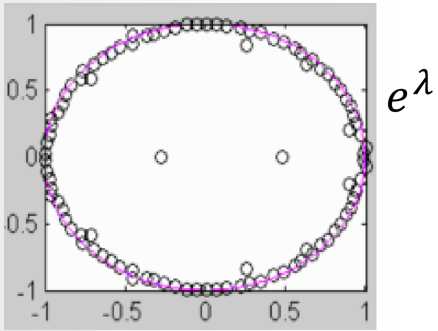
$$(U^t \psi)(x) = \psi(\Phi(t, x))$$

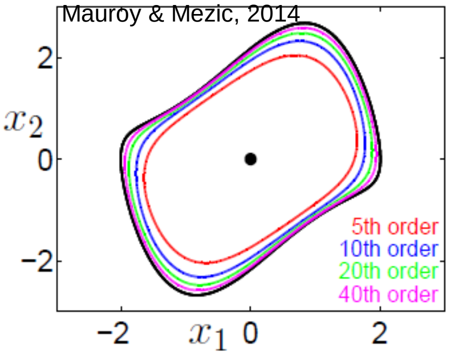
$Spectra$:

$$(U^t \phi)(x) = e^{\lambda t} \phi(x)$$

λ -Koopman eigenvalues (KEs) determine *oscillatory behavior and growth/decay rate*

$\phi(x)$ -Koopman eigenfunctions (KEFs) determine phase space geometry




Mauroy & Mezic, 2014

Koopman spectra *(Mezic et. al.)*

- Contains eigenvalues of linearization for fixed points, Floquet exponents for limit cycle, ..
- Intersection of level sets of KEFs encode solutions
- Basins of attraction, isostables, ergodic partitions, etc. can be characterized in terms of KEFs

Some Remarks

- Linear system $\dot{x} = Ax$

$Let\ w_i, \lambda_i\ are\ \ left\ eigenvector\ /eigenvalue\ of\ A$

$\phi_i(x) =< x, w_i >\ is\ KEF\ with\ KE\ \lambda_i$

$$U^t \phi_i(x_0) = e^{\lambda_i t} \phi_i(x_0)$$

$$x = \sum_{i=1}^{d} < x, w_i > \underbrace{v_i}_{right\ eigenvector\ \ of\ A}$$

$$\Rightarrow x(t, x_0) = \sum_{i=1}^{d} e^{\lambda_i t} \phi_i(x_0) v_i$$

There are infinite KEFs/KEs

$$\phi_1 \to \lambda_1 \qquad \phi_1\phi_2 \to \lambda_1 + \lambda_2$$
$$\phi_2 \to \lambda_2$$

Koopman Mode Decomposition (KMD)

$$\dot{x} = f(x) \rightarrow \ \Phi(t, x_0)$$

$$U^t \phi = \phi\big(\Phi(t, x_0)\big) = e^{\lambda t} \phi(x_0)$$

Observable/output: $h(x)\colon M \rightarrow R^p$

$$h(x) = \sum_{i=1}^{\infty} \phi_i(x) v_i, \qquad v_i \quad \in C^p$$

KMD

(Mezic, 2005)

$$\boxed{\Rightarrow h\big(\Phi(t, x_0)\big) = \sum_{i=1}^{\infty} e^{\lambda_i t} \phi_i(x_0) v_i}$$

Koopman Modes (KMs)

- KEFs/KEs are intrinsic to dynamics, KMs are specific to an observable
- KMs capture correlations between different measurements

Koopman in Discrete Time

Map $x_{k+1} = f(x_k)$

Koopman Operator
$$(K\psi)(x) = \psi\big(f(x)\big),$$
$$\big(K\phi_j\big)(x) = \lambda_j \phi_j(x)$$

$$h(x_k) = \sum_{j=1}^{\infty} \lambda_j^k \phi_j(x_0) v_j$$

- Along periodic orbit of linear/nonlinear system: KMD is equivalent to Discrete Fourier Transform (DFT)

$$S = \{x_0, \cdots, x_{P-1}\}, \qquad \mathrm{x_t} = \sum_{j=0}^{P-1} \underbrace{e^{2\pi i j \frac{t}{P}}}_{\text{Koopman eigenvalues}} \underbrace{v_j}_{\text{Koopman modes}}, \ \mathrm{t} = 0, \cdots, P-1$$

- KMD ≡ Generalized Fourier Analysis (Chen et. al., 2012)

	Discrete Fourier Transform	KMD
Captures modal growth/decay rates	No	Yes
Non-periodic data	Slow decay	Faster decay
Length of data	Sensitive	Less sensitive
Smallest frequency resolved	Length of data	Theoretically no lower bound

- Compared to Proper Orthogonal Decomposition (POD), KMD
  - more effectively decouples dynamics at different time scales (Susuki et. al. 2011)
  - provides evolution equation for observables

Numerical Approaches for KMD

$$\begin{tikzcd}
 & \text{Map} \\
x_{k+1}= f(x_k) \arrow[ur] \arrow[dr] & \\
 & \Phi(\Delta t, x), \Phi\ is\ flow\ map\ for\ an\ ODE
\end{tikzcd}$$

Approach	Remarks
Generalized Fourier/Laplacian Analysis (Mezic, 2012)	Computes KMs using time averages by subtracting most unstable mode; requires knowledge of KEs
Dynamic Mode Decomposition (DMD) and variants (Rowely et al., 2009; Schmid, 2010; Javanovic et al., 2013, Tu et al., 2013)	Based on Arnoldi type methods; necessary/sufficient conditions exist for it to recover KEs, KEFs, and KMs
Extended DMD (Williams et al., 2015)	Galerkin method, computes KEs, KEFs, and KMs using chosen dictionary or kernel function

$$Dictionary\text{:}\ D = \{\psi_1, \cdots, \psi_N\}, \psi_i\text{:} M \to C$$

$$let\ \ h = \psi^* a, \qquad \bar{h} = \psi^* \bar{a} \quad where\ \psi = (\psi_1, \cdots, \psi_N)^*$$

$$\bar{h}(x) \approx (Kh)(x)$$

$$\Rightarrow \psi^*\ Ka \approx (\psi \circ f)^* a$$

Numerical Approaches for KMD: Extended DMD

$Let\ \{x_m, \bar{x}_m = f(x_m)\}_{m=1}^{M}$ be obtained from simulations /experiments (no need for equations)

1. Compute $G = \Psi^{*}\Psi,\ \ \mathrm{A} = \Psi^{*}\bar{\Psi}$, where

$$\Psi = \begin{pmatrix} \psi^{*}(x_1) \\ . \\ \psi^{*}(x_M) \end{pmatrix} \text{and } \bar{\Psi} = \begin{pmatrix} \psi^{*}(\bar{x}_1) \\ . \\ \psi^{*}(\bar{x}_M) \end{pmatrix}$$

2. Compute eigenvalues λ_i , right eigenvectors ξ_i of

$$\bar{K} = G^{\#}A$$

3. KEfns are $\phi_i(x) \approx \psi^{*}(x)\xi_i$ with KEs $\lambda_i, i = 1 \cdots, N$

4. KMs are rows of

$$V_x \approx B_x W,\ \ V_h \approx B_h W$$

where, $x = B_x\psi$ and $\mathrm{h}(x) = B_h\psi$ and W is matrix with columns as left eigenvectors of $\bar{K}$

(Williams et al., 2015)

Challenges	Remarks
High dimensional state space	Kernel EDMD (Williams et al., 2016)
Choice of basis/kernel	Exploit machine learning concepts
Where to sample data and how much	System ID concepts
Limited data in experimental setting	Enrich data (e.g. Taken’s embedding)

KMD/DMD Applications

- Modal decomposition/time series analysis
  - Fluid mechanics (Schmid 2009, Rowley et. al. 2009, Mezic 2012, Bageri 2014)
  - Building energy modeling (Eisenhower et. al 2010, Georgescu & Mezic, 2015)
  - Video background & foreground separation (Grosek & Kutz, 2013)
  - Compressed sensing/sparsity/sensor placement (Brunton et. al, 2013)
  - Anomaly detection in power grid (Susuki & Mezic, 2014)
  - Dynamic texture modeling (Surana, 2015)
  - Data fusion (Willams et. al, 2015)
  - Multi resolution DMD (Kutz, 2015)
  - Coherent patterns extraction in neurological data (Brunton et. al, 2015)
  - Classification, forecasting, and fault detection/isolation (Surana et. al, 2016, in preparation)

  …….

- Systems/control applications
  - Perron-Frobenius (adjoint of Koopman)
    - stability/basins of attraction (Vaidya & Mehta, 2008; Wang & Vaidya, 2010)
    - observability gramian (Vaidya, 2007)
  - Koopman:
    - stability/basins of attraction (Mauroy & Mezic, 2014)
    - differentially positive systems (Mauroy et. al, 2015)
    - control using Koopman invariant subspaces (Brunton et. al, 2016)
  - DMD with control (Proctor et. al. 2016)

Koopman Observer Form: Preliminaries

- If $\phi(x), \lambda$ is KE/KEF, then so is $\bar{\lambda}, \bar{\phi}(x)$, and $\dot{x} = f(x)$

$$L_f \phi = f.\nabla\phi = \lambda\phi, \qquad L_f \bar{\phi} = \bar{\lambda}\bar{\phi},$$

$$L_f \begin{pmatrix} Re(\phi) \\ -Im(\phi) \end{pmatrix} = \mathrm{Q}_\lambda \begin{pmatrix} Re(\phi) \\ -Im(\phi) \end{pmatrix}, \qquad \mathrm{Q}_\lambda = |\lambda| \begin{pmatrix} \cos arg\lambda & \sin arg\lambda \\ -\sin arg\lambda & \cos arg\lambda \end{pmatrix}$$

- $F^n = span\{\phi_1, \ldots, \phi_n\}$, define

$$T(x) = \left(\hat{\phi}_1(x), \ldots, \hat{\phi}_n(x)\right)^* : M \to R^n$$

  - $\hat{\phi}_i = \phi_i$, if ϕ_i $is\ real\ valued$
  - $(\hat{\phi}_i, \hat{\phi}_{i+1}) = \left(2Re(\phi_i), -2Im(\phi_i)\right)\ if\ \phi_i\ is\ complex\ valued$

- For real valued observable $h(x) \in F^n$

$$h(x) = \sum_{i=1}^{n} \phi_i v_i = CT(x)$$

where, $C = [v_1 \ldots.Re(v_i) \quad -Im(v_i) \ldots.] \in R^{p\times n}$

$$L_f T(x) = \Lambda T(x), \qquad \Lambda \in R^{n\times n} \ \text{block diagonal}$$

$$-\Lambda_{i,i} = \lambda_i, \qquad -real\ \ \lambda_i$$

$$-\begin{pmatrix} \Lambda_{i,i} & \Lambda_{i,i+1} \\ \Lambda_{i+1,i} & \Lambda_{i+1,i+1} \end{pmatrix} = Q_\lambda, \qquad -complex\ \ \lambda_i$$

Input Output Koopman Observer Form (IOKOF)

Let there exists a finite set of KEFs

$$F^n = span\{\phi_1, \dots, \phi_n\}$$

of autonomous part $\dot{x} = f(x)$, such that

$$x, h(x) \in F^n.$$

$Then\ using\ T(x) = \left(\hat{\phi}_1(x), \dots, \hat{\phi}_n(x)\right)^*$

Low dimensional

$$\begin{aligned} \dot{x} &= f(x) + \sum_{l=1}^{m} g_l(x) u_i \\ y &= h(x) \end{aligned} \quad \xrightarrow{z = T(x)} \quad \begin{aligned} \dot{z} &= Az + \sum_{l=1}^{m} \tilde{g}_l(z)\, u_l \\ y &= Cz \end{aligned}$$

High dimensional

$where,$

$-A = \Lambda \rightarrow KEs$

$-C \rightarrow KMs\ for\ h(x)$

$-C^x \rightarrow KMs\ for\ \ x$

$-\tilde{g}_l(z) = L_{g_l} T(x) \Big|_{x = C^x z}$

Remarks:

- Truncation
- $T(x)$ may only be "computable" locally, e.g. in basin of attraction

A. Surana and A. Banaszuk, Linear observer synthesis for nonlinear systems using Koopman operator framework, NOLCOS, 2016.

A. Surana, Koopman Operator Based Observer Synthesis for Control-Affine Nonlinear Systems, accepted to appear in CDC 2016.

Special Cases of IOKOF

$$\begin{aligned}\dot{x} &= f(x) + \sum_{l=1}^{m} g_l(x)u_i \\ y &= h(x)\end{aligned} \xrightarrow{z = T(x)} \begin{aligned}\dot{z} &= Az + \sum_{l=1}^{m} \widetilde{g}_l(z)u_l \\ y &= Cz\end{aligned}$$

$$u \equiv 0$$

Linear

$$\dot{z} = Az$$
$$y = Cz$$

$$L_{g_l}T(x) = b_l + \underbrace{B_l T}_{\text{KMs}}(x)$$

$$l = 1, .., m$$

Bilinear

$$\dot{z} = Az + \mathrm{B}_0\mathrm{u} + \sum_{l=1}^{m} (B_l z)u_l$$

$$y = Cz$$

where, $B_0 = [b_1 \ldots b_m] \in R^{n\times m}$

$$\left|\left|\Phi(z_1, u) - \Phi(z_2, u)\right|\right| \le \gamma\left|\left|z_1 - z_2\right|\right|$$

$$\Phi(z, u) = \sum_{l=1}^{m} \widetilde{g}_l(z)\, u_l$$

Liptchitz

$$\dot{z} = Az + \Phi(z, u)$$
$$y = Cz$$

Linear KOF

- Nonlinear Observability Characterization

$$\begin{matrix}\dot{x} = f(x) \\ y = h(x)\end{matrix} \xrightarrow{x, h(x) \in F^n} \begin{matrix}\dot{z} = Az \\ y = Cz\end{matrix}$$

$$If\ pair\ (A, C) is\ observable \Rightarrow nonlinear\ system\ is\ nonlineary\ observable$$

"indistinguishability"

- Observer Design: Luenberger/Kalman

$$\dot{\hat{z}} = A\hat{z} + L(y - C\hat{z}) \longrightarrow \hat{x} = C^x \hat{z}$$

$$e_z = z - \hat{z},$$
$$\dot{e_z} = (A - LC)e_z \to 0$$

$$e_x = x - \hat{x} = C^x e_z \to 0$$

Compared to	KOF based approach
EKF	-"more global" convergence
Observer forms (Krener et. al)	-immersion rather than diffeomorphism -more readily computable -data driven/equation free
Carleman linearization	-may provide more compact representation

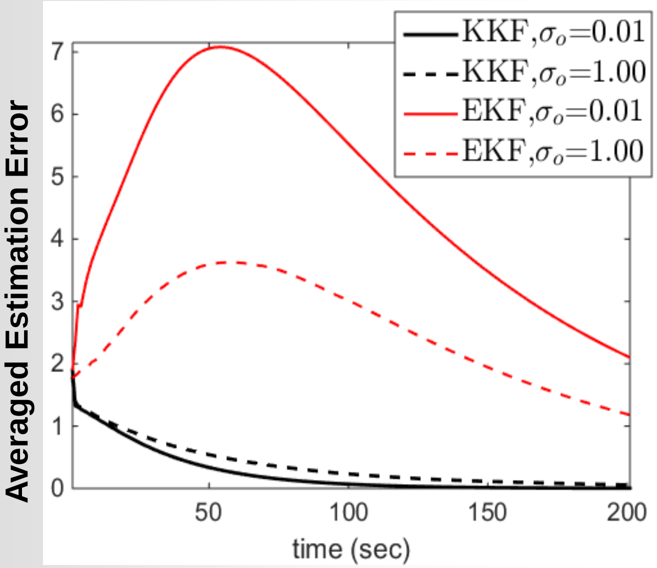
Nonlinear Estimation Application: Example I

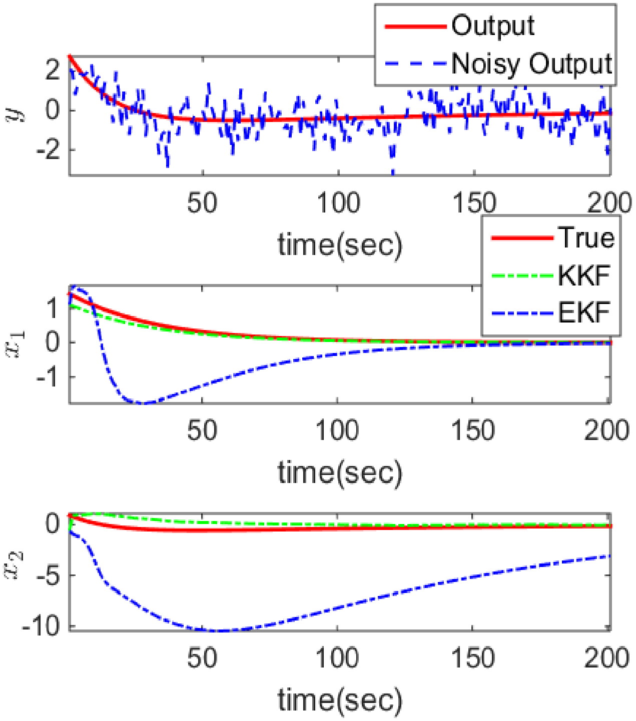
$$x_{k+1} = f(x_k) = \begin{pmatrix} \rho x_{1,k} \\ \mu x_{2,k} + (\rho^2 - \mu)\mathrm{c}x_{1,k}^2 \end{pmatrix}$$

$$y_k = h(x_k) = x_{1,k}^2 + x_{2,k}$$

$$x, h(x) \in span\{\phi_1, \phi_2, \phi_3\}$$
$$\phi_1 = x_1, \rightarrow \rho$$
$$\phi_2 = x_2 - cx_1^2, \rightarrow \mu$$
$$\phi_3 = x_1^2, \rightarrow \rho^2$$

$$A = \begin{pmatrix} \rho & 0 & 0 \\ 0 & \mu & 0 \\ 0 & 0 & \rho^2 \end{pmatrix}$$
$$C = (0 \quad 1 \quad 1 + c)$$
$$C^x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & c \end{pmatrix}$$



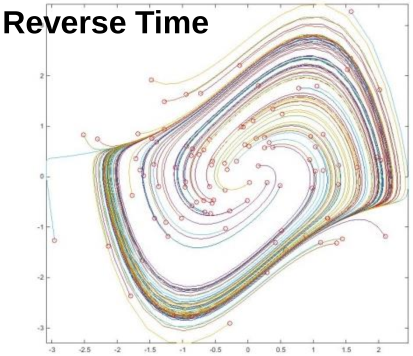


Superior performance than EKF

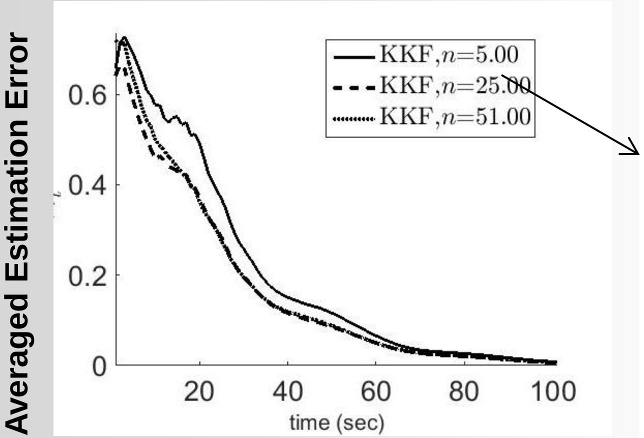
Nonlinear Estimation Application: Example II

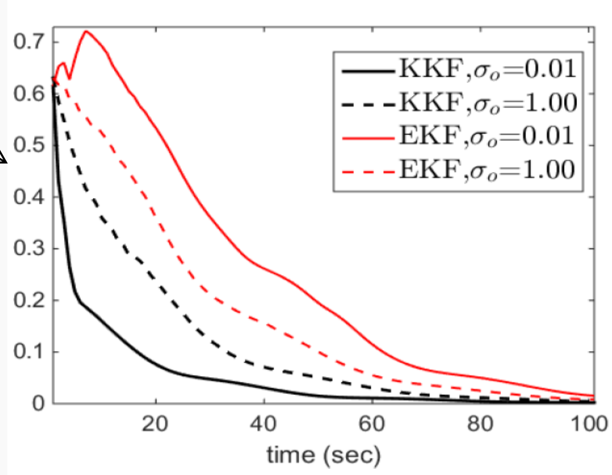
$$x_{k+1} = f(x_k) = \begin{pmatrix} x_{1,k} - x_{2,k}dt \\ x_{2,k} + dt\,(x_{1,k} - x_{2,k} + x_{1,k}^2 x_{2,k}) \end{pmatrix}$$

$$y_k = h(x_k) = x_{1,k}^2 + x_{2,k}$$

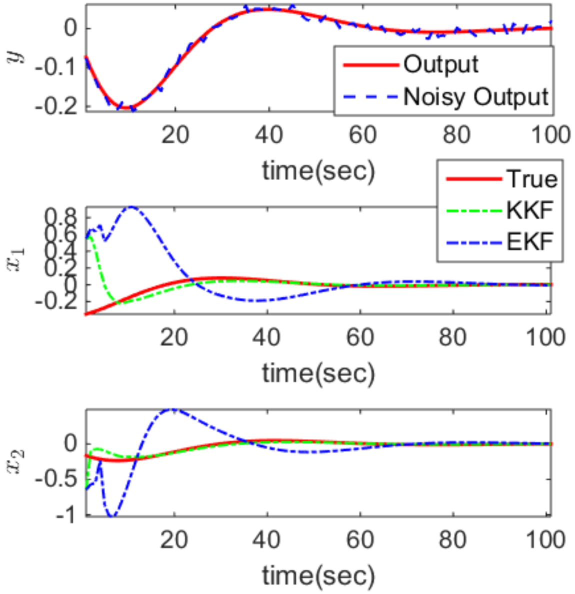


Used Kernel extended DMD (Williams, et. al, 2016) for computation of KE/KEF/KM





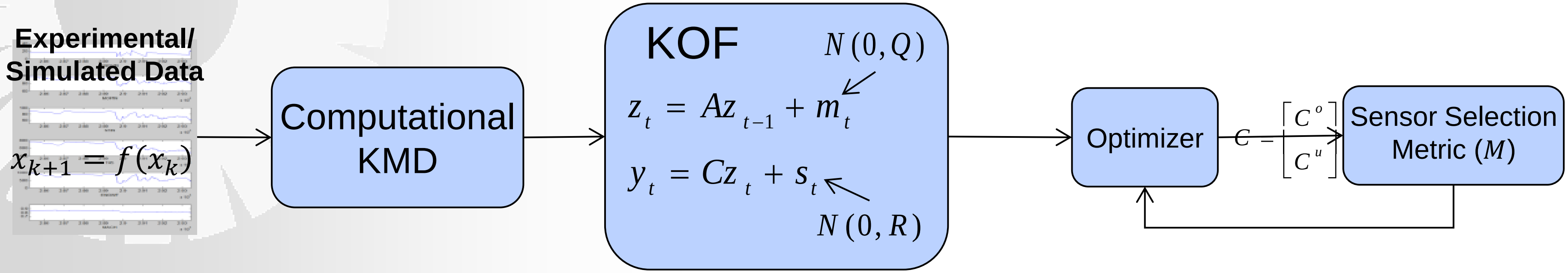
EKF diverges:
$21\%\ (\sigma_0 = 0.01), 5\%(\sigma_0 = 1)$



Superior performance than EKF

United Technologies Research Center

KOF Based Sensor Selection



Experimental/ Simulated Data

$x_{k+1} = f(x_k)$

Computational KMD

KOF

$N(0,Q)$

$z_t = Az_{t-1} + m_t$

$y_t = Cz_t + s_t$

$N(0,R)$

Optimizer

$C = \begin{bmatrix} C^o \\ C^u \end{bmatrix}$

Sensor Selection Metric (M)

- Enables use of linear concepts for M such as based on
  - Observability gramian
  - Error covariance (algebraic Riccati Eqn)

- Desirable properties of metrics:
  - Incorporates model error/sensor noise: Q, R
  - Invariant to sensor scaling: $y \leftarrow Sy, M$ does not change
  - Optimization amenable to efficient heuristics

Mutual information based metric

$$M = MI(y_{ss}^u, y_{ss}^o) = \frac{1}{2}\log\frac{|P_{oo}||P_{uu}|}{\begin{vmatrix} P_{oo} & P_{ou} \\ P_{ou} & P_{uu} \end{vmatrix}}$$

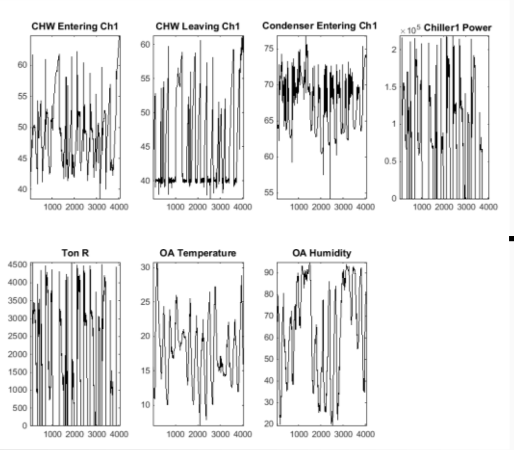
$$y_{ss}^o \sim N(\cdot, P_{oo}),\ P_{oo} = C^o P_{ss}^z (C^o)' + R^{oo}$$
$$y_{ss}^u \sim N(\cdot, P_{uu}),\ P_{uu} = C^u P_{ss}^z (C^u)' + R^{uu}$$
$$P_{ou} = C^o P_{ss}^z (C^u)' + R^{ou}$$

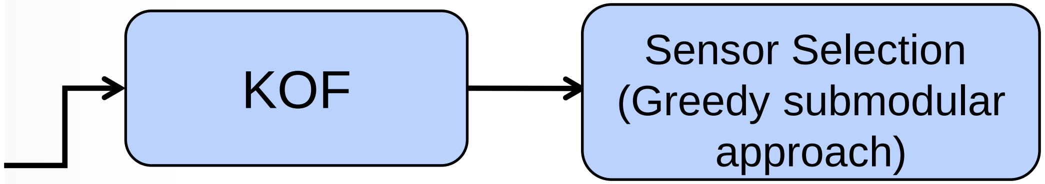
P_{ss}^z – Steady state error covariance (solution of discrete ARE for pair (A, C^o, Q, R^{oo})

Submodularity concepts for scalable optimization

Data Driven Sensor Selection Application

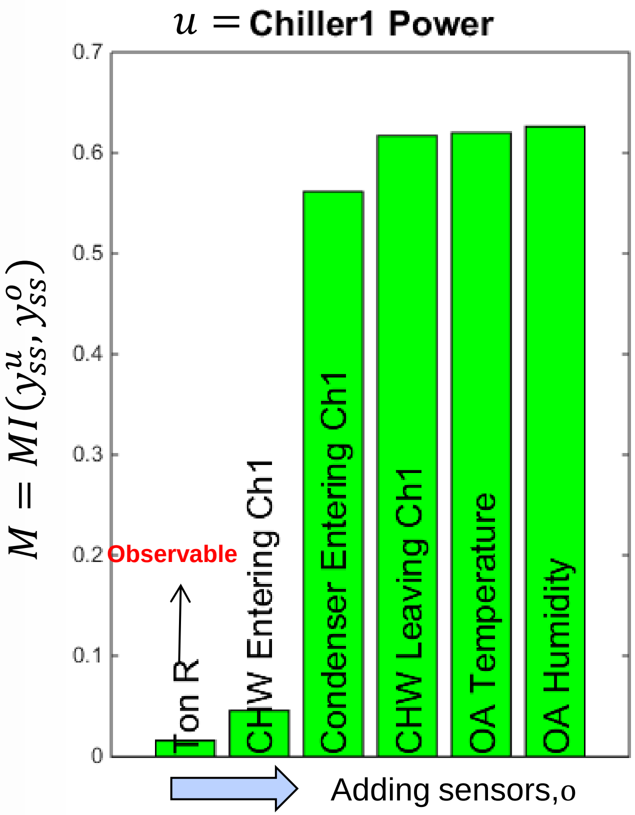
Experimental Data:





KOF

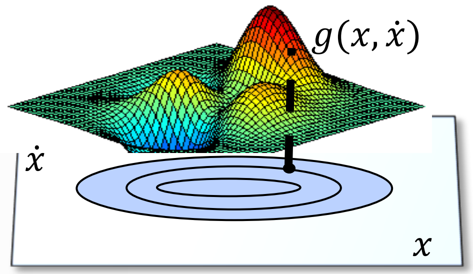
Sensor Selection (Greedy submodular approach)



United Technologies Research Center

Dynamical Systems Framework

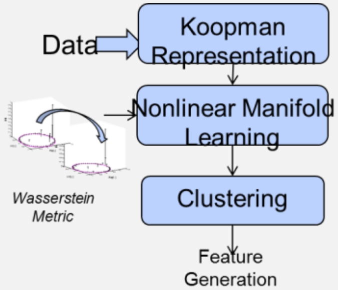
Dynamics of observables (Koopman)



Provides unifying approach*

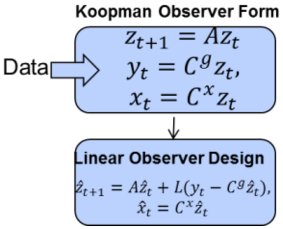
Machine Learning for Time Series Data

- *Forecasting/Anomaly Detection*
- *Automatic feature generation*
- *Causal inference*



Data Driven System & Controls

- Nonlinear stability
- System Id
- *Estimation/control*

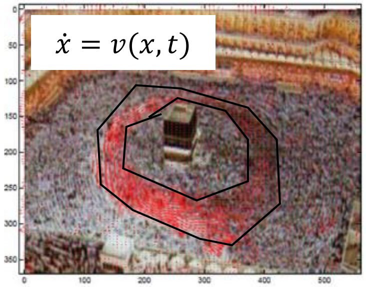


$$z_{t+1} = Az_t$$
$$y_t = C^g z_t,$$
$$x_t = C^x z_t$$

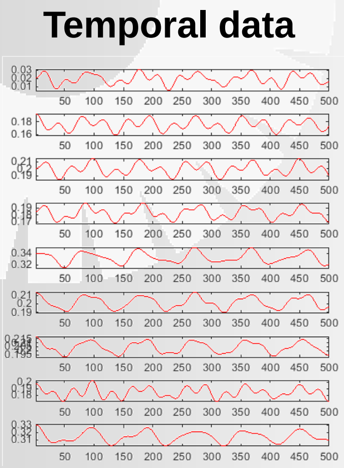
$$\hat{z}_{t+1} = A\hat{z}_t + L(y_t - C^g \hat{z}_t),$$
$$\hat{x}_t = C^x \hat{z}_t$$

Signal Processing

- Background subtraction
- *Crowd video analysis*



Koopman Based Time Series Analysis



How is it being done today?

- PCA/POD, Fourier, Bayesian Nets (BN), Neural nets..

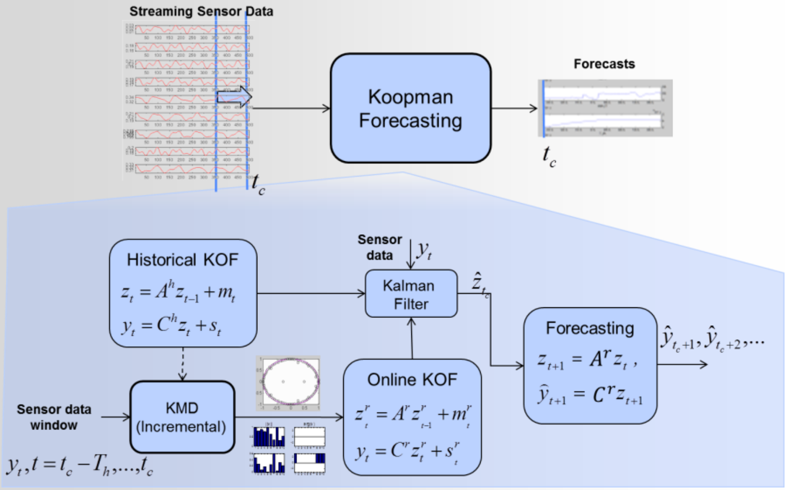
What are the deficiencies?

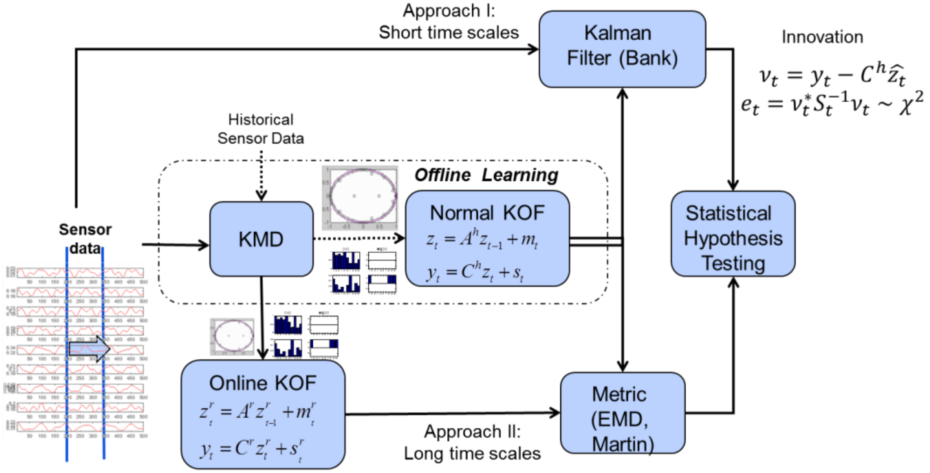
- Poor scalability due to discretization
- Hand crafted feature extraction to reduce dimensionality
- Spatiotemporal characteristics often not captured: limited ability to forecast & capture transients/nonlinearities

Enablers in machine learning

- Mixed BN, Dynamic BN, Recurrent Neural nets, Deep Learning,…

Use KOF as generative model and combine with machine learning/classical FDI techniques



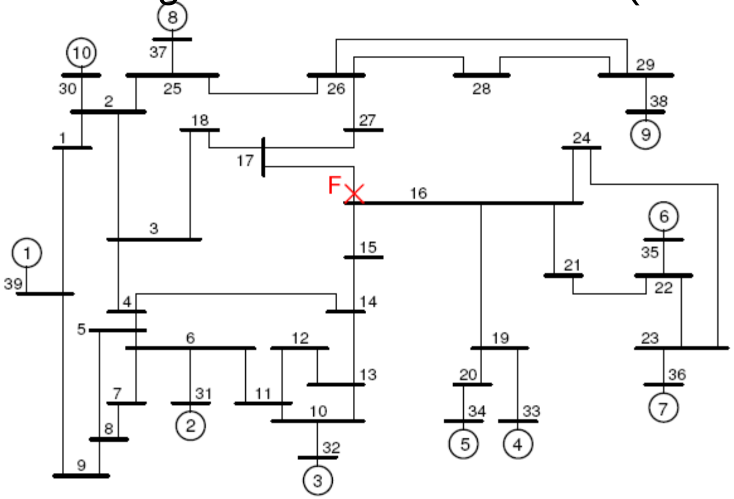


United Technologies Research Center

This page contains no technical data subject to EAR or ITAR.

Time Series Prediction/Forecasting/Anomaly Detection

New England Power Grid Model (Kundur,1994)



$i = 2, \cdots, 10$

$$\dot{\delta}_i = \omega_i$$

$$\frac{H_i}{\pi f_s} \dot{\omega}_i = -D_i \omega_i + P_{mi} - G_{ii} E_i^2 - \sum_{j=1, j \neq i}^{10} E_i E_j \{ G_{ij} \cos(\delta_i - \delta_j) + B_{ij} \sin(\delta_i - \delta_j) \}$$

Damping Coefficient

Mechanical Input Power

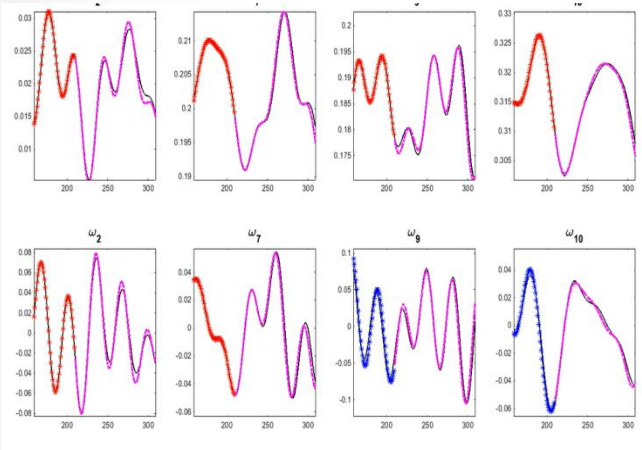
Internal Voltage

Inertia

$G_{ij} + \bar{j} B_{ij}$ Transfer impedance between generators

- ▪Simulate to generate historical data with different initial conditions
- ▪Obtain normal model: time delay embedding, exact DMD
- ▪Test data with/without fault

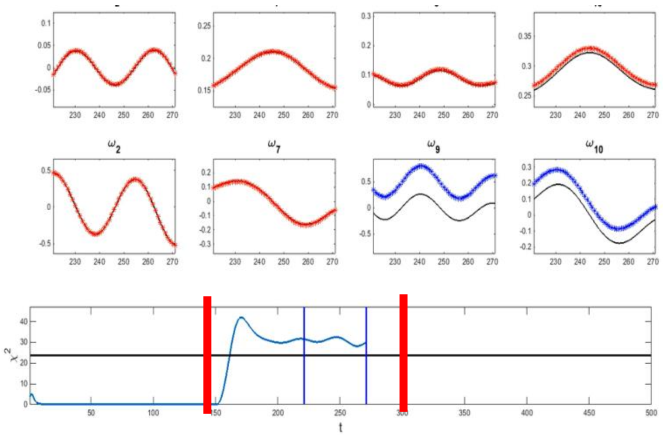
Prediction/Forecasting



----Measured ----Forecasted

----Estimated ----Ground truth

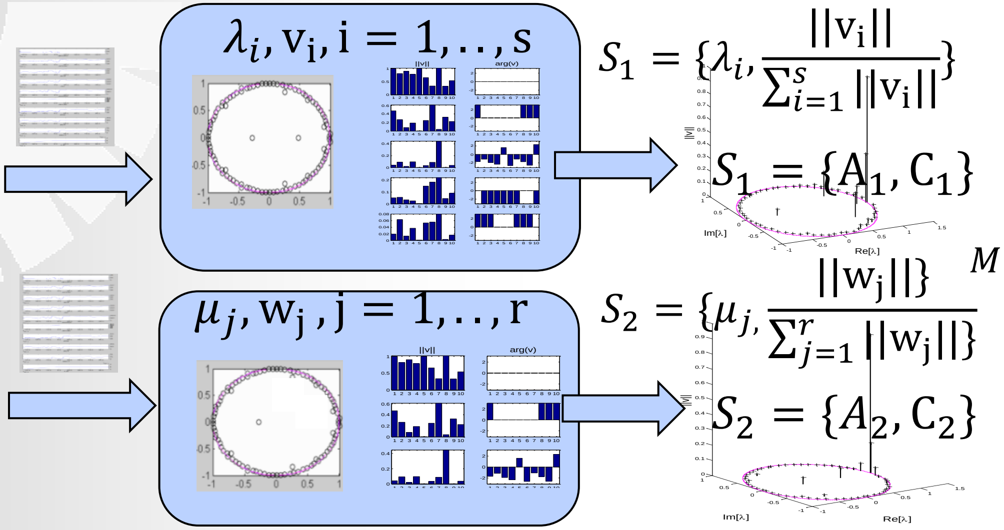
Anomaly Detection



Anomaly window: $t = [150,300]$

A. Surana, Koopman operator framework for time series analysis, to be submitted to J. Computational Dynamics.

Metrics for Koopman Based Representation



System1

$$x_t = f_1(x_{t-1})$$
$$y_t = g_1(x_t)$$

$$\lambda_i, \mathrm{v_i}, \mathrm{i} = 1, .., \mathrm{s}$$

$$S_1 = \{\lambda_i, \frac{||\mathrm{v_i}||}{\sum_{i=1}^{s} ||\mathrm{v_i}||}\}$$

$$S_1 = \{\mathrm{A_1}, \mathrm{C_1}\}$$

System 2

$$x_t = f_2(x_{t-1})$$
$$y_t = g_2(x_t)$$

$$\mu_j, \mathrm{w_j}, \mathrm{j} = 1, .., \mathrm{r}$$

$$S_2 = \{\mu_j, \frac{||\mathrm{w_j}||\}}{\sum_{j=1}^{r} ||\mathrm{w_j}||\}}$$

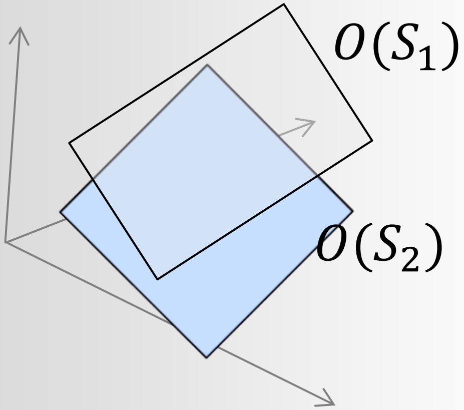
$$S_2 = \{A_2, \mathrm{C_2}\}$$

Earth Mover's Distance
(Rubner, et al., 1998)

$$M(S_1, S_2) = \left(\sum_{i=1}^{s} \sum_{j=1}^{r} f_{ij}^{*} \left\| \lambda_i - \mu_j \right\|^p \right)^{1/p}$$

Solved using linear program

Metrics for linear systems



$O(S_1)$

$O(S_2)$

1. Solve Lyapunov Eqn.

$$A^T P A - P = -C^T C$$

to obtain $P = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix}$ *where* $A = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix}, C = [C_1 \quad C_2]$

2. Compute cosine of subspace angles

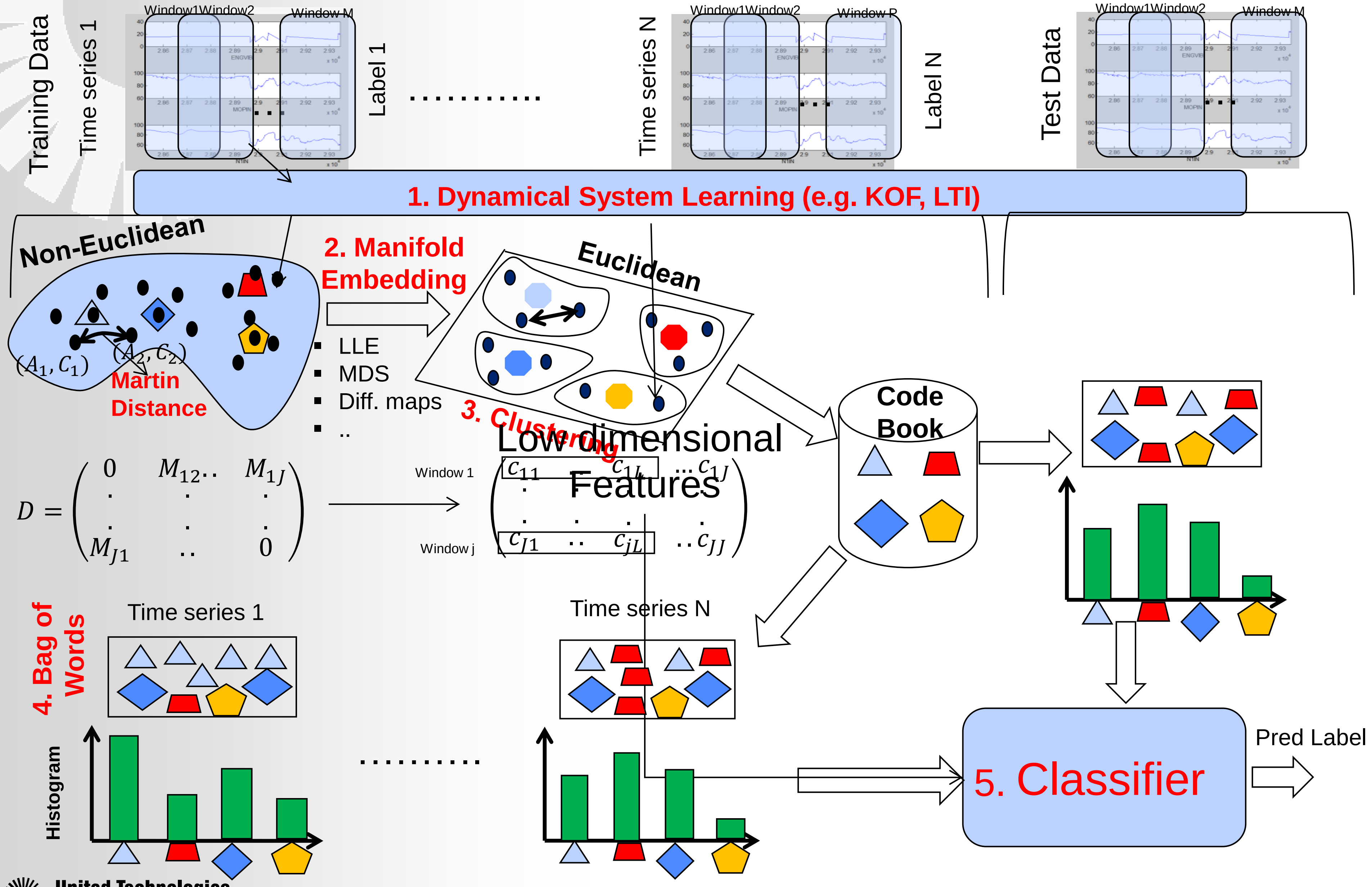
$$\cos^2 \theta_i = ith\ eigenvalue(P_{11}^{-1} P_{12} P_{22}^{-1} P_{21})$$

3. Martin distance

$$M(S_1, S_2) = \sqrt{-ln \prod_{i=1}^{n} \cos^2 \theta_i}$$

Martin, 2000

Koopman Based Automatic Feature Generation Framework



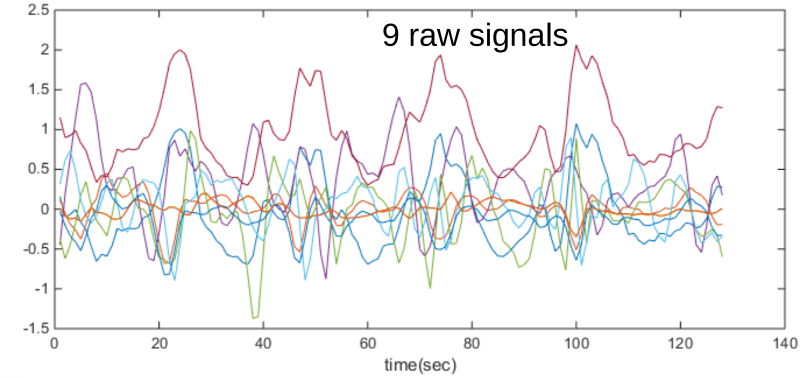
$$D = \begin{pmatrix} 0 & M_{12}.. & M_{1J} \\ . & . & . \\ . & . & . \\ M_{J1} & .. & 0 \end{pmatrix}$$

$$\begin{pmatrix} c_{11} & .. & c_{1L} & ...c_{1J} \\ . & . & . & . \\ c_{J1} & .. & c_{jL} & ..c_{JJ} \end{pmatrix}$$

Time Series Classification Application

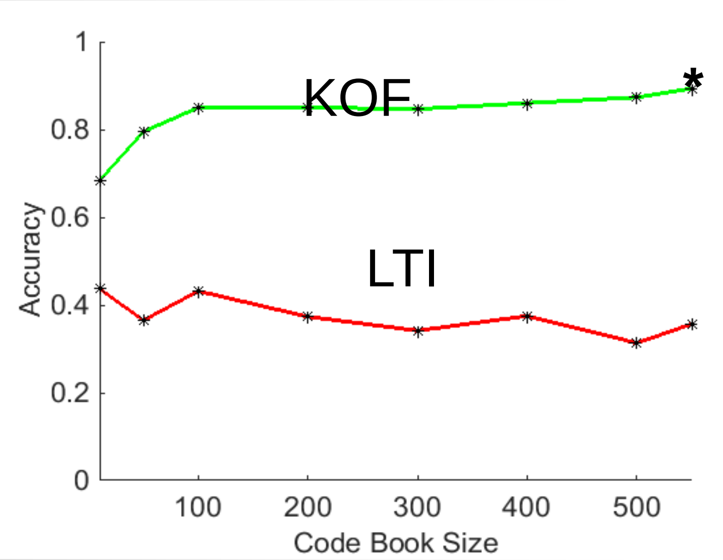
Human Activity Recognition: Healthcare, security, fitness,..

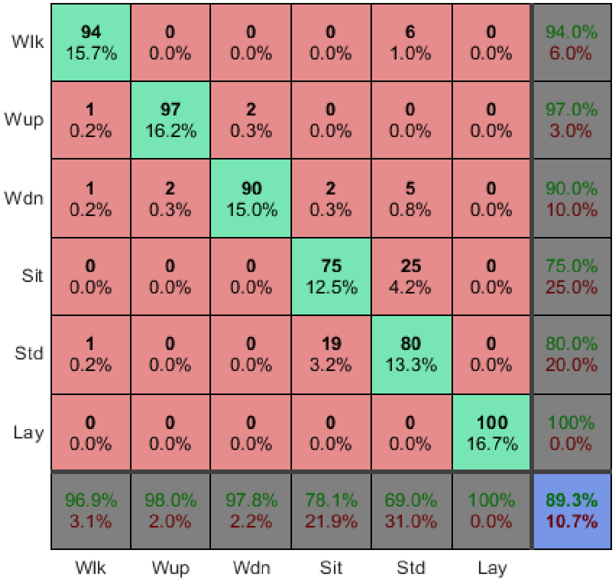
- UCI data (Anguita et al., 2012) : Walk, Walk up, Walk down, Sit, Stand, Lay
- Training: 30 subjects
- Testing: Different subjects than in training data



- Accelerometer body/total
- Gyro
- 50Hz, 2.56sec window

89% accuracy-550 features (KOF based automatic feature generation)





	Wlk	Wup	Wdn	Sit	Std	Lay	
Wlk	94 15.7%	0 0.0%	0 0.0%	0 0.0%	6 1.0%	0 0.0%	94.0% 6.0%
Wup	1 0.2%	97 16.2%	2 0.3%	0 0.0%	0 0.0%	0 0.0%	97.0% 3.0%
Wdn	1 0.2%	2 0.3%	90 15.0%	2 0.3%	5 0.8%	0 0.0%	90.0% 10.0%
Sit	0 0.0%	0 0.0%	0 0.0%	75 12.5%	25 4.2%	0 0.0%	75.0% 25.0%
Std	1 0.2%	0 0.0%	0 0.0%	19 3.2%	80 13.3%	0 0.0%	80.0% 20.0%
Lay	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 16.7%	100% 0.0%
	96.9% 3.1%	98.0% 2.0%	97.8% 2.2%	78.1% 21.9%	69.0% 31.0%	100% 0.0%	89.3% 10.7%

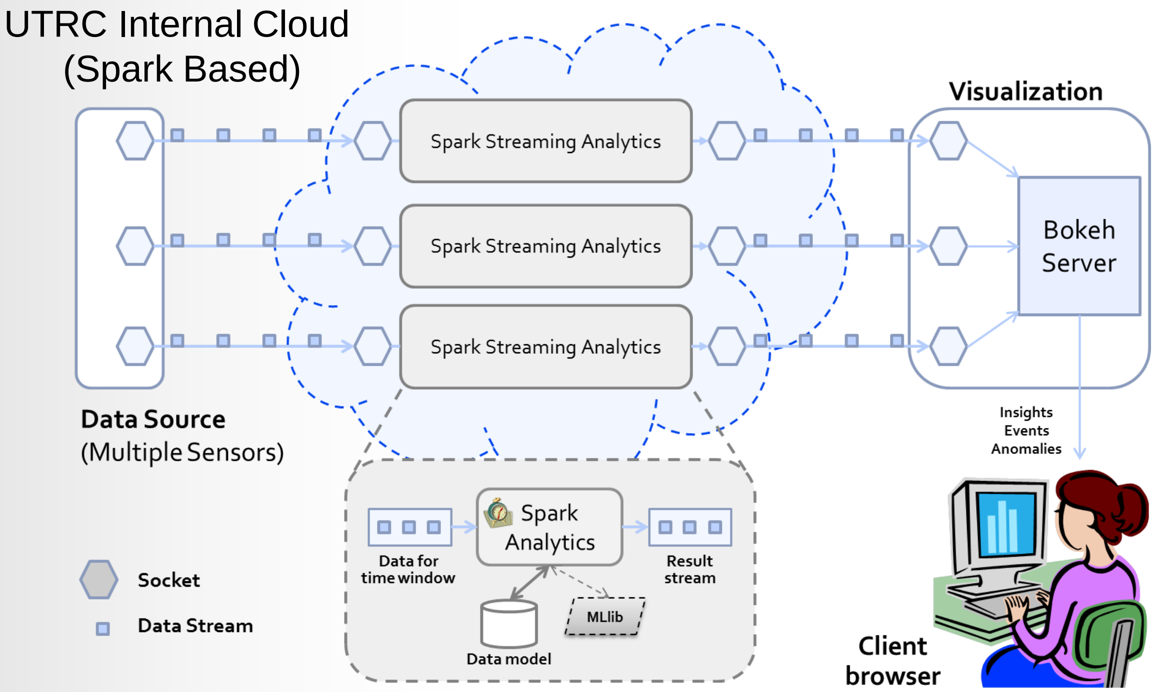
*manual expert feature (e.g. statistical, FFT based etc.) generation

A. Surana, Koopman operator framework for time series analysis, to be submitted to J. Computational Dynamics.

Cloud Based/Streaming Computations

With S. Bopardikar, N. Trcka @UTRC

Model learning/update	DMD and variants relying on matrix algebra (e.g. SVD) ▪Distributed SVD ▪Streaming: Incremental DMD (Hemati et. al. 2014)/ SVD (Brand, 2006)
Classification/clustering/ indexing & retrieval	Distance Matrix computation ▪ Pairwise Lyapunov eqn/Linear program Classical machine learning techniques (e.g. k-means, SVMs) ▪ Distributed MLLib
Estimation/Forecasting/ Anomaly Detection	Kalman filter on reduced KOF ▪Centralized/distributed ▪Recursive



UTRC Internal Cloud
(Spark Based)

Spark Streaming Analytics

Spark Streaming Analytics

Spark Streaming Analytics

Visualization

Bokeh Server

Data Source
(Multiple Sensors)

Insights
Events
Anomalies

Data for time window

Spark Analytics

Result stream

MLlib

Data model

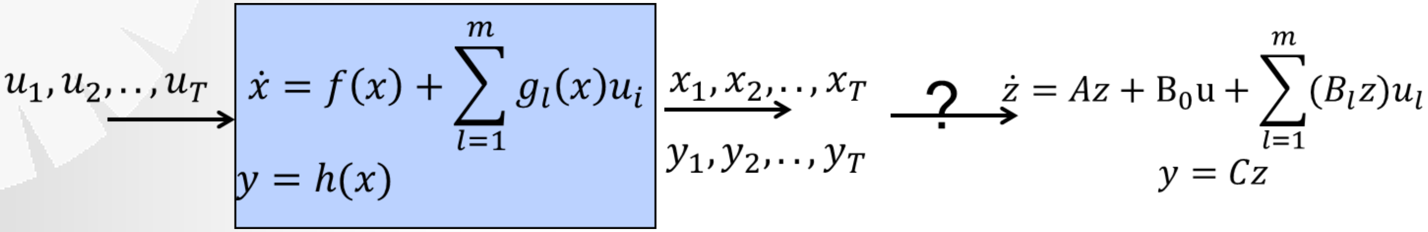
Socket

Data Stream

Client browser

Ongoing Work

- IOKOF identification



$$u_1, u_2, \ldots, u_T \longrightarrow \boxed{\begin{array}{l}\dot{x} = f(x) + \sum_{l=1}^{m} g_l(x) u_i \\ y = h(x)\end{array}} \xrightarrow[y_1, y_2, \ldots, y_T]{x_1, x_2, \ldots, x_T} \xrightarrow{?} \begin{array}{l}\dot{z} = Az + \mathrm{B}_0 \mathrm{u} + \sum_{l=1}^{m} (B_l z) u_l \\ y = Cz\end{array}$$

- Truncation accuracy vs. size tradeoff

$$h(x) \approx \sum_{i=1}^{n} v_i^h \phi_i(x)$$

$$x \approx \sum_{i=1}^{n} v_i^x \phi_i(x)$$

$$\dot{z} = Az + \sum_{l=1}^{m} \tilde{g}_l(z) u_l$$

$$y = Cz + \Delta_h(x)$$

$$x = C^x z + \Delta_x(x)$$

$$\searrow$$

$$L_{g_l} T(x) \approx b_l + B_l T(x)$$

- Comparison with other approaches (Carleman linearization, approximate linearization etc.)

- Engine control applications

United Technologies Research Center

This page contains no technical data subject to EAR or ITAR.

Ongoing Work

- Non-stationary data (with Prof. Igor Mezic@UCSB)[1]

$$x_t = f(x_{t-1}, t),$$
$$y_t = g(x_t)$$

For periodic/quasi periodic time dependence Floquet theory to extend notion of KMD

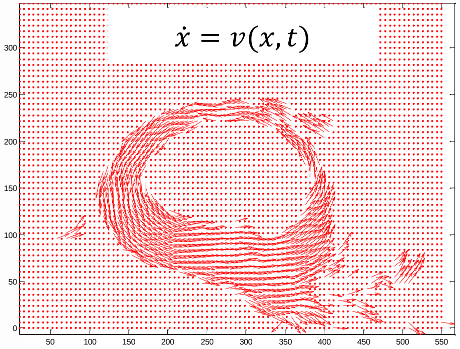
- Data driven causality vs. correlation (with K. Srivastava@UTRC)

$$\begin{aligned} x_t^1 &= f(x_{t-1}^1), \\ x_t^2 &= f_2(x_{t-1}^1, x_{t-1}^2) \end{aligned} \xrightarrow{\quad y_t = g(x_t^1, x_t^2) \quad}$$

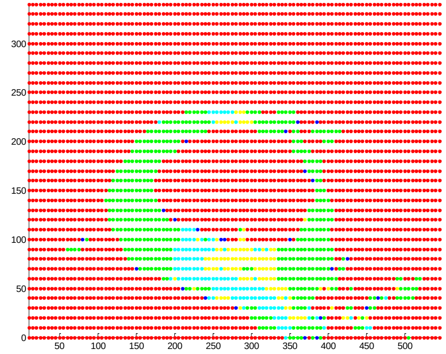
Use Koopman representation and information theoretic measures

- Crowd video analysis (with Prof. Igor Mezic@UCSB)[2]

Optical Flow



KMD

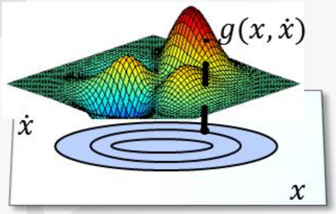


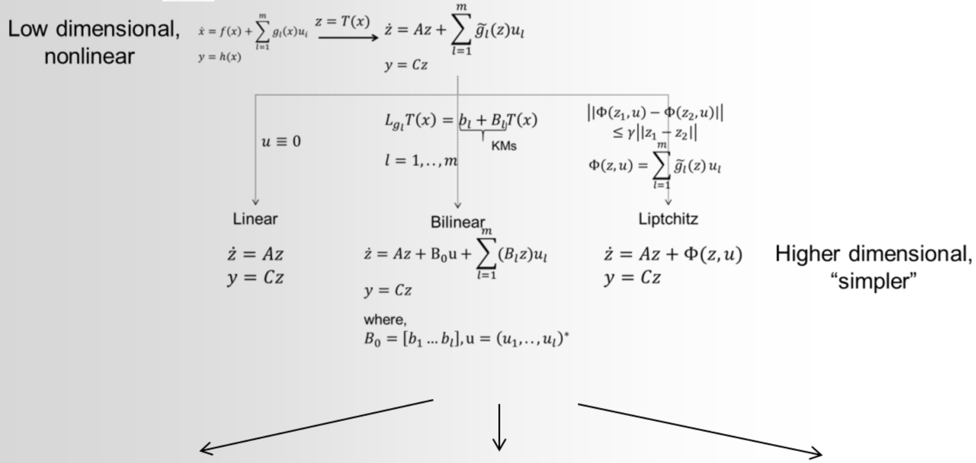
---

1. *I. Mezic,, & A. Surana, Koopman Mode Decomposition for Periodic/Quasi-periodic Time Dependence, NOLCOS, 2016.*
2. *A. Surana & I. Mezic , Dynamical systems framework for crowd video analysis, in preparation.*

Summary

Dynamics of observables





Low dimensional, nonlinear

$$\dot{x} = f(x) + \sum_{l=1}^{m} g_l(x)u_l \qquad y = h(x) \quad \xrightarrow{z = T(x)} \quad \dot{z} = Az + \sum_{l=1}^{m} \tilde{g}_l(z)u_l \qquad y = Cz$$

$u \equiv 0$

$$L_{g_l}T(x) = \underbrace{b_l + B_l T(x)}_{\text{KMs}} \qquad l = 1,..,m$$

$$||\Phi(z_1,u) - \Phi(z_2,u)|| \leq \gamma||z_1 - z_2|| \qquad \Phi(z,u) = \sum_{l=1}^{m} \tilde{g}_l(z)u_l$$

Linear

$$\dot{z} = Az \qquad y = Cz$$

Bilinear

$$\dot{z} = Az + \mathrm{B}_0\mathrm{u} + \sum_{l=1}^{m} (B_l z)u_l \qquad y = Cz$$

where,
$B_0 = [b_1 \dots b_l], \mathrm{u} = (u_1,..,u_l)^*$

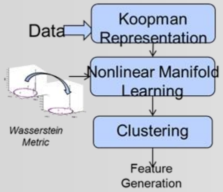
Liptchitz

$$\dot{z} = Az + \Phi(z,u) \qquad y = Cz$$

Higher dimensional, “simpler”

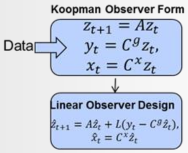
Machine Learning for Time Series Data

- *Forecasting/Anomaly detection*
- *Automatic feature generation*
- *Causal inference*



Data Driven System & Controls

- Nonlinear stability
- *Estimation/control*

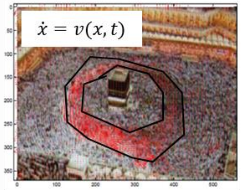


$$z_{t+1} = Az_t \qquad y_t = C^g z_t, \qquad x_t = C^x z_t$$

$$\hat{z}_{t+1} = A\hat{z}_t + L(y_t - C^g\hat{z}_t), \qquad \hat{x}_t = C^x\hat{z}_t$$

Signal Processing

- Background subtraction
- *Crowd video analysis*



- Introduced Koopman Observer Form
- Complements and readily combines with controls and machine learning techniques
- Data driven (model free), computable
- Amenable to scalable/cloud based/streaming computations
- Preliminary results promising, much more work needed

Thank you, Questions?