Dynamic Mode Decomposition

For Dynamic System Estimation

Daniel Lane
MA 506 – Fall 2022

Abstract

Modeling, simulation, and control design of complex systems through system identification can be challenging, especially when there is limited prior knowledge about the system. This is particularly relevant to traffic modeling, where group-level features arise from local interactions among vehicles. In this study we examine the effectiveness of two system estimation tools – Koopman with input and control as well as dynamic mode decomposition with control, to estimate a non-linear car following model's input response. Results reveal that both measures accurately predict the system response with varying levels of accuracy, with the main trade-off being computational efficiency.

Introduction

Dynamic system modeling is a complex task, especially when limited or no prior system information is available. There are many different methods of system identification available, though many of them require large amounts of data, and some require neural networks to train for hours. Dynamic mode decomposition with control, or DMDc, provides a very quick and easy way to estimate the system and its response to inputs. Its formulation comes as an extension of Koopman Operator theory, first proposed in 1931 – and reintroduced to the modern academic community by Schmidt et. all in 2010. Both methods give a closed form solution to a minimization problem, in much the same way as the basic form of linear regression does. While Koopman operator theory looks to precisely linearize system dynamics by performing a potentially infinite series of transformations on the state space, DMDc looks to find localized system dynamics, using as few dynamic modes as necessary to generate a reasonable estimation.

In this paper, these methods are used on data produced from synthetic traffic data. In truth, the originally proposed research was intended to find the full complex system dynamics of a multi-agent traffic system, though as research was conducted this was found to be an ill-conditioned problem for this particular methodology. Instead, Koopman with inputs and control and DMDc are used to estimate the input response of one agent from this system and compare it against the actual system dynamics.

Methodology

Dynamic Mode Deconstruction

Dynamic Mode Deconstruction (DMD) originated as a method for representing complex fluid models as simpler spatiotemporal structures [3]. It combines the methods of Proper Orthogonal Decomposition (POD) with Fourier transformations to characterize complex systems, state estimation and control. The method uses $\mathrm{t_k}$ "snapshots" of state measurement data $\mathbf{x}_\mathrm{k}$ and regresses the data onto linear dynamics similar to that of a state-space representation:

$$\mathbf{x}_{k+1} = A\mathbf{x}_k$$

Where $\mathbf{x}_\mathrm{k}$ is a matrix of state data measurements where columns are time snapshots from $k : m - 1$ and $\mathbf{x}_{\mathrm{k+1}}$ is a similar matrix containing snapshots from $k + 1 : m$ and A is calculated to minimize the error $\|\mathbf{x}_{\mathrm{k+1}} - \mathrm{A}\mathbf{x}_\mathrm{k}\|_2$ over the provided data [1]. Given the limited scope of provided system information, the resultant future state estimation is only locally valid. However, given the simplicity of the algorithm, it can be regenerated very quickly for any time window desired, given there is sufficient data.

If we consider a continuous-time initial value problem, with the initial condition $\mathbf{x}(0)$, and known solution:

$$\frac{d\mathbf{x}}{dt} = \mathcal{A}\mathbf{x}$$

$$\mathbf{x}(t) = \sum_{k=1}^{n} \boldsymbol{\phi}_k e^{(\omega_k t)} b_k = \mathbf{\Phi} \boldsymbol{e}^{\mathbf{\Omega} t}\mathbf{b}$$

(where $\boldsymbol{\phi}_k$ and ω_k are the eigenvectors and eigenvalues of the continuous-time dynamics matrix $\mathcal{A}$, and b_k are the coefficients associated with the eigenvector basis), we can describe a discrete-time series system by sampling at intervals of Δt:

$$\mathbf{x}_{k+1} = A\mathbf{x}_k$$

Where

$$A = e^{\mathcal{A}\Delta t}$$

whose solution is given by the eigenvalues λ_k and eigen vectors $\boldsymbol{\phi}_k$ from the singular value decomposition of the discrete-time dynamics A:

$$\mathbf{x}_k = \sum_{j=1}^{r} \boldsymbol{\phi}_j \lambda_j^k b_j = \mathbf{\Phi}\mathbf{\Lambda}^k\mathbf{b}.$$

This eigen decomposition of the matrix A is a least-squares optimized solution for the measured trajectory $\mathbf{x}_k$ for $k = 1,2, \dots, m$.

The strength of the DMD method is the fact that the dynamics of the system described by A does not need to be known and can be calculated directly from the state measurements (and control input). If we consider a non-linear system with n states and m snapshots, we construct 2 matrices of size $n \times (m - 1)$:

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_{m-1} \\ | & | & & | \end{bmatrix},$$

$$\mathbf{X}' = \begin{bmatrix} | & | & & | \\ \mathbf{x_2} & \mathbf{x}_3 & \dots & \mathbf{x}_m \\ | & | & & | \end{bmatrix}$$

Then we can give a locally approximation of our non-linear system,

$$\mathbf{X}' \approx A\mathbf{X}$$

Where we estimate the matrix A with a Moore-Penrose pseudoinverse of $\mathbf{X}$ **:**

$$(1).\, Take\ the\ SVD\ of\ X: \qquad \mathbf{X} \approx \mathbf{U\Sigma V}^*$$
$$(2).\, The\ psudoinverse\ of X: \quad \mathbf{X}^\dagger = \mathbf{V\Sigma^{-1}U}^*$$
$$(3).\, Then: \qquad A = \mathbf{X}'\mathbf{X}^\dagger = \mathbf{X}'\mathbf{V\Sigma^{-1}U}^*$$

This solution minimizes the Frobenious norm $\| \, . \, \|_F$:

$$\|\mathbf{X}' - A\mathbf{X}\|_F$$

$$\| \mathbf{X} \|_F = \sqrt{\sum_{j=1}^{n} \sum_{k=1}^{m} X_{jk}^2}$$

For higher dimensional systems, an approximation $A \approx \tilde{A}$ can be obtained of size of $r \times r$, where r represents the number of eigenvalues whose absolute value is above some defined threshold by projecting A onto POD modes:

$$\widetilde{\mathrm{A}} = \mathbf{U}^*\mathbf{AU} = \mathbf{U}^*\mathbf{X}'\mathbf{V\Sigma}^{-1}$$

Now we can obtain the eigen decomposition of the matrix A (or $\widetilde{\mathrm{A}}$):

$$\widetilde{\mathrm{A}}\mathbf{W} = \mathbf{W\Lambda}$$

Where $\mathbf{W}$ is constructed with columns of eigenvectors, and $\mathbf{\Lambda}$ is a diagonal matrix with the eigenvalues λ_k. And from these eigen values and eigenvectors of $\widetilde{\mathrm{A}}$, we can obtain the DMD modes of the system:

$$\mathbf{\Phi} = \mathbf{X}'\mathbf{V\Sigma}^{-1}\mathbf{W}$$

compute the DMD mode coefficients using the initial conditions provided in the snapshot $\mathbf{x_1}$:

$$\mathbf{b} = \mathbf{\Phi}^\dagger\mathbf{x}_1$$

reconstruct the continuous-time eigenvalues:

$$\omega = ln(\lambda_k)/\Delta t$$

and finally produce an estimated trajectory:

$$\mathbf{X}_{DMD} = \mathbf{\Phi} e^{\omega t}$$

For the case of systems that utilize a control input**:** $\mathbf{x}_{k+1} = A\mathbf{x}_k + B\boldsymbol{u}_k$, we conveniently can just augment the $\mathbf{X}'$ matrix with q rows of input measurements $\boldsymbol{u}_k$ over the same time snapshots and along with to the matrix $\mathbf{U}^* = [\mathbf{U_1 U_2}]$, where $\mathbf{U_1} \in \mathbb{R}^{n \times r}$ and $\mathbf{U_2} \in \mathbb{R}^{q \times r}$ (the first $n + q$ columns of $\mathbf{U}^*$ from the decomposition of $\mathbf{X}$) to obtain use the equivalent of $\mathbf{\Phi}$ and the A and B matricies [1]:

$$A \approx \mathbf{X}'\mathbf{V\Sigma}^{-1}\mathbf{U_1}$$
$$B \approx \mathbf{X}'\mathbf{V\Sigma}^{-1}\mathbf{U_2}$$
$$\mathbf{\Phi} = \mathbf{X}'\mathbf{V\Sigma}^{-1}\mathbf{U_1}^*\mathbf{UW}$$

Koopman with Inputs and Control

Koopman with Inputs and Control (KIC) is another data-driven method of estimating non-linear systems: $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \boldsymbol{u}_k)$. Generally with Koopman operator theory, an explicit transformation is performed on the state variables to obtain a linear approximation of the system (often infinitely dimensional as closed systems are dependent on the non-linearity and rarely obtainable).

$$\mathcal{K}g(\mathbf{x},\mathbf{u}) \triangleq g(\mathbf{f}(\mathbf{x},\mathbf{u}),\mathbf{0})$$

Where each Koopman operator acts on the Hilbert space of observable functions $\mathcal{K}: \mathcal{H} \to \mathcal{H}$, and each function $g(\mathbf{x},\mathbf{u})$ is an element of infinite-dimensional Hilbert space $\mathcal{H}$ [3].
The linearity of this new function space allows for eigen decomposition:

$$\mathcal{K}\varphi_j(\mathbf{x},\mathbf{u}) = \lambda_j\varphi_j(\mathbf{x},\mathbf{u}), j = 1,2, \ldots$$

Where $\mathcal{K}$ is now spanned by eigenfunctions dependent on the inputs $\mathbf{u}$ and $\mathbf{x}$, and the functions g_i can be given in terms of the eigenvalues [2]:

$$\mathbf{g}(\mathbf{x},\mathbf{u}) = \begin{bmatrix} g_1(\mathbf{x},\mathbf{u}) \\ g_2(\mathbf{x},\mathbf{u}) \\ \vdots \\ g_p(\mathbf{x},\mathbf{u}) \end{bmatrix} = \sum_{j=1}^{\infty} \varphi_j(\mathbf{x},\mathbf{u})\mathbf{v}_j$$

$$\mathcal{K}\mathbf{g}(\mathbf{x},\mathbf{u}) = \mathbf{g}(\mathbf{f}(\mathbf{x},\mathbf{u}),\mathbf{0}) = \sum_{j=1}^{\infty} \lambda_j\varphi_j(\mathbf{x},\mathbf{u})\mathbf{v}_j$$

However, we wish to solve for this without knowing the dynamics *a priori*. Instead, we will assume a linearization using a polynomial of order 2 on our state and input variables, which will stand in for the functions g_i, and transforms our states and inputs $\mathbf{x}_k$ and $\mathbf{u}_k$ to a new coordinate system $\mathbf{z}_k$**,** and we can consider the same system as before:

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\boldsymbol{u}_k$$

And use KIC to linearize our system:

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{u}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} \\ \mathbf{G}_{21} & \mathbf{G}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix}$$

$$\mathbf{z}_{k+1} = \mathrm{G}\mathbf{z}_k$$

As we do not care what the predicted control state is at t_{k+1}, we only need to care about the rows of the resultant matrix G that correspond to the state predictions, and reduce our equations:

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{u}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix}$$

$$[\mathbf{x}_{k+1}] = [\mathbf{G}_{11} \quad \mathbf{G}_{12}] \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix}$$

$$\mathbf{x}_{k+1} = \mathrm{G}\mathbf{z}_k$$

This system is now similar to the one described for DMD described in the section above, and the same methodology is used to predict future state trajectories [1].

Optimal Velocity Model

The Optimal Velocity Model (OVM) is a car-following traffic model which predicts the Optimal Velocity (OV) of a given car as a function of the headway h between the target vehicle and its proximally front car:

$$\dot{x_1} = x_2$$
$$\dot{x_2} = a[V(h) - x_2]$$

$$V(h) = \alpha \tanh[\beta(h - s_0)] + v_o$$
$$h = x_1^{k+1} - x_1^k$$

where x_1 and x_2 are the position and velocity of the target vehicle, and $\alpha, \beta, a, s_0, v_0$ are all constants determined from empirical data [6].

To deal with the non-linearity of the hyperbolic tangent function, a Pade' (2,2) approximation of the function is used:

$$tanh(x) \approx \frac{3x}{x^2 + 3}, \text{ on the domain } [-\pi/2, \pi/2]$$

Which alters our OV function:

$$V(h) = \alpha \frac{3\beta\left((x_1^{k+1} - x_1^k - s_0\right)}{\left(\beta(x_1^{k+1} - x_1^k - s_0)\right)^2 + 3} + v_o$$

which is almost consistent with the order-2 polynomial chosen for the KIC linearization performed in the previous section. To deal with the denominator terms which effectively have a negative power, we will include $\dot{x_2}$ terms when considering the permutations for $\mathcal{K}\mathbf{g}(\,.\,)$:

$$\mathcal{K}\mathbf{g}(\mathrm{x}_1, x_2, \dot{\mathrm{x}}_2, \mathbf{u})$$

This approach is consistent with similar methods that deal with rational functions by using their implicit formulations [7].

Results

The methods of DMD with control and KIC were examined for a system modeled with OVM dynamics, using a custom sinusoidal forcing function as the input control $\boldsymbol{u}$. The methods described above were then applied to obtain a trajectory estimation for the same set of inputs.

For DMD estimation, the root-mean-squared error (RMSe) for state $x_1$was 0.38793, and 0.34773 for x_2 over a trajectory of eighty seconds.

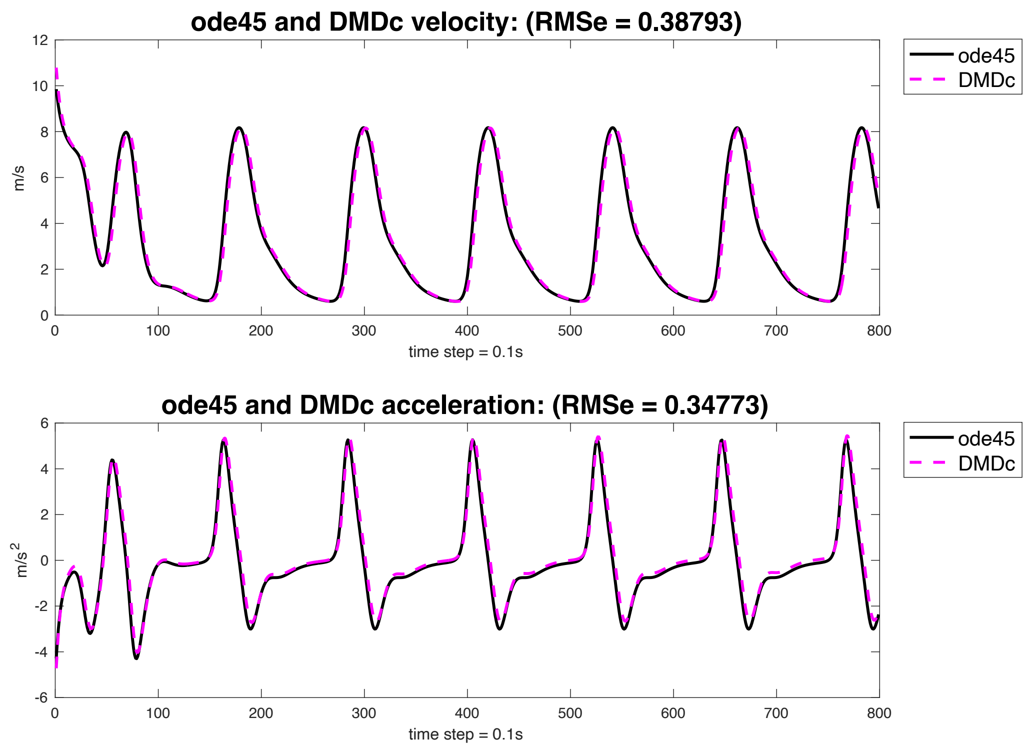

Fig. 1 – DMDc estimation for the same forcing inputs $\boldsymbol{u}$. 800 time steps of 0.1s for a total of 80 seconds

It can be seen from figure 1 that estimation worsened as time evolved, which is consistent with DMD as it is described, most easily seen in the acceleration trajectory around the 700th time step.

For KIC estimation, trajectories were found with 4 sets of functions g_i. The first set uses all possible combinations for the proposed functional inputs $\mathcal{K}\mathbf{g}(\mathrm{x}_1, x_2, \dot{\mathrm{x}}_2, \mathbf{u})$. This resulted in a $\mathbf{z}_k$ large matrix with 54 rows. Expanding the Pade' 22 estimation of the dynamics shows that sparse set of only 17 of those 54 terms are required to model the equation successfully. Therefore, this knowledge led to the desire to test the efficacy of KIC using 3 smaller subsets of the function space. The first was to eliminate $\dot{\mathrm{x}}_2$ from the input, resulting in $\mathbf{z}_k$ with 27 rows. Next, I removed the permutations not necessary to describe the dynamics (again without the $\dot{\mathrm{x}}_2$ permutation), resulting in 12 rows. For the last permutation, I added the $\dot{\mathrm{x}}_2$ term back to those 12 terms to get 24 rows.

The resultant trajectories all showed a very close match to the system solved with full dynamics using Runga-Kutta.

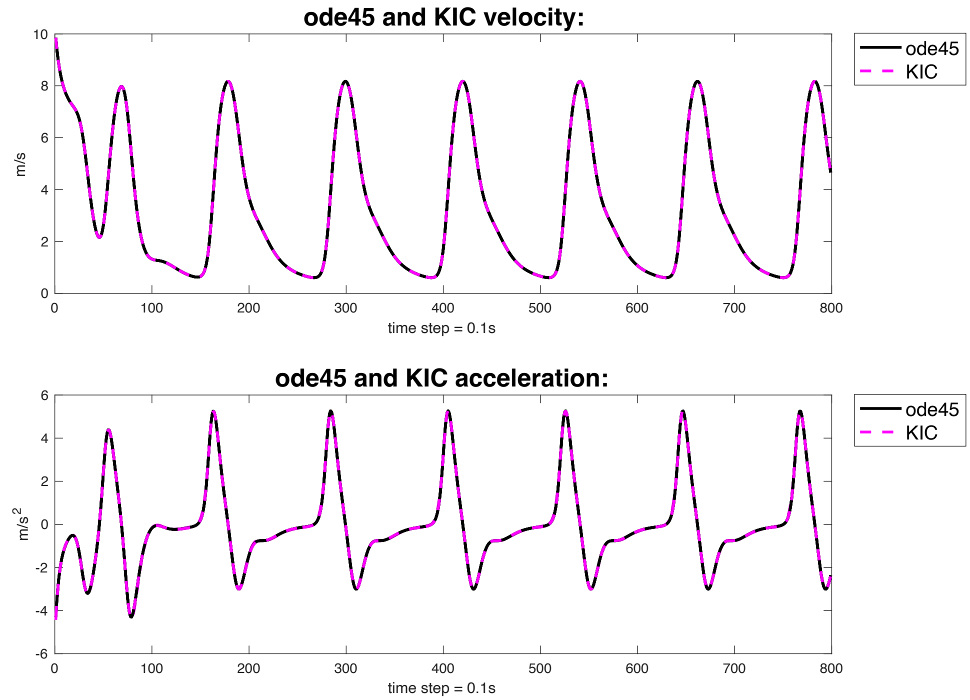

Fig. 2 – KIC and Runga-Kutta estimation for the same forcing inputs $\boldsymbol{u}$. Plots from all 4 formations of KIC are graphically identical, so only the result from the full KIC 54-row estimation is shown.

Z terms	x_1 RMSe	x_2 RMSe
12	2.200E+01	4.010E+02
24	4.431E-04	7.900E-03
27	8.776E-04	1.640E-02
54	3.928E-05	3.392E-04

Table. 1 – root mean squared estimation error from KIC future state estimation

The dynamic modes given in the resultant matrix $\boldsymbol{\Phi}$ show an interesting correlation to actual coefficients found from the Pade' 2,2 approximation of the OVM dynamics, though further research would be required to address the precise relation.

Discussion and Conclusion

Koopman operator theory and DMD provide reasonable system estimations when system dynamics are fully observable. KIC gives a much more accurate estimation, but with a cost. The system examined in this paper only had 2 state variables, and one input. For larger MIMO systems, like quad copters which can have as many as 9 states, and 4 inputs, this method quickly can become computationally expensive.

DMD with control gives a reasonable state estimation with the benefit that it is extremely fast to calculate. Given that the solution is only valid for small time windows, the simplicity and speed of the algorithm allows it to be continuously re-calculated if necessary for ongoing system estimation.

The approximation of A and B matrices provided by DMDc give opportunity to apply more advanced control structures such as LQR.

References

[1] Kutz, J. N., Brunton, S. L., Brunton, B. W., & Proctor, J. L. (2016). *Dynamic mode decomposition: data-driven modeling of complex systems*. Society for Industrial and Applied Mathematics.

[2] Proctor, J. L., Brunton, S. L., & Kutz, J. N. (2016). Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1), 142-161.

[3] Brunton, S. L., Budišić, M., Kaiser, E., & Kutz, J. N. (2021). Modern Koopman theory for dynamical systems. *arXiv preprint arXiv:2102.12086*.

[4] Schmid, P. J. (2010). Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656, 5-28.

[5] Koopman, B. O. (1931). Hamiltonian systems and transformation in Hilbert space. *Proceedings of the National Academy of Sciences*, 17(5), 315-318.

[6] Nakayama, A., Kikuchi, M., Shibata, A., Sugiyama, Y., Tadaki, S. I., & Yukawa, S. (2016). Quantitative explanation of circuit experiments and real traffic using the optimal velocity model. *New Journal of Physics*, 18(4), 043040.

[7] Kaheman, K., Kutz, J. N., & Brunton, S. L. (2020). SINDy-PI: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics. *Proceedings of the Royal Society A*, 476(2242), 20200279.