Modeling the Lorentz System and the Van der Pol Equation: Dynamic Mode Decomposition and Data Assimilation

Lourdes Coria, Horacio Lopez, Matteo Polimeno

San Diego State University

05/03/2018

Outline

- DMD
  - Background and theory
  - Usefulness
  - Lorentz Equation
  - Van der Pol Equation
- Data Assimilation
  - Background and theory
  - Lorentz Equation
  - Van der Pol Equation
- Conclusion

Motivation for Dynamic Mode Decomposition

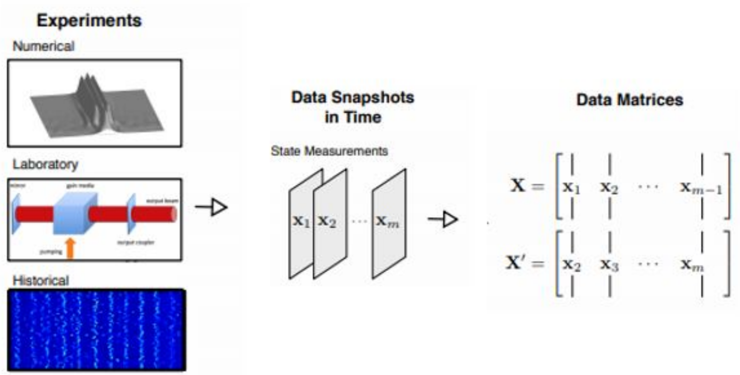
- The majority of model based algorithms have a given set of governing equations that determine the dynamics of a system
- What if we did not know or have the governing equations?
- New approach without equations

Dynamic Mode Decomposition

- Equation free modeling
- DMD is a powerful data-driven method for analyzing complex systems
- Uses data measurements from either numerical simulations or experiential results
- DMD extracts important dynamical characteristics

- Builds on the idea that DMD is able to characterize nonlinear dynamics through an analysis of some approximating linear system
- Theoretical results of DMD build on the theory of Krylov subspaces due to the original derivation of DMD as a variant of the Arnoldi method

Data Collection Process



Algorithm to compute DMD

- SVD-based algorithm has become the dominating algorithm among DMD users due to its numerical stability
- Data comes from a dynamical system whose evolution is defined as

$$x_{k+1} = Ax_k$$

- Relating the dynamical system to the data snapshots leads to

$$AX = X'$$

- Solving for A using the data matrices X and X' is the primary objective of DMD

Model Reduction Algorithm

Goal: Find the dynamic properties of A

$$X' = AX$$

1. Find the truncated SVD of X
2. Solve the dynamic modes A

- The SVD-based algorithm discuses the meaning of the DMD modes and eigenvalues in terms of a linear operator that satisfies
$$X' = AX$$
- Works if the trajectory of $\{x_k\}_{k=1}^{m}$ were generated from linear dynamics
- Works for nonlinear as well by approximating locally by a linear operator

Remarks

- DMD is relatively new
- Does not work for all cases
  1. If data matrix is rank-deficient
  2. In actuated systems DMD is incapable of producingan inpit-output model

Lorentz Equation

From [2], the Lorentz equations are given by

$$x' = \sigma(y - x) \tag{1}$$
$$y' = rx - y - xz \tag{2}$$
$$z' = xy - bz. \tag{3}$$

- We drive the system to chaos.
- Parameters: $\sigma = 10$, $b = 8/3$ and $r = 28$.
- Initial condition vector: $\mathbf{x}(0) = [5\ 5\ 5]$.
- The time step: Δt, will be 0.01 seconds.

Data Matrix Setup

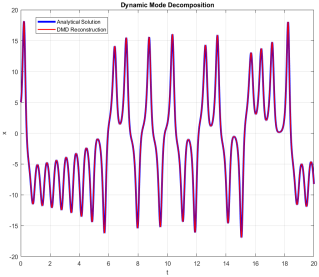
- Data from x solution of Lorentz system.
- Organized in a Hankel Matrix [6]:

$$\mathbf{H} = \begin{bmatrix} x(t_1) & x(t_2) & x(t_3) & \dots & x(t_p) \\ x(t_2) & x(t_3) & x(t_4) & \dots & x(t_{p+1}) \\ x(t_3) & x(t_4) & x(t_5) & \dots & x(t_{p+2}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x(t_q) & x(t_{q+1}) & x(t_{q+2}) & \dots & x(t_m) \end{bmatrix} \tag{4}$$

.

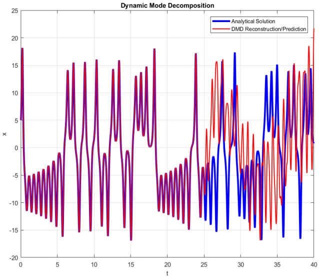
- DMD is used to both recreate the data matrix and predict future time states by extraction of the vector $x_m = [x(t_m)x(t_{m+1})...x(t_{m+M})]$ where M dictates how many time steps want to be solved for in the prediction.

Lorentz Results: Reconstructing Data



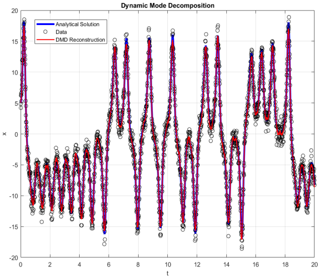
- DMD replicated the existing data accurately.
- Norm of the error vector is in the magnitude of $1e-12$.

Lorentz Results: Predicting Future States



- DMD fails at predicting future time steps without data.
- In order for DMD to work in chaotic systems, HAVOK analysis is used to identify the intermittently forced linear system representation of chaos [6].

Lorentz Result: Noise Reduction



- Select the dynamics we want to keep through eigenvalue selection and truncation at the time of taking the SVD of the data matrix [9].
- Cleared out high frequency dynamics generated through the random noise.
- Norm of the error vector between underlying system and DMD reconstruction was more or less 78 with small decimal variations.

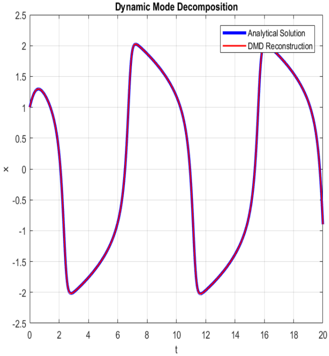
Van der Pol Equation

From [2], the Van der Pol equation is given by

$$x'' - \mu(1 - x^2)x' + x = 0 \tag{5}$$

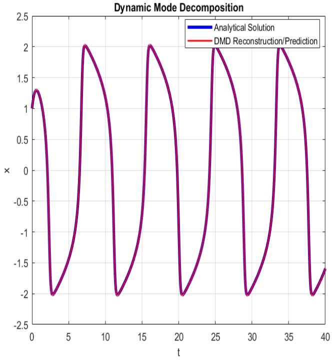
- We drive the system past hopf bifurcation.
- Parameter: $\mu = 3$.
- Initial condition vector: $\mathbf{x}(0) = [1\ 1]$.
- The time step: Δt, will be 0.01 seconds.

Van der Pol Results: Reconstructing Data



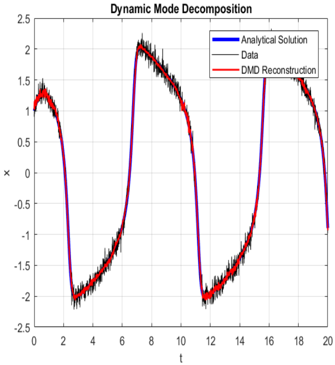
- DMD replicated the existing data accurately.
- Norm of the error vector is in the magnitude of $7e-11$.

Van der Pol Results: Predicting Future States



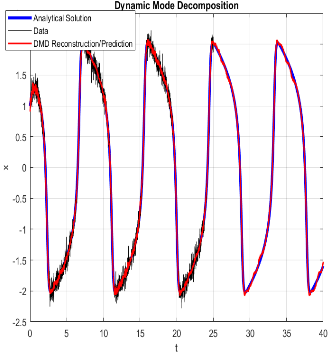
- DMD predicted future system values fairly accurately.
- Norm of the error vector is in the magnitude of 0.11.

Van der Pol Results: Reducing Noise



- DMD error: 1.3765

Van der Pol Results: Reducing Noise



- Does a good job at approximating future states
- Will eventually break down.

Data Assimilation

- Hybrid technique
- Unlike DMD, it makes use of governing equations
- Combines data collection and equations to improve forecast

Errors can often occur in the model

$$\frac{d\mathbf{x}}{dt} = f(t, \mathbf{x}) + \mathbf{q}_1(t) \tag{6}$$

in the initial conditions

$$\mathbf{x}(\mathbf{0}) = \mathbf{x}_0 + \mathbf{q}_2 \tag{7}$$

and/or in the data collected

$$g(t, \mathbf{x}) + \mathbf{q}_3 = 0 \tag{8}$$

We are going to look at how well the data assimilation algorithm reproduce the dynamics of the Lorentz system and the Van der Pol equation with perturbed initial conditions and noisy data

$$\mathbf{x}(0) = \mathbf{x_0} + \sigma_2 \mathbf{q}, \tag{9}$$

$$\mathbf{y}(t_n) = \mathbf{x}(t_n) + \sigma_3 \mathbf{q} \tag{10}$$

Only the x-dynamics is discussed in this presentation.

For a derivation of the algorithm, see [2]. The main idea is to compare a previous forecast with newly observations, then update the model to reflect those observations, so a new forecast is initiated and we can start over. The algorithm is illustrated as a pseudocode in the next slide

Data Assimilation Lorentz — Code

```
%solve system exactly
x0=[IC] %unperturbed initial conditions
%apply RK4 (ode45) to solve ode
%define solution
xtrue %true dynamics that data assimilation tries to
    reproduce
%perturb initial data
s2=1; %error variance in IC
x_IC = x0+s2*randn(1,3);
%solve with new perturbed initial data w/ ode45
x %perturbed solution
%generate random data and perturb
tdata %interval of data collection (every 0.5 steps in
    this case)
xn=randn(n,1);
s3=4; %error variance in data
xdata=x_true(1:50:end)+s3*xn;
%apply data assimilation
x_da=[]; %data assimilation solution
for j=1:length(tdata)-1 %step through every t=0.5;
```

Data Assimilation Algorithm - Pseudocode

```
    tspan=0:0.01:0.5; %time b/w data collection
    [tspan,xsol]=ode45('lor_rhs',tspan,x_IC,[],s,b,r);

    x_IC0=[xsol(end,1); xsol(end,2); xsol(end,3)]; %model
        estimate
    x_dat=[xdata(j+1); ydata(j+1); zdata(j+1)]; %data
        estimate
    K=s2/(s2+s3); %Kalman gain %s2=s3=1
    x_IC=x_IC0+K*(x_dat-x_IC0); %adjusted state vector

    x_da=[x_da; xsol(1:end-1,:)]; %store the data
end
x_da=[x_da; xsol(end,:)]; %store last data time
x_dax=x_da(:,1);
%plot
```

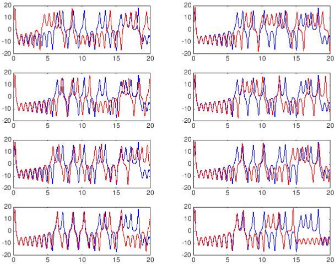


Figure 1: Unperturbed solution vs. perturbation of initial conditions with error variance $\sigma_2 = 1$

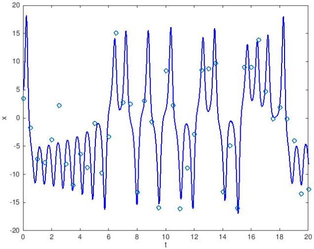


Figure 2: Noisy data with error variance $\sigma_3 = 4$

Data Assimilation Algorithm - Pseudocode

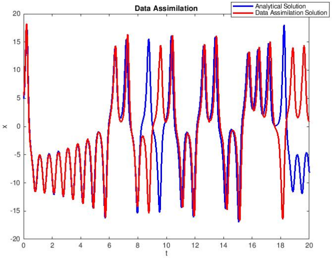


Figure 3: Data-assimilated solution with error variance $\sigma_2 = 1$ and $\sigma_3 = 4$

Van der Pol - Data Assimilation

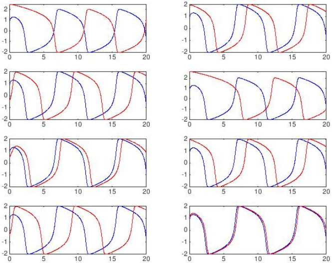


Figure 4: Perturbed initial conditions vs. True Dynamics

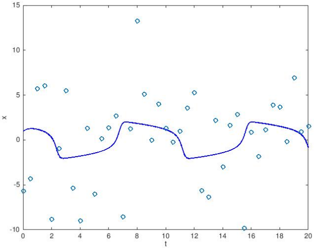


Figure 5: Noisy data with error variance $\sigma_3 = 4$

Van der Pol - Data Assimilation

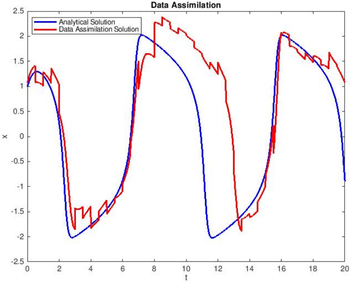


Figure 6: Model and data vs. Data-assimilated solution with error variance $\sigma_2 = 1$

Van der Pol - Data Assimilation

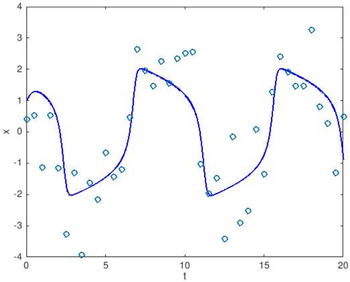


Figure 7: Noisy data $\sigma_3 = 1$

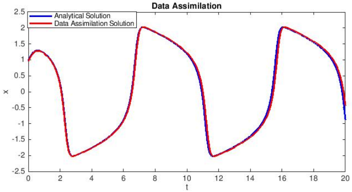


Figure 8: Data Assimilation solution for $\sigma_2 = 0.1$ and $\sigma_3 = 1$

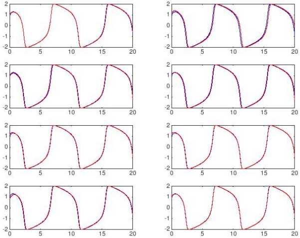


Figure 9: Perturbed initial conditions vs. true dynamics with $\sigma_2 = 0.1$

.

Questions?

- Grewal, M.S., Andrews, A.P. *Kalman Filtering: Theory and Practice Using MATLAB*. Wiley, 2008.
- J.Nathan Kutz. *Data-Driven Modeling & Scientific Computation Methods for Complex Systems and Big Data*. Oxford University Press, 2013.
- Joshua L. Proctor, Steven L. Brunton, J. Nathan Kutz. *Dynamic mode decomposition with control*. September 2014.
- Peter J. Schmit. *Dynamic Mode Decomposition of Numerical and Experimental Data*. Journal of Fluid Mechanics, vol. 656, 2010, pp. 5–28.

- Steven H. Strogatz. *Nonlinear Dynamics and Chaos, with applications to Physics, Biology, Chemistry and Engineering*. Westview Press, 2015
- Steven L. Brunton, Bingni W. Brunton, Joshua L. Proctor, Eurika Kaiser, J. Nathan Kutz, *Chaos as an intermittently forced linear system*. Nature Communications vol. 8, 2017
- Jonathan H. Tu *Dynamic Mode Decomposition: Theory and Applications*. Diss. Princeton University, 2013.
- J. Julier, Simon, K. Uhlmann, Jeffrey. *A New Extension of the Kalman Filter to Nonlinear Systems*. SPIE 3068, Signal Processing, Sensor Fusion, and Target Recognition VI, 28 July 1997.

Xin Liu, Zaiwen Wen and Yin Zhang *Limited Memory Block Krylov Subspace Optimization for Computing Dominant Singular Value Decompositions*. SIAM Journal on Scientific Computing, 35-3 (2013), A1641-A1668.