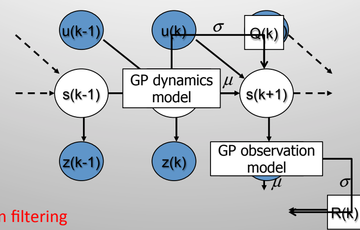
CSE-571
Probabilistic Robotics

Gaussian Processes for Bayesian Filtering

Dieter Fox
University of Washington

[Ko-F: RSS-08, ARJ-09]

GP-BayesFilters



- Bayesian filtering
  – Parametric dynamics and observation models
  – Approximate posterior via sampling (PF), sigma points (UKF), linearization (EKF), moment matching (ADF)
- GP-BayesFilters
  – Use Gaussian Process regression to learn dynamics and observation models
  – Noise derived from GP prediction uncertainty
  – Can be integrated seamlessly into Bayes filters: EKF, UKF, PF, ADF

Overview

- Gaussian Processes and Bayes Filters
- GP-BayesFilters
- Filtering and Control
- System Identification with GP-BayesFilters
- Predictive State Representations
- Conclusions

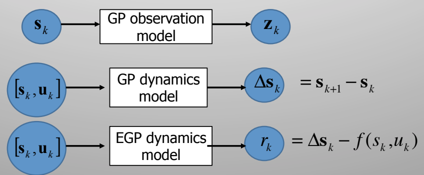
[Ko-Fox: ARJ-09]

GP-BayesFilters

- Learn GP:
  – Input: Sequence of ground truth states along with controls and observations: <s, u, z>
  – Learn GPs for dynamics and observation models
- Filters
  – Particle filter: sample from dynamics GP, weigh by GP observation function
  – EKF: GP for mean state, GP derivative for linearization
  – UKF: GP for sigma points

Learning GP Dynamics and Observation Models

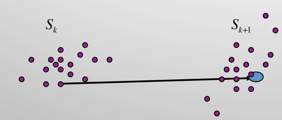
- Ground truth training sequence:
  $S=[s_1,s_2,\ldots,s_n], Z=[z_1,z_2,\ldots,z_n], U=[u_1,u_2,\ldots,u_n]$
- Learn observation and dynamics GPs:



- Learn separate GP for each output dimension
- Diagonal noise matrix

[Deisenroth-etal] introduced GP-ADFs and EP for smoothing in GP dynamical systems

GP-PF Propagation

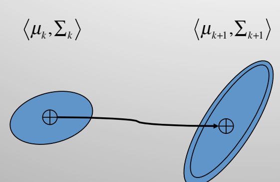


for $m=1\ldots M$:

$s_{k+1}^m = GP_\mu(s_k^m,u) + sample(GP_\Sigma(s_k^m))$

- Propagate each particle using GP prediction
- Sample from GP uncertainty
- One GP mean and variance prediction per particle

GP-EKF Propagation



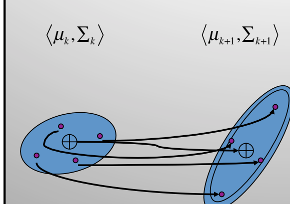
$\mu_{k+1} = GP_\mu(\mu_k)$

$G = \frac{\partial GP_\mu(\mu_k)}{\partial s}$

$\Sigma_{k+1} = G\,\Sigma_k\,G^T + GP_\Sigma(\mu_k)$

- Propagate mean using GP prediction
- Use gradient of GP to propagate covariance

GP-UKF Propagation



$\chi_k = \left(\mu_k, \quad \mu_k + \gamma\sqrt{\Sigma_k}, \quad \mu_k - \gamma\sqrt{\Sigma_k} \right)$

for $i=0\ldots 2n: \chi_{k+1} = GP_\mu(\chi_k)$

$\mu_{k+1} = \sum_{i=0}^{2n} \omega_m^i \chi_{k+1}^i$

$\Sigma_{k+1} = \sum_{i=0}^{2n} \omega_c^i (\chi_{k+1}^i - \mu_{k+1})(\chi_{k+1}^i - \mu_{k+1})^T + GP_\Sigma(\mu_k)$

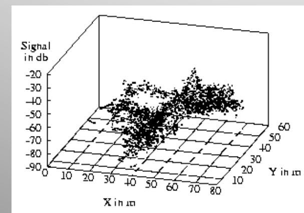
- Propagate each sigma point using GP prediction
- 2d+1 sigma points -> 2d+1 GP mean predictions

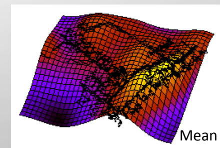
Overview

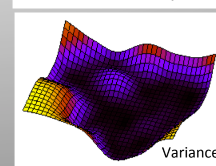
- Gaussian Processes and Bayes Filters
- GP-BayesFilters
- **Filtering and Control**
- System Identification with GP-BayesFilters
- Predictive State Representations
- Conclusions

[Ferris-Haehnel-Fox: RSS-06]

WiFi-Based Location Estimation

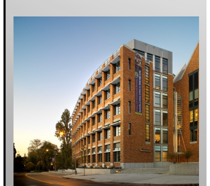


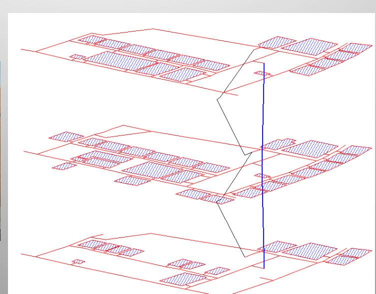




Similar to [Schwaighofer-etal: NIPS-03]

Building Model

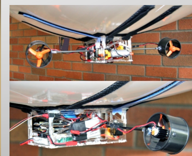




Tracking Example

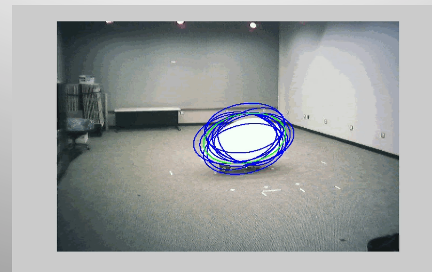
[Ko-F: ARJ-09]

Blimp Testbed



- Task: Track a blimp with two webcams
- Baseline: Parametric model that takes drag, thrust, gravity, etc, into account
- GP-BayesFilters and parametric model trained on ground truth data obtained with Vicon motion capture system

GP-UKF Tracking Example



- Blue ellipses: sigma points projected into observation space
- Green ellipse: Mean state estimate

Tracking Results

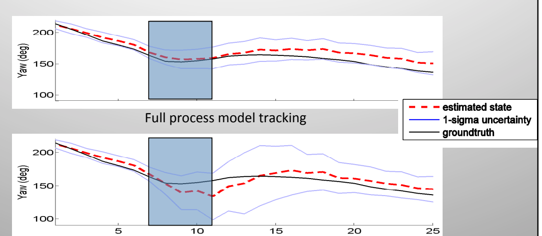
	GP	*EGP*	*hetGP*	*sparseGP*
UKF	30.75 ± 1.41	34.10 ± 1.76	35.76 ± 1.61	32.05 ± 2.02
EKF	27.66 ± 1.04	31.44 ± 2.43	33.70 ± 2.09	29.72 ± 1.90
PF	33.93 ± 7.24	35.95 ± 6.91	na	38.92 ± 2.17

Percentage reduction in RMS over parametric baseline

- Parametric model takes drag, thrust, gravity, etc, into account
- Cross validation with 900 timesteps for training

- hetGP: Heteroscedastic GP with variable noise [Kersting-etal: ICML-07]
- sparseGP: sparsified to 50 active points [Snelson-Ghahramani: NIPS-06]

Dealing with Training Data Sparsity

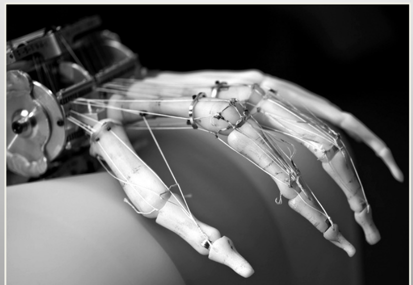


Full process model tracking

No right turn process model tracking

- Training data for right turns removed

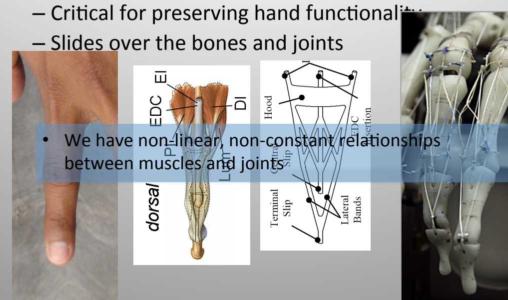
ACT Hand Control



1. Investigation of muscle-joint kinematical relationship
2. How to control joints with muscles?

ACT Hand Tendon Arrangements

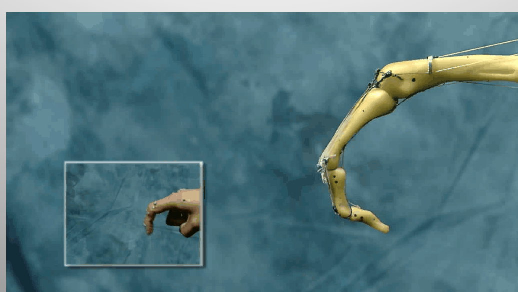
- Tendon hood structure for extensors
  - Critical for preserving hand functionality
  - Slides over the bones and joints



- We have non-linear, non-constant relationships between muscles and joints

[Wilkinson et al, ICRA 03]

GP-Based Control



Overview

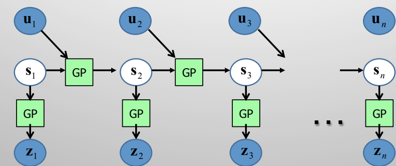
- Gaussian Processes and Bayes Filters
- GP-BayesFilters
- Filtering and Control
- System Identification with GP-BayesFilters
- Predictive State Representations
- Conclusions

GP Latent Variable Models

- Sometimes ground truth states are not or only partially available
- Instead of optimizing over GP hyperparameters only, optimize over latent states *S* as well

[Ko-F: RSS-09, ARJ-10]

GP Latent Variable Models

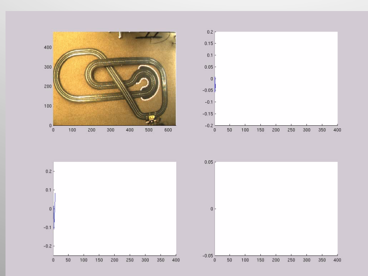


- Latent variable models [Lawrence: NIPS-03, Wang-etal: PAMI-08]
- Learn latent states and GPs in one optimization

$$\arg\max_{S,\Theta_Z,\Theta_S} \log p(S,\Theta_Z,\Theta_S \mid Z,U,\hat{S})$$
$$= \log p(Z \mid S,\Theta_Z) + \log p(S \mid U,\Theta_S) + \log p(S \mid \hat{S}) + \log p(\Theta_Z) + \log p(\Theta_S) + const$$

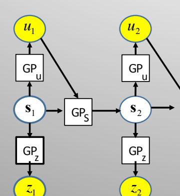
- Can take noisy labels into account

Slotcar Testbed



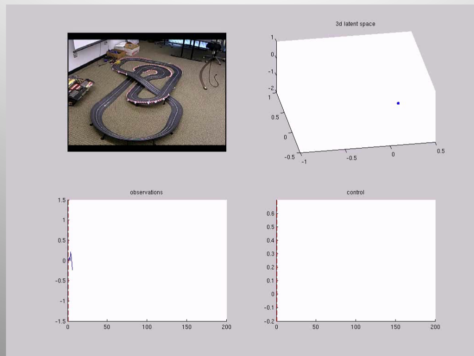
- Track contains banked curves, elevation changes
- Custom IMU with gyro and accelerometer built by Intel Research Seattle
- Observations very noisy, perceptual aliasing

Simple Trajectory Replay



- Learning
  - Human demonstrates control
  - Learn latent states using GPBF-Learn
  - Learn mapping from state to control
- Replay
  - Track state using GP-BayesFilter
  - Use control given by control GP

Trajectory Replay



Overview

- Gaussian Processes and Bayes Filters
- GP-BayesFilters
- Filtering and Control
- System Identification with GP-BayesFilters
- Predictive State Representations
- Conclusions

In Hand Manipulation

[Mordatch-Popovic-Todorov: SCA-12]

Contact-invariant optimization for hand manipulation

Mordatch, Popovic and Todorov
SCA 2012

Learning Models for Manipulation

- Soon manipulators / hands / robots will be equipped with a variety of complex sensors (e.g. touch sensitive skin)
- Are accurate physics-based models the most appropriate representation for controlling such complex systems?
- Rather than imposing a model on the dynamical system, learn a state space that's suitable for prediction and control
- Question: Can we learn expressive models from raw, high-dimensional sensor data?

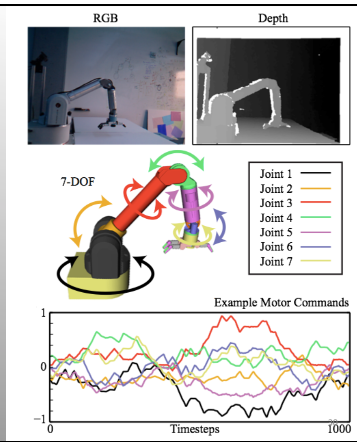
Predictive State Representations (PSRs)

$$\cdots \quad o_{t-1} \quad a_t \quad o_t \quad a_{t+1} \quad o_{t+1} \quad a_{t+2} \quad \cdots$$

- Expressive dynamical system model
- Test: ordered sequence of action observation pairs
  $$\tau = a_1 o_1 \ldots a_t o_t$$
- Prediction of a test: $\mathbb{P}\left[\tau^{\mathcal{O}} \mid \mathrm{do}\left(\tau^{\mathcal{A}}\right), h_t\right]$
- PSR state is a prediction over a set of core tests (future observable quantities)

Test Case

[Boots-Byravan-F: ICRA-14]



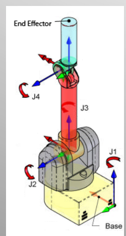
Learning Predictive Models of a Depth Camera & Manipulator from Raw Execution Traces

Byron Boots, Arunkumar Byravan, and Dieter Fox
Computer Science and Engineering
University of Washington

Summary

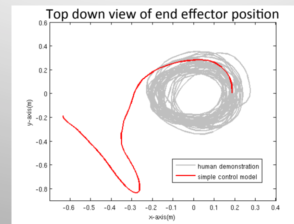
- GPs provide flexible modeling framework
- Take data noise and uncertainty due to data sparsity into account
- Seamless integration into Bayes filters
- Combination with parametric models increases accuracy and reduces amount of training data
- Subspace identification via latent variable models
- Computational complexity of GPs is a key problem
- Predictive state representations: scale to high-dimensional systems

WAM Trajectory Replay



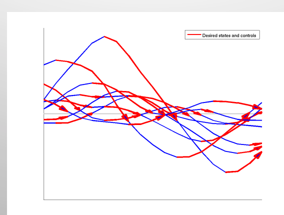
- System: Barrett Whole Arm Manipulator
  - Four joints/degrees of freedom
  - 4D control (change in joint angles)
  - Significant control noise
- Observations:
  - 3D position of end effector
- User demonstration:
  - Manipulate to trace out circular trajectory of end effector

Control Experiment



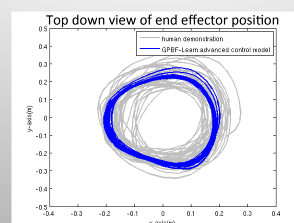
- Learn 3D latent states for system
- Replay assuming noisy encoders
- Both time-based and simple control model fail

Simple Fix



- Want controls which decrease prediction uncertainty
- Prediction uncertainty obtained from GP
- Learn control model using only desired state-control pairs

Advanced Control Experiment



- Learn 3D latent states for system
- Replay assuming noisy encoders
- Both time-based and simple control model fail
- Advanced control model achieves proper replay