Gaussian Process Dynamic Modeling of Bat Flapping Flight

The First Nonlinear Dimensionality Reduction of Bat Flight

Matt Bender[1], Xu Yang[2], Hui Chen[2], Andrew Kurdila[1], & Rolf Müller[1,3]

[1] Department of Mechanical Engineering, Virginia Tech

[2] School of Information Science and Engineering, Shandong University

[3] Shandong University - Virginia Tech International Lab, Jinan, China

Invent the Future

Abstract

- Bats can serve as inspiration for flapping-wing air vehicles.
- Understanding bat flight requires detailed kinematics data.
- We use *many low-cost,low frame rate* cameras which results in nonlinear, large-baseline motions in image space.
- We apply Gaussian Process Dynamic Modeling (GPDM) to learn a low dimensional model of bat flight.
- **This paper presents the first nonlinear dimensionality reduction of bat flight.**

Introduction

- Bats are very agile, stable, and efficient which makes them excellent models for air vehicle design.
- Their flight characteristics are due to an articulated skeleton and deformable wing membrane which are difficult to study via imaging due to frequent incidents of self-occlusion.
- Occlusions can be avoided by using many low-cost/low-frame rate cameras.
- The low frame rates result in large-baseline nonlinear motions of points in the image which makes correspondence difficult.
- In [**6**], the authors bootstrap optical flow with a Square Root Unscented Kalman Filter to perform tracking which works well when videos are captured at frame rates >250fps.
- Our experiments are conducted with much lower frame rate cameras (120fps), so a better motion model is needed.
- We use Gaussian Process Dynamic Models (GPDM) presented in [**8-10**] to learn a low-dimensional *nonlinear* manifold to represent bat flight.
- **We learn a dynamic model of bat flight which can be used as a more accurate motion prior than random walk.**
- **This is the first nonlinear dimensionality reduction of bat flight.**

Experimental Facility

Figure 1: Flight Tunnel. 28 GoPro Hero 3+ Cameras running at 720p 120fps

Gaussian Process Dynamic Models

Given a sequence of data $\mathbf{Y} := \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_K\}$ which is assumed to be zero mean along each dimension where $\boldsymbol{y}_k \in \mathbb{R}^N \quad \forall k \in [1, 2, ..., K]$, we want to determine the model

$$\boldsymbol{x}_k = \boldsymbol{f}(\boldsymbol{x}_{k-1}, \mathrm{A}) + \boldsymbol{n}_{x,k} \qquad \boldsymbol{y}_k = \boldsymbol{g}(\boldsymbol{x}_k, \mathrm{B}) + \boldsymbol{n}_{y,k} \tag{1}$$

where, $\boldsymbol{x}_k \in \mathbb{R}^D$ is a D-dimensional latent space which supports the dynamics, $\boldsymbol{f}$ is a nonlinear dynamic model, $\boldsymbol{g}$ projects latent states into state space, and A and B are matrices of linear parameters. We assume that

$$\boldsymbol{f}(\boldsymbol{x}, \mathrm{A}) = \sum_i \boldsymbol{a}_i \phi_i(\boldsymbol{x}), \qquad \boldsymbol{g}(\boldsymbol{x}, \mathrm{B}) = \sum_j \boldsymbol{b}_j \psi_j(\boldsymbol{x}), \tag{2}$$

where $\mathrm{A} := [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots]$ and $\mathrm{B} := [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots]$ are matrices of scaling parameters and $\phi(\boldsymbol{x})$ and $\psi(\boldsymbol{x})$ are scalar output functions.

Authors in [**8-10**] formulate the joint probability of the latent variables $\mathbf{X}$, collected data $\mathbf{Y}$, hyper-parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, and a weighting matrix $\mathbf{W}$ as

$$p(\mathbf{Y}, \mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{W}) = p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\beta}, \mathbf{W}) p(\mathbf{X}|\boldsymbol{\alpha}) p(\boldsymbol{\alpha}) p(\boldsymbol{\beta}) p(\mathbf{W}). \tag{3}$$

The first term is obtained by placing a Gaussian prior on the columns of B and marginalizing $\boldsymbol{g}$. This eliminates the dependency of the model on the linear parameters, and only hyper-parameters must be identified. The marginalization can be done in closed form [**15**] to obtain

$$p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\beta}, \mathbf{W}) = \frac{|\mathbf{W}|^K}{\sqrt{(2\pi)^{KN} |\mathbf{K}_y|^N}} \exp\left(-\frac{1}{2}\mathrm{trace}\left(\mathbf{K}_y \mathbf{Y}\mathbf{W}^2\mathbf{Y}^T\right)\right), \tag{4}$$

where $\mathbf{W} := \mathrm{diag}([w_1, w_2, ... w_N])$ serves to weight $\mathbf{Y}$ so that dimensions which are large in magnitude do not dominate the optimization. Additionally, $\mathbf{K}_y$ is the covariance kernel with hyper-parameters $\boldsymbol{\beta}$. This matrix is assembled as

$$(\mathbf{K}_y)_{ij} = \beta_1 \exp\left(-\frac{\beta_2}{2}||\boldsymbol{x}_i - \boldsymbol{x}_j||^2\right) + \beta_3^{-1}\delta(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{5}$$

The second term on the right hand side of equation 3 describes the dynamics on the latent space. This probability is formulated as

$$p(\mathbf{X}|\boldsymbol{\alpha}) = \int p(\mathbf{X}|\mathrm{A}, \boldsymbol{\alpha}) p(\mathrm{A}|\boldsymbol{\alpha}) d\mathrm{A}. \tag{6}$$

Formulating the probability distribution in this manner removes the dependency on the parameters in $\mathbf{A}$ with the smaller set of hyper-parameters contained in $\boldsymbol{\alpha}$. Assuming a Gaussian prior on A, and assuming the process is Markovian, this probability can also be computed in closed form as

$$p(\mathbf{X}|\boldsymbol{\alpha}) = \frac{p(\boldsymbol{x}_1)}{\sqrt{(2\pi)^{(K-1)D} |\mathbf{K}_x|^D}} \exp\left(-\frac{1}{2}\mathrm{trace}\left(\mathbf{K}_x^{-1}\mathbf{X}_{2:K}\mathbf{X}_{2:K}^T\right)\right). \tag{7}$$

Note that the weighting matrix is not required here because the latent variables are nondimensional. To define the covariance function we choose the linear plus radial basis function kernel

$$(\mathbf{K}_x)_{ij} = \alpha_1 \exp\left(-\frac{\alpha_2}{2}||\boldsymbol{x}_i - \boldsymbol{x}_j||^2\right) + \alpha_3 \boldsymbol{x}_i^T \boldsymbol{x}_j + \alpha_4^{-1}\delta(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{8}$$

The only terms that remain in the joint distribution in equation 3 are the priors on $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. These serve to constrain the hyper-parameters and the weight matrix $\mathbf{W}$.

The joint probability distribution can be maximized by minimizing the negative log likelihood of the distribution

$$\begin{aligned} L = &\frac{D}{2}\ln||\mathbf{K}_x|| + \frac{N}{2}\ln||\mathbf{K}_y|| - K\ln|\mathbf{W}| + \frac{1}{2}\mathrm{trace}(\mathbf{K}_x^{-1}\mathbf{X}_{2:K}\mathbf{X}_{2:K}^T) + \\ &\frac{1}{2}\boldsymbol{x}_1^T\boldsymbol{x}_1 + \frac{1}{2}\mathrm{trace}(\mathbf{K}_y^{-1}\mathbf{Y}\mathbf{W}^2\mathbf{Y}^T) + \sum_j \ln(\beta_j) + \\ &\frac{1}{2\kappa^2}\mathrm{trace}(\mathbf{W}^2) + \sum_j \ln\alpha_j. \end{aligned} \tag{9}$$

There are many proposed methods for solving this equation: maximum a posteriori (MAP) estimation, Balanced GPDM methods, manually specify hyper-parameters, or two stage MAP. We use two stage MAP estimation where the first stage optimizes the weight matrix $\mathbf{W}$ and the second stage optimizes $\mathbf{X}$, $\boldsymbol{\alpha}$, and $\boldsymbol{\beta}$.

After optimizing, we propagate dynamics in the learned latent space using the distribution

$$\mu_X(\boldsymbol{x}) = \mathbf{X}_{out}^T \mathbf{K}_x^{-1} \boldsymbol{k}_X(\boldsymbol{x}), \tag{10}$$

$$\sigma_X^2(\boldsymbol{x}) = k_X(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}_X(\boldsymbol{x})^T \mathbf{K}_x^{-1} \boldsymbol{k}_X(\boldsymbol{x}). \tag{11}$$

These dynamics can be used for simulation of trajectories in the latent space and compared to those identified from the experimental data. Once trajectories have been simulated in latent space, they can be projected into feature space to generate simulated motions similar to the training data.

To project latent trajectories back into feature space, we can write

$$\mu_Y(\boldsymbol{x}) = \mathbf{Y}^T \mathbf{K}_y^{-1} \boldsymbol{k}_Y(\boldsymbol{x}), \tag{12}$$

$$\sigma_Y^2(\boldsymbol{x}) = k_Y(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}_Y(\boldsymbol{x})^T \mathbf{K}_y^{-1} \boldsymbol{k}_Y(\boldsymbol{x}). \tag{13}$$

Note that the projected trajectories will be zero mean due to the subtraction of the mean applied to the original data.

Results

Latent Space Trajectories

- The identified manifolds are shown in Figure 2.
- All manifolds use the RBF plus linear kernel from Equation 8 on the dynamics and the RBF kernel in equation 5 for the mapping between latent space and feature space.
- The subfigures show GPDMs identified with 2, 3, and 4 latent DOFs, respectively.
- The blue points are the identified latent trajectories, red points are the simulated trajectories, and green points are HMC samples on the simulated trajectories to represent the uncertainty in the manifold.
- Arrows indicate the direction of motion.
- Ideally, the manifold should appear to be cyclic and have little variation between cycles.

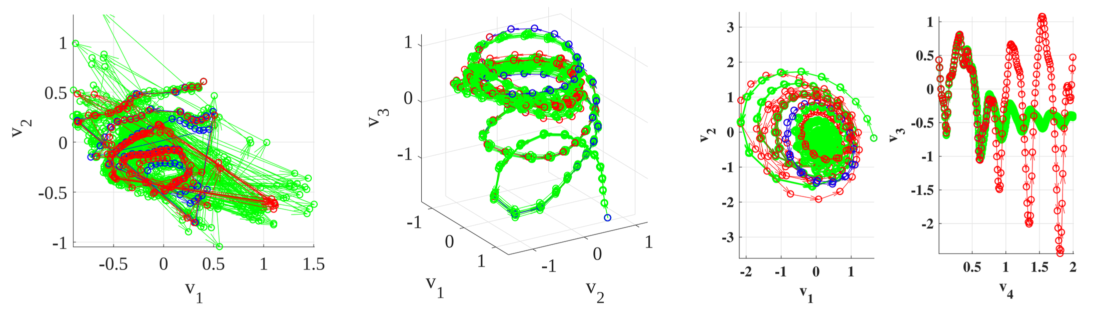

Figure 2: Left to Right: 2 DOF, 3DOF, and 4 DOF Latent Spaces

- To evaluate the consistency of the manifold we project it back into feature space which is shown in Figure 3.
- The predictions (solid line) match the experimental data (dots) very closely.
- Note that the dark blue cycles for $\theta_1 - \theta_3$ are larger in amplitude than the remaining cyan cycles.
- This behavior is likely the cause of the semi-periodic latent space trajectories discussed before.
- The extrapolated data (magenta) appears to be periodic repetitions of the cyan portion of the experimental data.

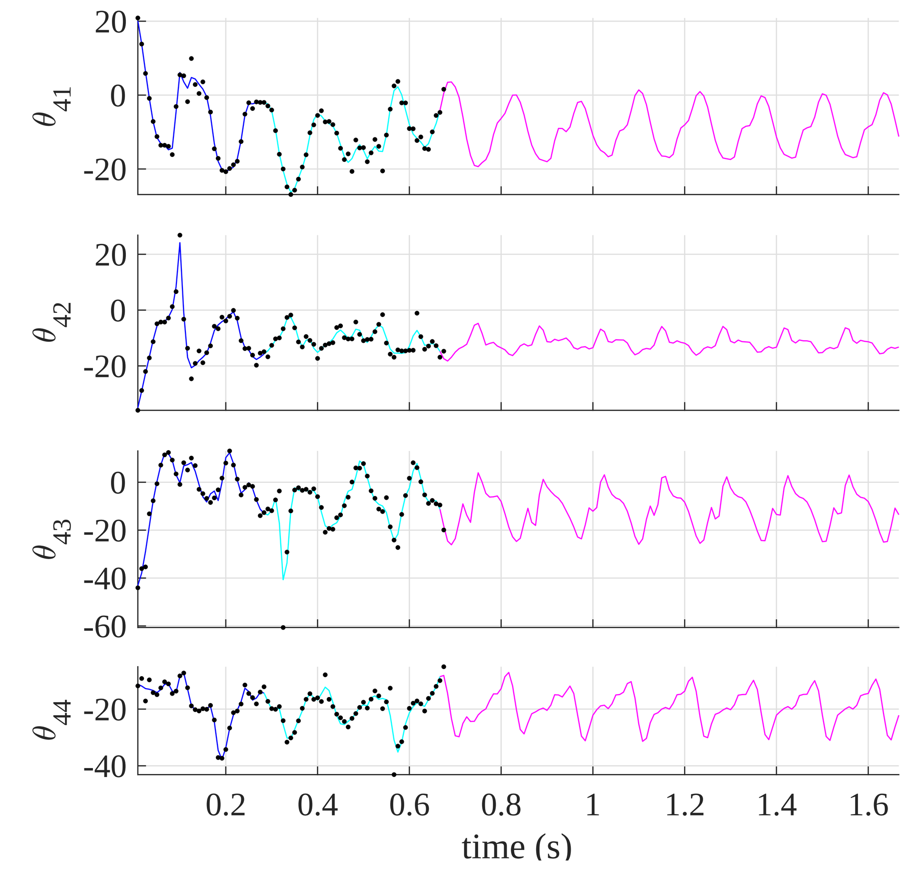

Figure 3: Feature Space Projection of Latent Trajectories.

Conclusions

- We use Gaussian Process Dynamic Models (GPDM) to learn both a latent space representation for bat flight dynamics and a mapping from latent space back to joint space.
- This is—to the authors' knowledge—the first nonlinear dimensionality reduction of bat flapping flight.
- We have successfully identified a model which closely resembles the experimental data provided and produces plausible synthesized motions.
- In the future, this model will be used as a motion prior to enable automated tracking of feature points.

Acknowledgements

The authors would like to acknowledge the following funding sources for their support: National Natural Science Foundation of China (grant numbers 11374192 and 11574183); Fundamental Research Fund of Shandong University (grant no. 2014QY008); Minister of Education of China Tese grant for faculty exchange; U.S. National Science Foundation (grant no. 1510797); and Virginia Tech Institute for Critical Technology and Applied Science (ICTAS, through support for the BIST Center).

The authors would also like to thank the following individuals for their support in data collection and processing: Li Tian, Xiaozhou Fan, Chenhao Wang, Mengfan Wang, Yuxian Ye, Yuxiang Zhu, Xizhe Ding, Yang Shao, Yiwei Jiang, Kefan Chen, Yanan Zhao, Derek Liu.