Learning dynamics for robot control under varying contexts

Giorgos Petkos

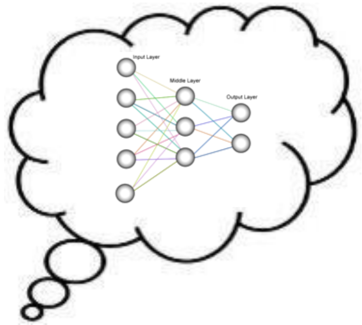
Informatics and telematics institute

Motor control

- Biological motor systems show a remarkable level of adaptability and robustness under different conditions

- We are interested to achieve similar levels of ability in robots as well

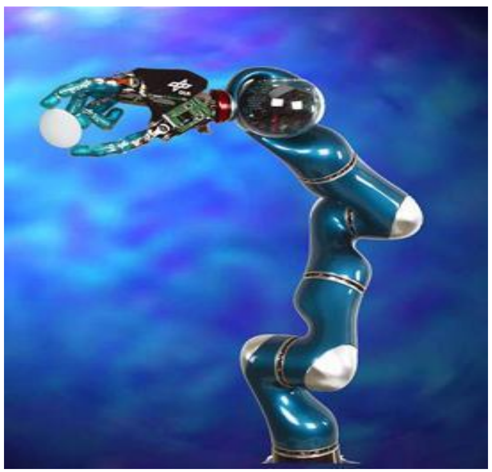
Motor control / dynamics

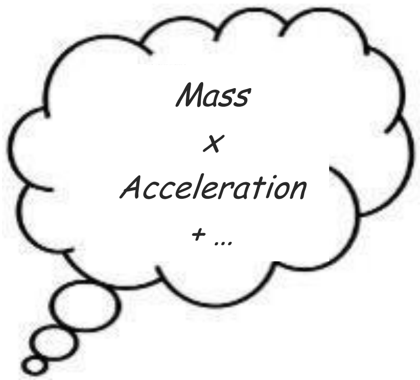


- High performance control requires an accurate dynamics model

- Usually model is derived using knowledge of the physical properties of the robot

Learning dynamics

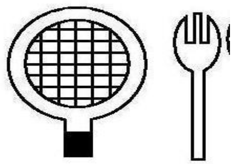


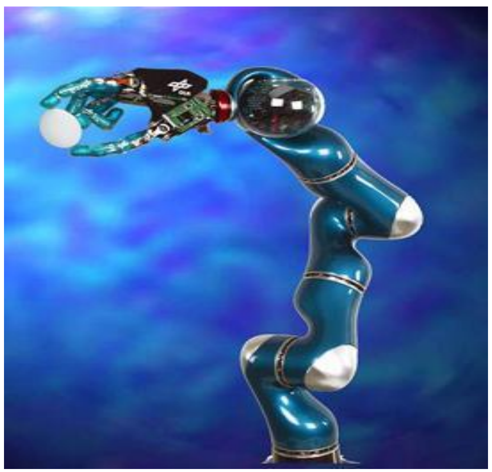


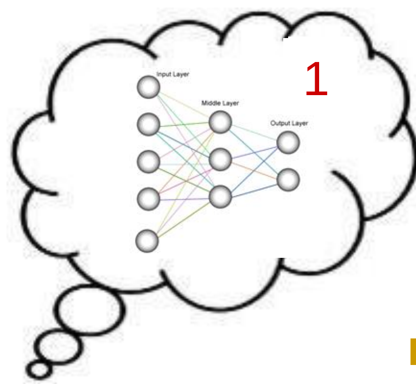
- Accurate derivation of dynamics may not be possible due to:
  - Unmodeled effects
  - Inaccurate knowledge or complete lack of knowledge of physical parameters

- Alternative is to learn the dynamics

Nonstationarity

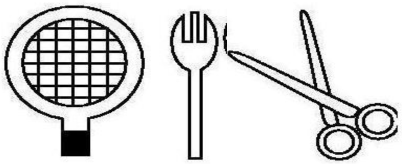


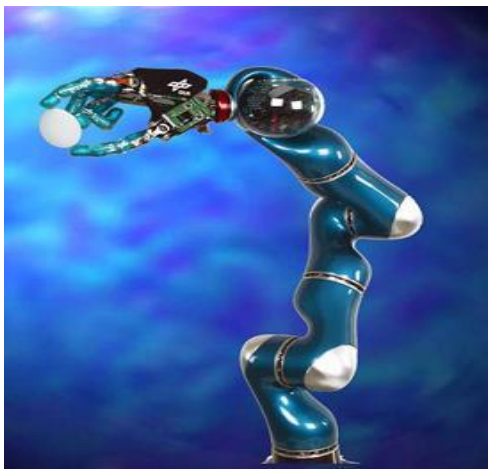




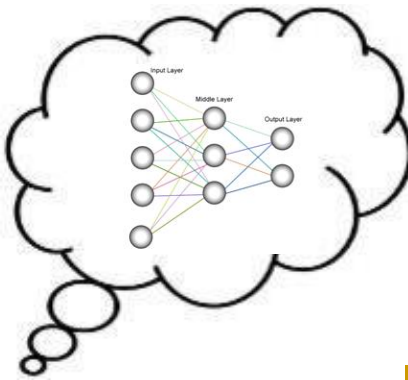
- Dynamics may change due to interaction with different environments and objects. Varying context of the dynamics.
- Model needs to be adapted every time the context changes
- The context is in most cases not directly observed

Goals





1 ? >1

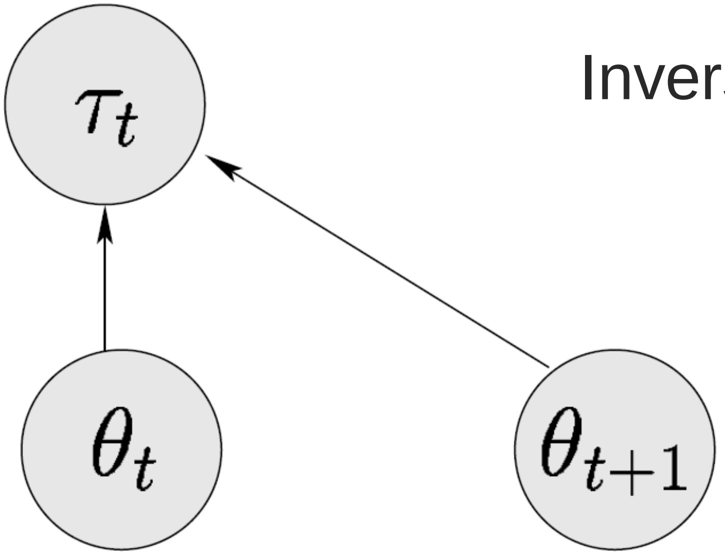


- However, if dynamics switch back and forth, readapting every time is suboptimal. Need to cope with recurring contexts.
- What happens with novel contexts? We would like to generalize from seen contexts to novel contexts.

Overview

1. Learning single context dynamics for control
2. Learning multiple models for a set of recurring contexts
3. Learning a single model with continuous latent contextual variables for a range of varying contexts
4. Learning a single model with observed continuous contextual variables

Learning single context dynamics

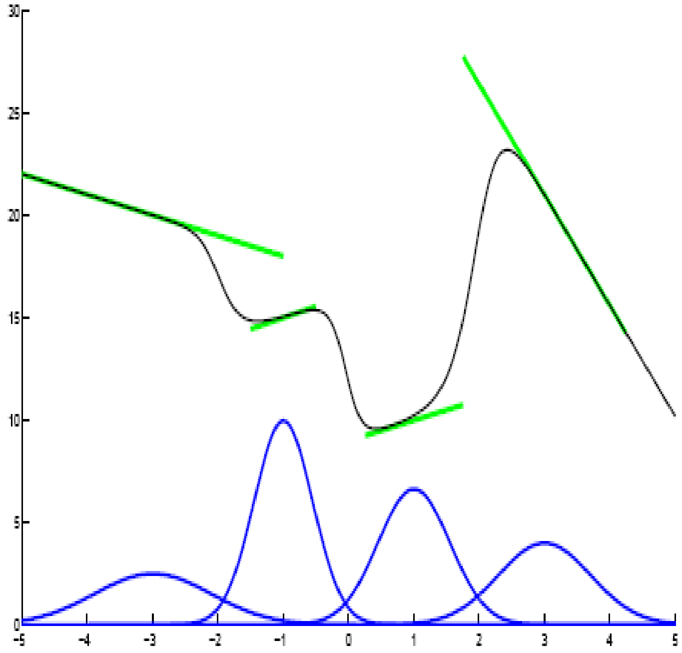


Inverse dynamics for a single context

$$\tau_t = g(\Theta_t, \Theta_{t+1})$$

Learning the inverse model for a single context is (relatively) straight forward since all data is observed, we just need a learning algorithm

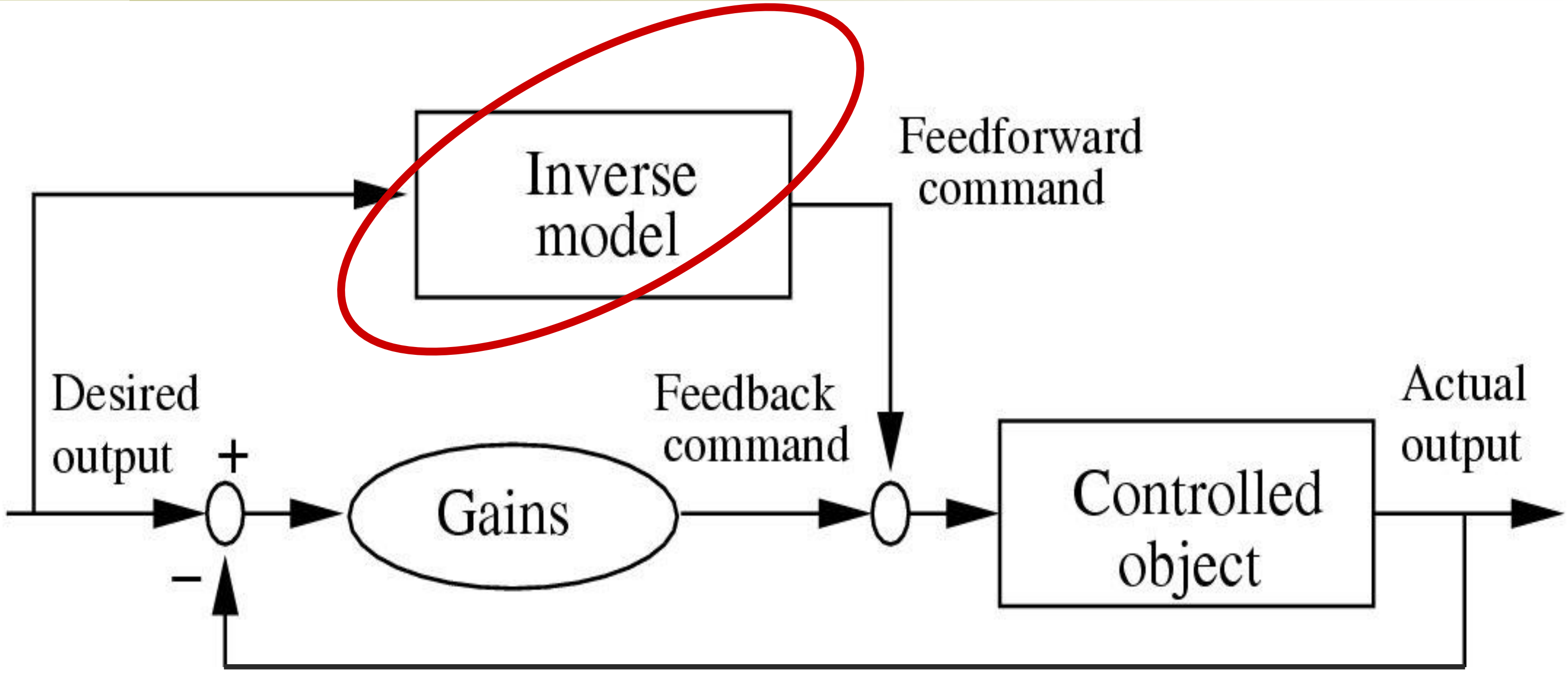
Locally Weighted Projection Regression



- Pairs of linear models and gaussian locality kernels
- Overall prediction weighted sum of individual models' predictions
- Local model PLS
- Online learning of local projection directions, regression coefficients and kernel shape
- Provides input dependent confidence bounds
- Appropriate for motor learning tasks in high dimensional spaces

Vijayakumar, D'Souza and Schaal. Incremental online learning in high dimensions. Neural Computation, 17:2602-2634

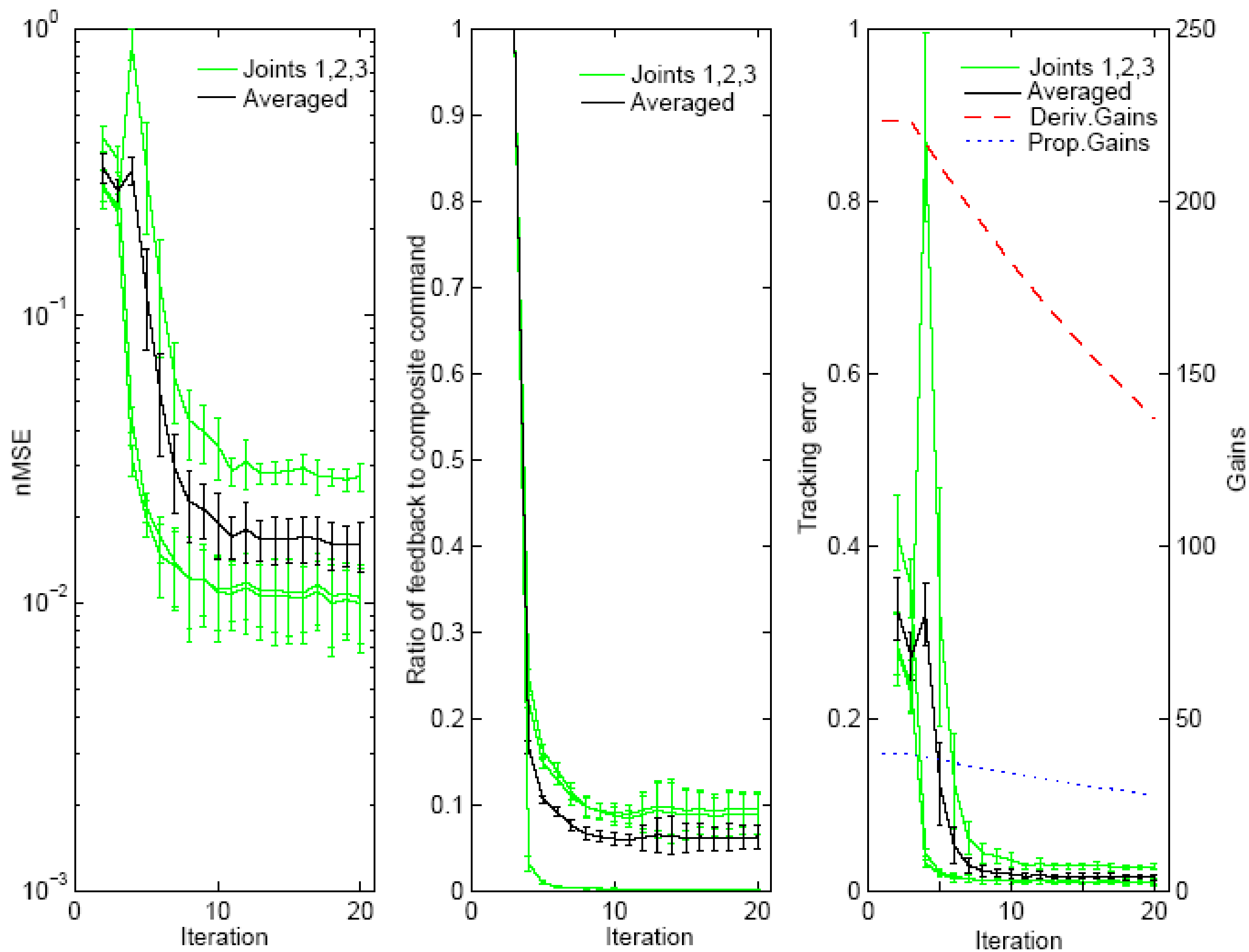
…and using it for control



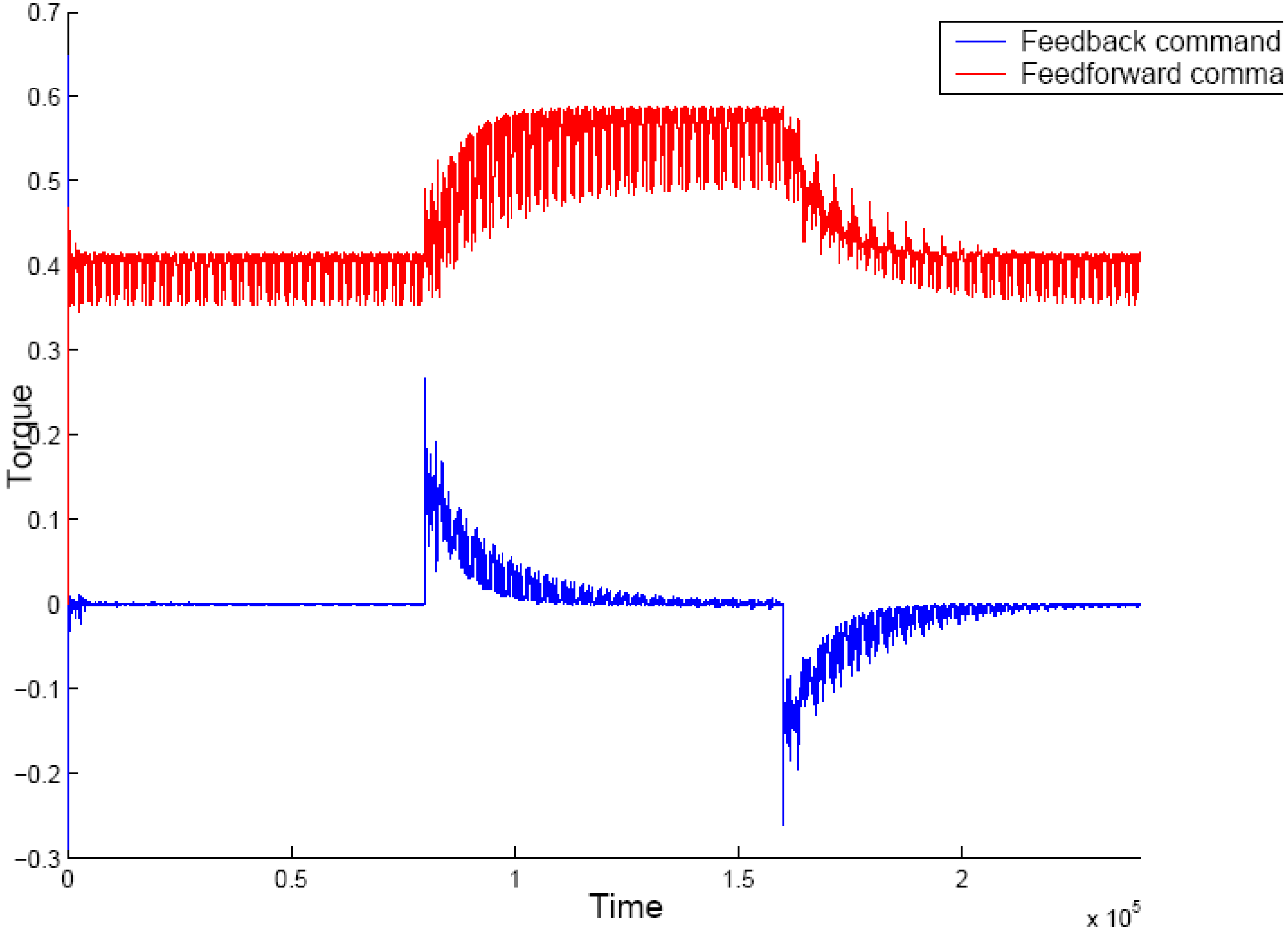
$$\tau_t = g(\theta_t^d, \theta_{t+1}^d) + K_p e_t + K_d \frac{de}{dt}$$

Example of learning dynamics for control

- Simple 3 Dof arm
- Sinusoidal movement task
- Switch from PD to composite control at iteration 4
- Results averaged over 10 trials

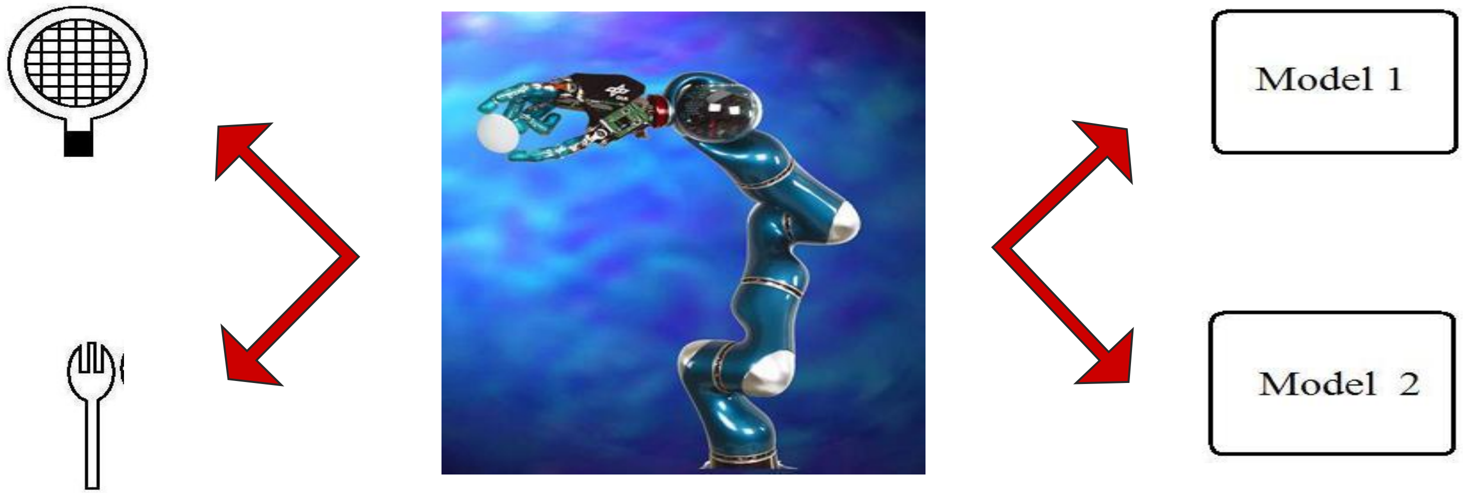


Recurring contexts?



- Tracking error behaves similarly

Multiple Models



Learn one model for each different context and switch between them appropriately

Studies claim that the CNS uses multiple models and switches between them.

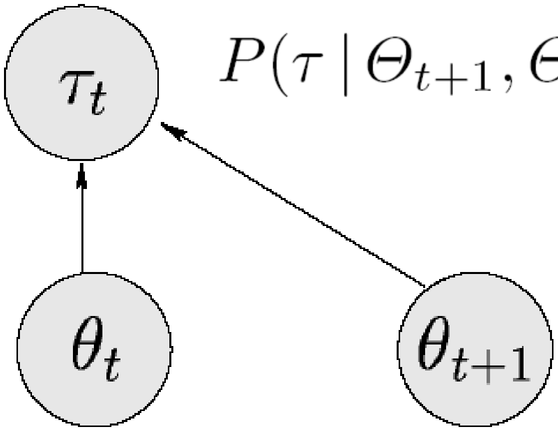
Existing multiple models approaches

- MMST (Narendra) : limited learning, prior knowledge on possible dynamics required
- MPFIM (Wolpert) : pairs of forward and inverse models, only shown to work with linear dynamics models, learning with online gradient descent
- MOSAIC (Haruno) : similar to MPFIM, HMM used to smooth context estimates, trained with EM algorithm, shown to work with linear dynamics

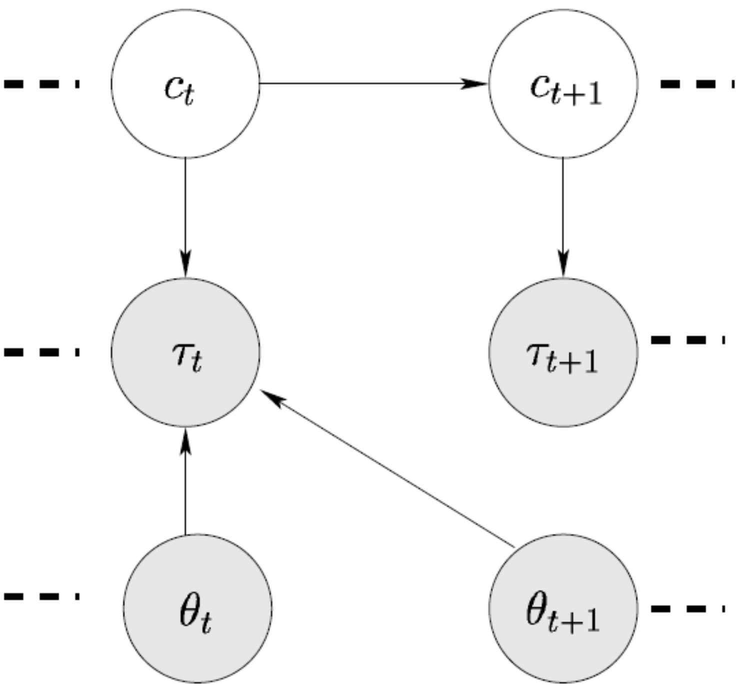
Multiple model formulation

$$P(\tau \mid \Theta_{t+1}, \Theta_t) = \mathcal{N}(\phi(\Theta_{t+1}, \Theta_t),\ \sigma(\Theta_{t+1}, \Theta_t))$$

$$P(\tau \mid \Theta_{t+1}, \Theta_t, c_t = i) = \mathcal{N}(\phi_i(\Theta_{t+1}, \Theta_t),\ \sigma_i(\Theta_{t+1}, \Theta_t))$$



Incorporating temporal context

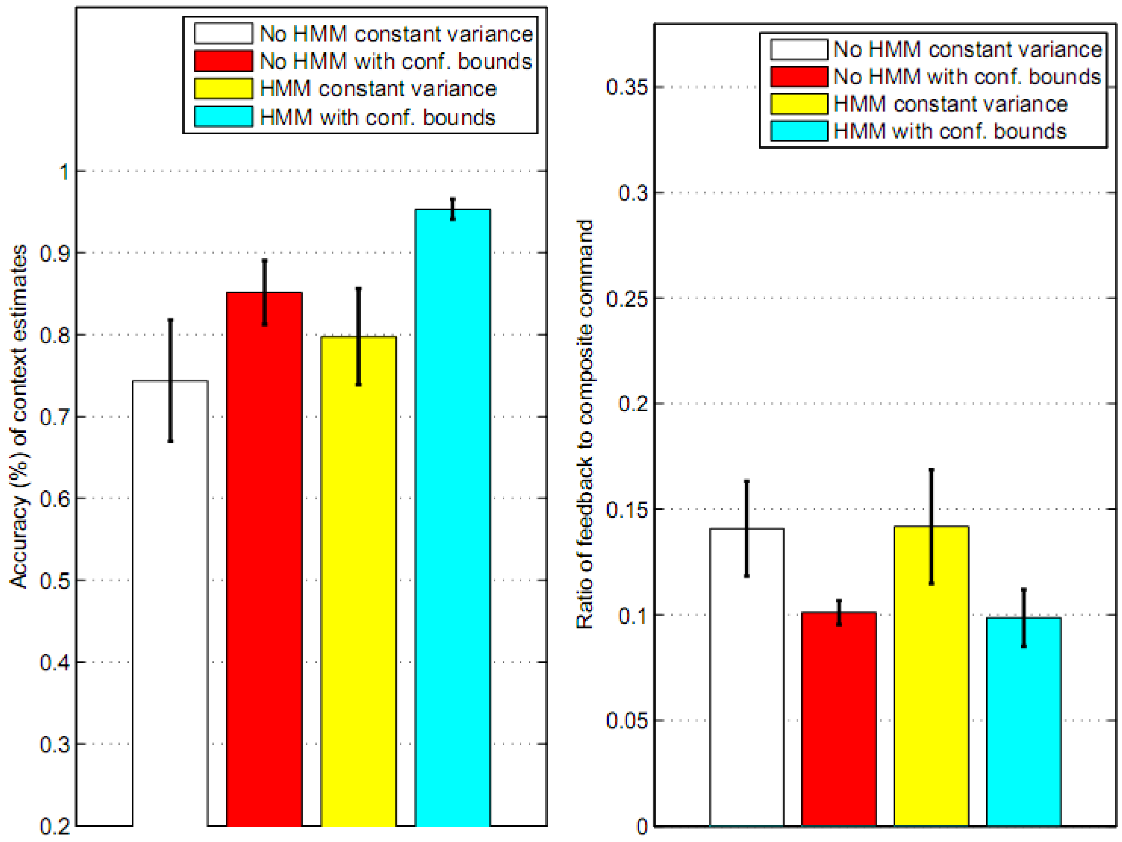


Inference and learning

- Inference simply involves using Bayes rule / standard graphical model mechanisms (Viterbi alignment / filtering / smoothing)

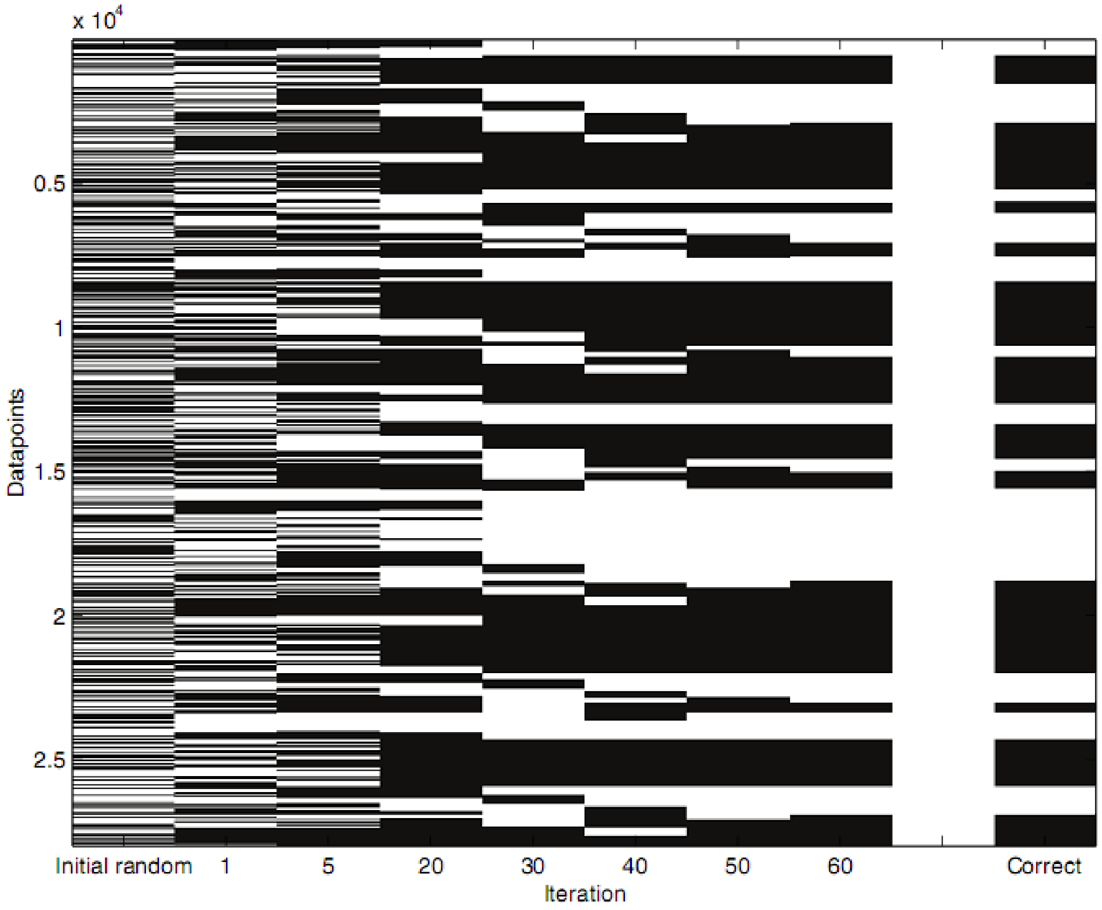
- Learning can be performed using EM (needs to be slightly modified when using LWPR)

Experiments: context estimation



- Random switching between 10 contexts.
- Results averaged over 5 trials

Experiments: learning



- Random switching between 2 contexts.
- Executed EM.
- Use of local models poses some difficulties.
- Need to switch from smoothed estimates to filtered estimates in the E-step.
- Confidence bounds used.

Other multiple model approaches

Previous methods

- Have limited or no learning ability (Multiple Model Switching and Tuning, Narendra)
- Have not been shown to work with non-linear dynamics (Modular Selection and Identification for Control, Wolpert and Kawato)

Our approach

- Is able to learn nonlinear dynamics and uses LWPR, a robust local based nonlinear regression algorithm
- Modified EM algorithm in order to deal with the problems induced by using a local regression algorithm

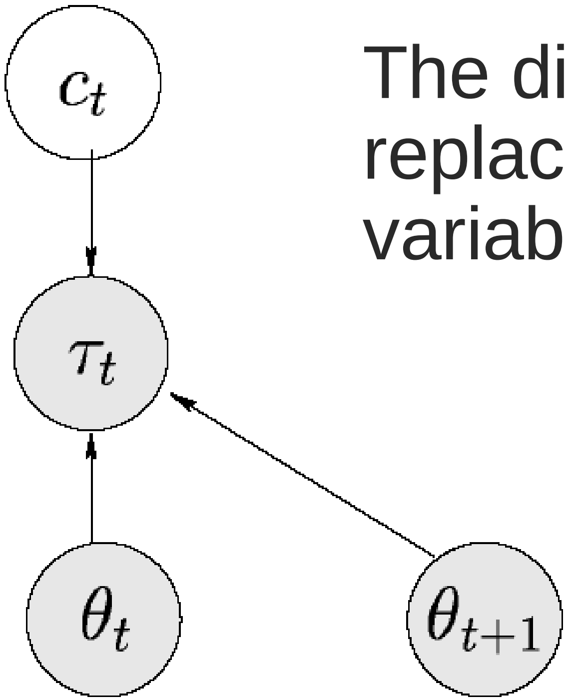
Issues with the multiple model scenario

- How can we generalize to new contexts?
- What is the appropriate number of contexts?
- What if there is an infinite number of possible contexts?

Instead of a set of models use a single one with additional contextual information as input

$$\tau_t = G(\theta_t, \theta_{t+1}, c_t)$$

Reformulating the model



The discrete contextual variable is replaced by a set of continuous variables

$$\tau_t = G(\theta_t, \theta_{t+1}, c_t)$$

$$P(\tau \mid \theta_{t+1}, \theta_t, c_t = k) = \mathcal{N}\left(g_k(\theta_{t+1}, \theta_t),\ \sigma_k(\theta_{t+1}, \theta_t)\right) c_t))$$

Learning the augmented model

- Use EM like before possible but too difficult
- How many variables?
- When possible it is better to use prior knowledge on the relationship of proper contextual variables to the dynamics

Special case: object manipulation

- This is possible in the case of manipulation of different objects (linear relationship of dynamics to a properly defined set of inertial parameters)

$$\sum_{j=1}^{J} y_{ij}(q,\dot{q},\ddot{q})^T \pi_j = \tau_i$$

- Y_{ij} are nonlinear functions of the joint angles, velocities and accelerations.
- The vector π_j holds the inertial parameters of the j_{th} link (mass, positions of center of mass and inertia tensor)

Special case: object manipulation

- Let us consider the dynamics of the context m.

$$\sum_{j=1}^{J} y_{ij}^{m}(q,\dot{q},\ddot{q})^{T}\pi_{j}^{m} = \tau_{i}^{m}$$

- The y_{ij} s do not depend on the context, only on kinematic properties that do not change:

$$y_{ij}(q,\dot{q},\ddot{q}) = y_{ij}^{1}(q,\dot{q},\ddot{q}) = \ldots = y_{ij}^{M}(q,\dot{q},\ddot{q}), \forall i,j$$

- Only the inertial parameters of the last link change between contexts:

$$\pi_{j} = \pi_{j}^{1} = \pi_{j}^{2} = \ldots = \pi_{j}^{M}, for\, j = 1 \ldots J-1$$

Obtaining the augmented model

- Thus, the model for the mth context can be written as:

$$\left(\sum_{j=1}^{J-1} y_{ij}(q,\dot{q},\ddot{q})^T \pi_j\right) + y_{iJ}(q,\dot{q},\ddot{q})^T \pi_J^m = \tau_i^m$$

$$h_i(q,\dot{q},\ddot{q}) + y_{iJ}(q,\dot{q},\ddot{q})^T \pi_J^m = \tau_i^m$$

Obtaining the augmented model

- Compiling the dynamics equations for all joints:

$$\begin{bmatrix} \tau_1^m \\ \tau_2^m \\ \vdots \\ \tau_J - 1^m \end{bmatrix} = \begin{bmatrix} h_1(q,\dot{q},\ddot{q}) \\ h_2(q,\dot{q},\ddot{q}) \\ \vdots \\ h_J(q,\dot{q},\ddot{q}) \end{bmatrix} + \begin{bmatrix} y_{1J}(q,\dot{q},\ddot{q})^T \\ y_{2J}(q,\dot{q},\ddot{q})^T \\ \vdots \\ y_{JJ}(q,\dot{q},\ddot{q})^T \end{bmatrix} \pi_J^m$$

$$\tau^m = A(q,\dot{q},\ddot{q}) + B(q,\dot{q},\ddot{q})\pi_J^m$$

Obtaining the augmented model

- Using π_j as the continuous context variables the augmented model can be:

$$\tau_t = G(\theta_t, \theta_{t+1}, c_t) = A(q, \dot{q}, \ddot{q}) + B(q, \dot{q}, \ddot{q})\pi_J^m$$

or

$$G(\theta_t, \theta_{t+1}, c_t) = \tilde{Y}(q, \dot{q}, \ddot{q})\tilde{\pi}_J^m = \tau$$

$$[A(q, \dot{q}, \ddot{q}) \; B(q, \dot{q}, \ddot{q})] \qquad [1 \; \pi_J^m]$$

Obtaining the augmented model means obtaining the matrix Y

Obtaining the augmented model

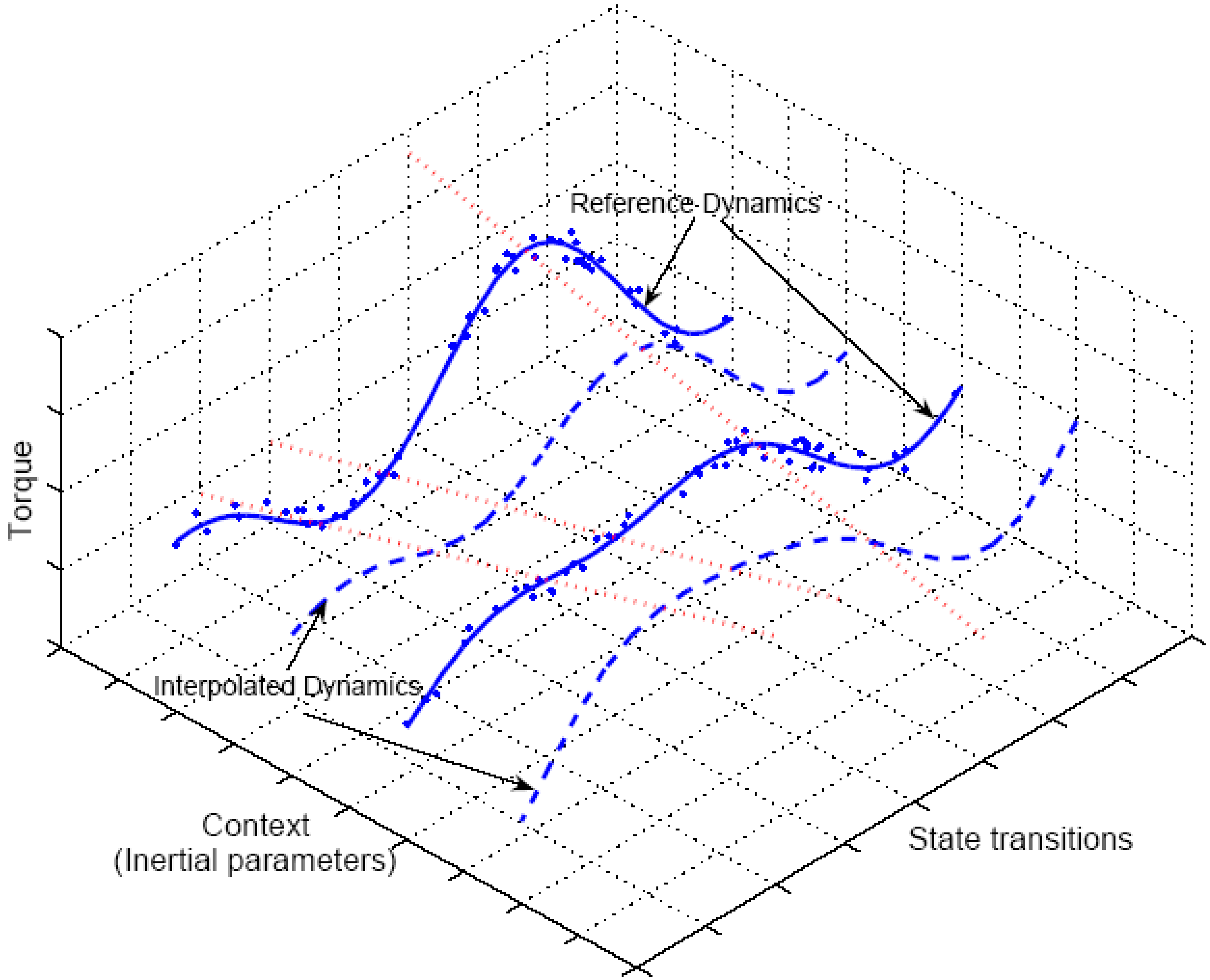
- If we have M learned reference models τ^1, τ^2 .. τ^M, together with their corresponding context variables π^1,π^2…π^M we can obtain Y as:

$$\left[\ \tau^1\tau^2\ldots\tau^M\ \right] = \tilde{Y}\left[\ \tilde{\pi}_J^1\tilde{\pi}_J^2\ldots\tilde{\pi}_J^M\ \right]$$

$$R = \tilde{Y}\tilde{\Pi}$$

$$\tilde{Y}(q,\dot{q},\ddot{q}) = R\tilde{\Pi}^T(\tilde{\Pi}\tilde{\Pi}^T)^{-1} = R\tilde{\Pi}^+$$

Obtaining the augmented model



Using the augmented model

- Infer the current inertial parameters in a probabilistic manner similar to the discrete case. Can use:

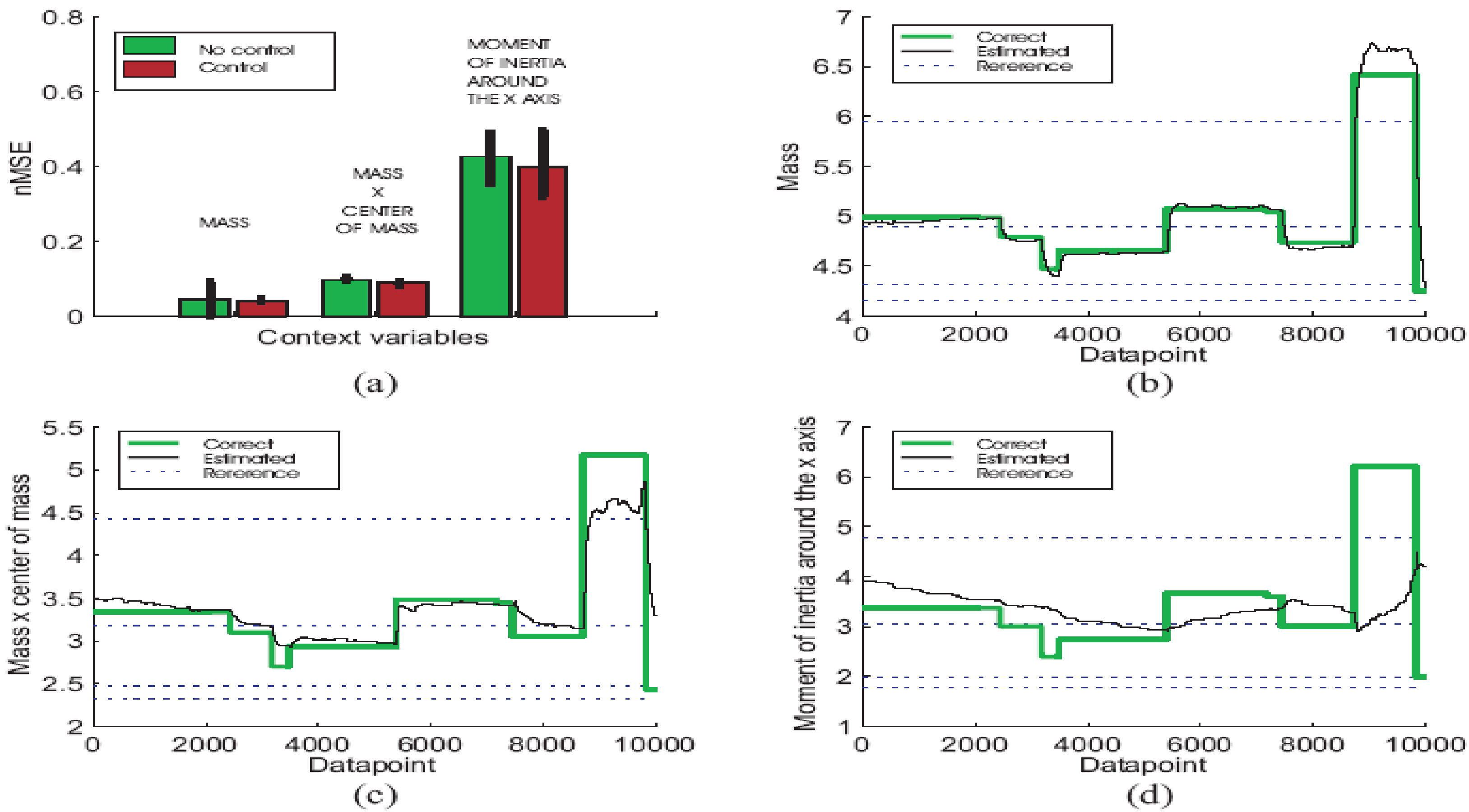
1. Non –temporal setting
2. Temporal setting (similar to kalman filtering)

- Use the estimated inertial parameters to get the feedforward command

Experiments: continuous context

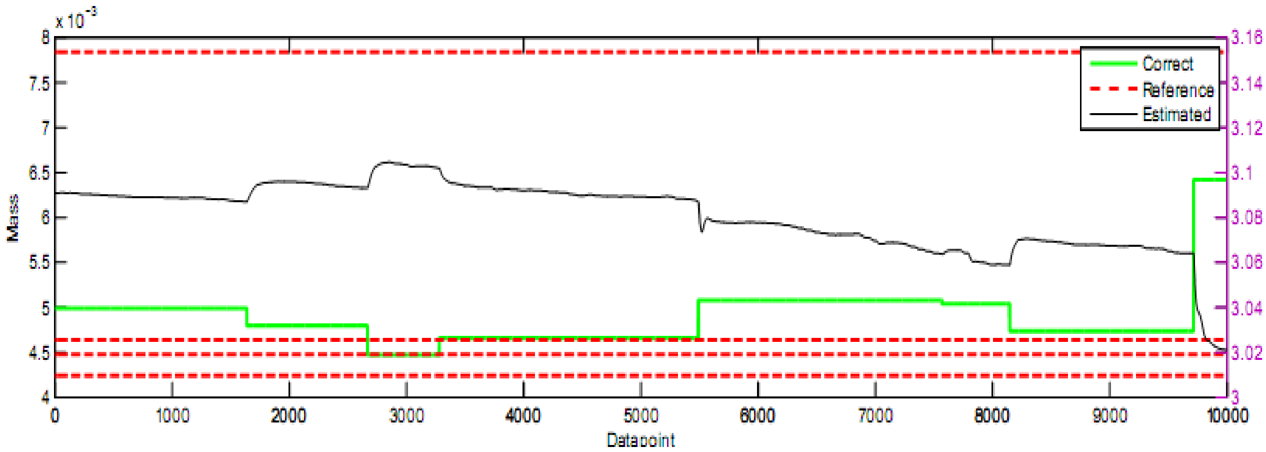
- Simulated 3 degree of freedom arm
- Randomly varying mass and shape.
- Inferred 3 inertial parameters
- Used the estimates for control
- 5 runs

Results



Ratio of feedback to composite command over the 3 joints was 0.1101 with std. 0.0176

Unavailability of reference models' intertial parameters

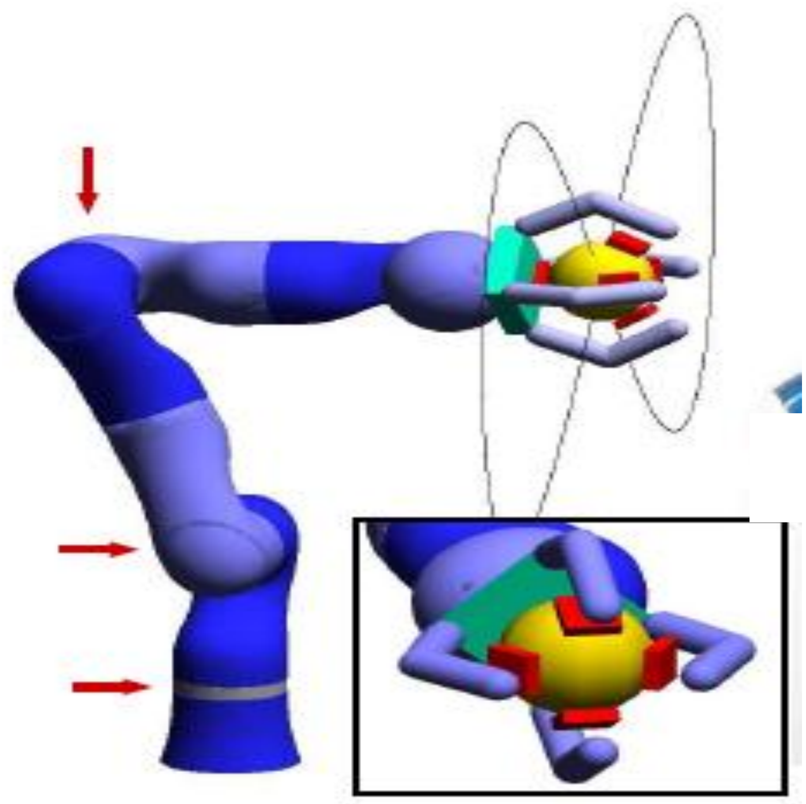


- Possible to reformulate so that reference models' inertial parameters are not required.
- Estimated context is a linear transformation of the actual context.
- Ratio of feedback to composite command 0.0893, std. 0.0239

Remarks

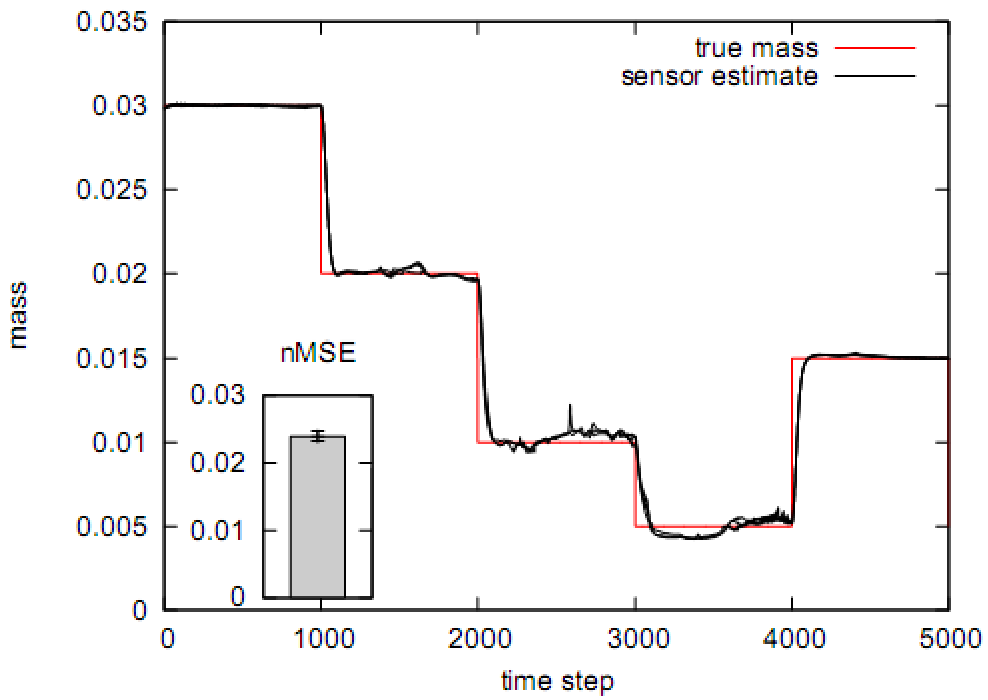
- This method is related to classical load estimation work on robotics
- Some inertial parameters cannot be estimated

Using tactile sensing to estimate the context



- Simplistic model of tactile sensing is linear in the mass of the manipulated object
  - Use this prior knowledge to generalize across context estimates
  - Use context estimates with an augmented dynamics model for control

Using tactile sensing to estimate the context

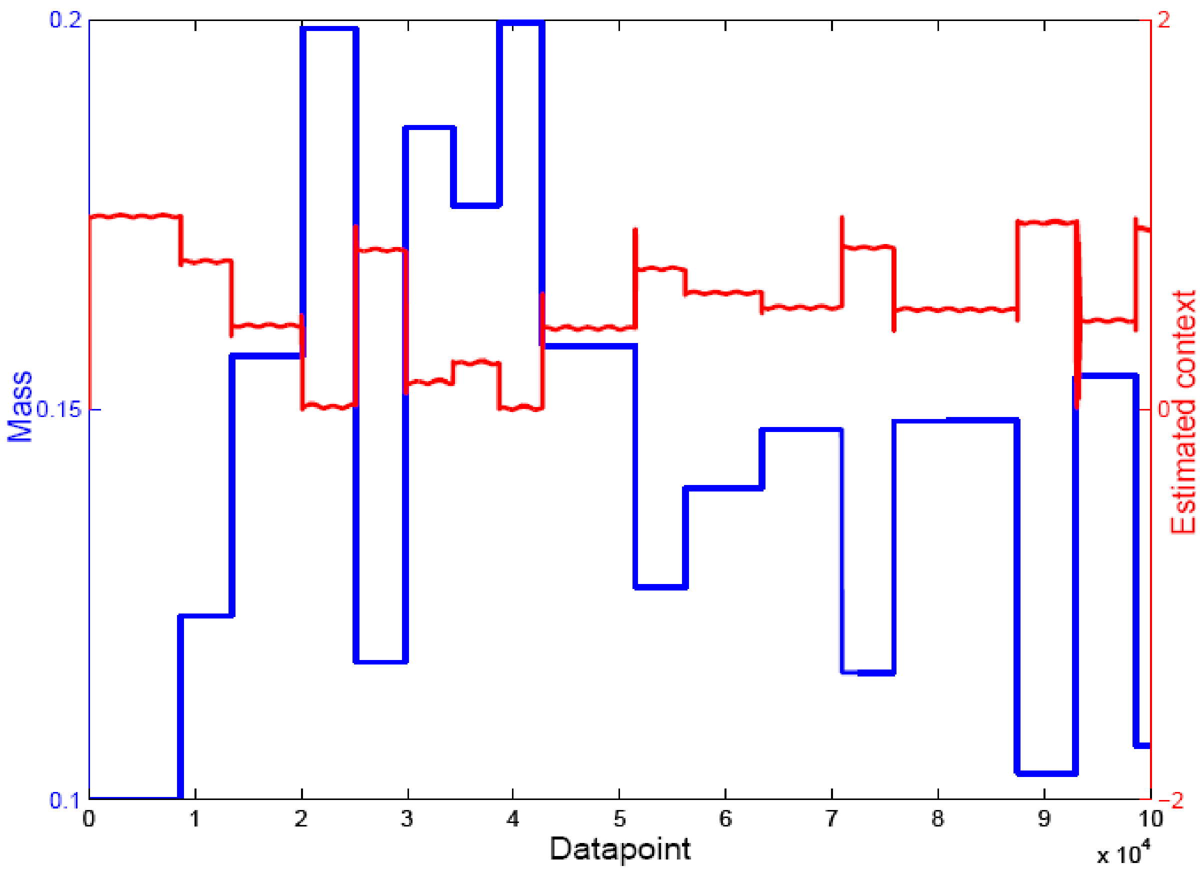


- Simulation of DLR arm
- Both sensor and augmented inverse models learned
- Three trials
- nMSE of estimates around 0.02

Unknown nature of context

- Relationship of context to dynamics may be unknown, possibly nonlinear
- Using the same temporal graphical model and just using a nonlinear augmented inverse model, we obtain a nonlinear state space model
- Exact inference and maximum likelihood learning not possible
- Possible choices for inference and learning include Extended Kalman Filtering, variational and Monte Carlo methods
- We use a variant of Monte Carlo EM (only best particle used for training).

Context estimation using a nonlinear state space model



- Simulation of simple 1DOF arm
- 100 particles
- 50 EM iterations
- Had to fix transition model in order to get a nice representation of the latent variable (mixture of uniform and gaussian with low noise variance).

An alternative

- Previous approaches require to perform context estimation before applying a control command
- Could use an augmented model with observed variables that convey information about the context indirectly
- Tactile for varying loads

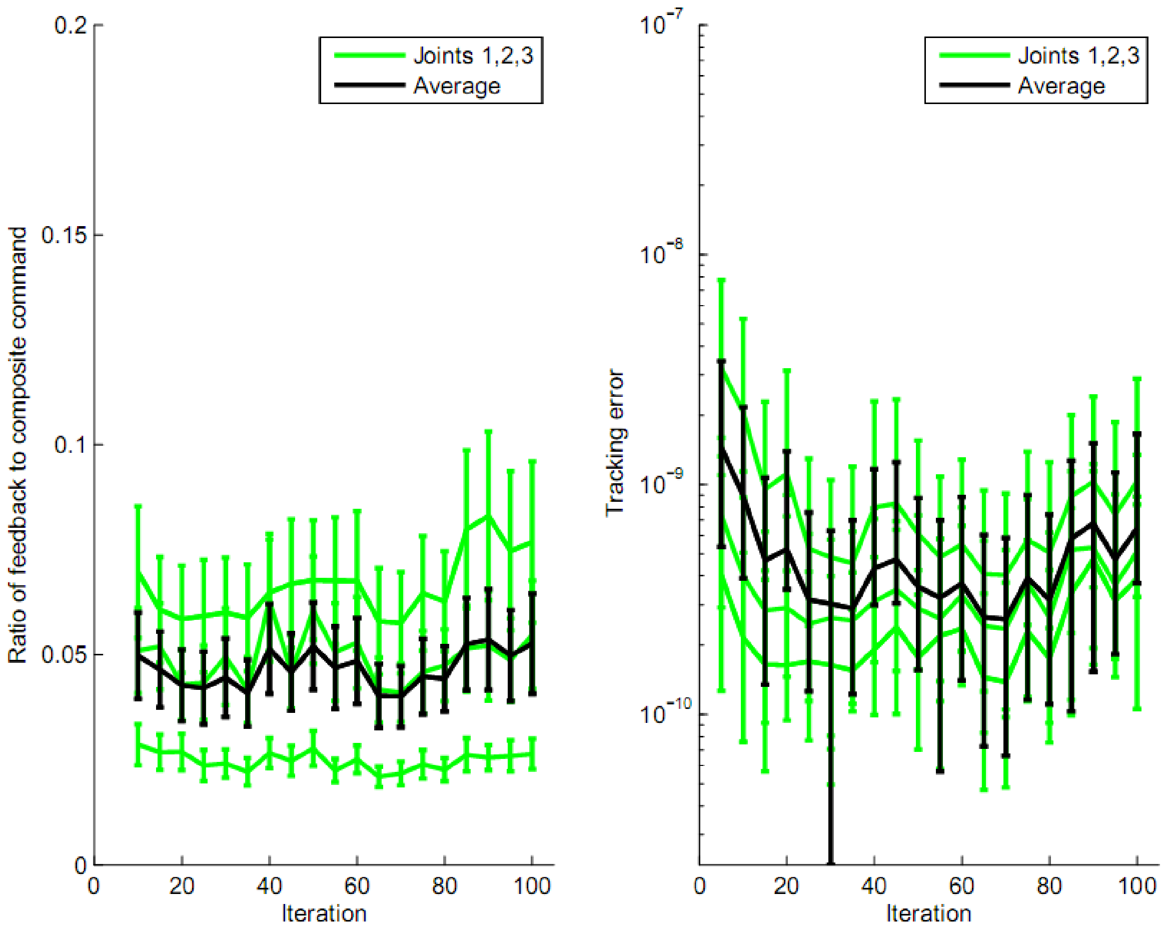
$$\tau = G(q, \dot{q}, \ddot{q}, T)$$

- Time delayed state transitions and commands. Essentially an autoregressive model

$$\tau_t = G(q_t, \dot{q}_t, \ddot{q}_t, q_{t-1}, \dot{q}_{t-1}, \ddot{q}_{t-1}, \tau_{t-1})$$

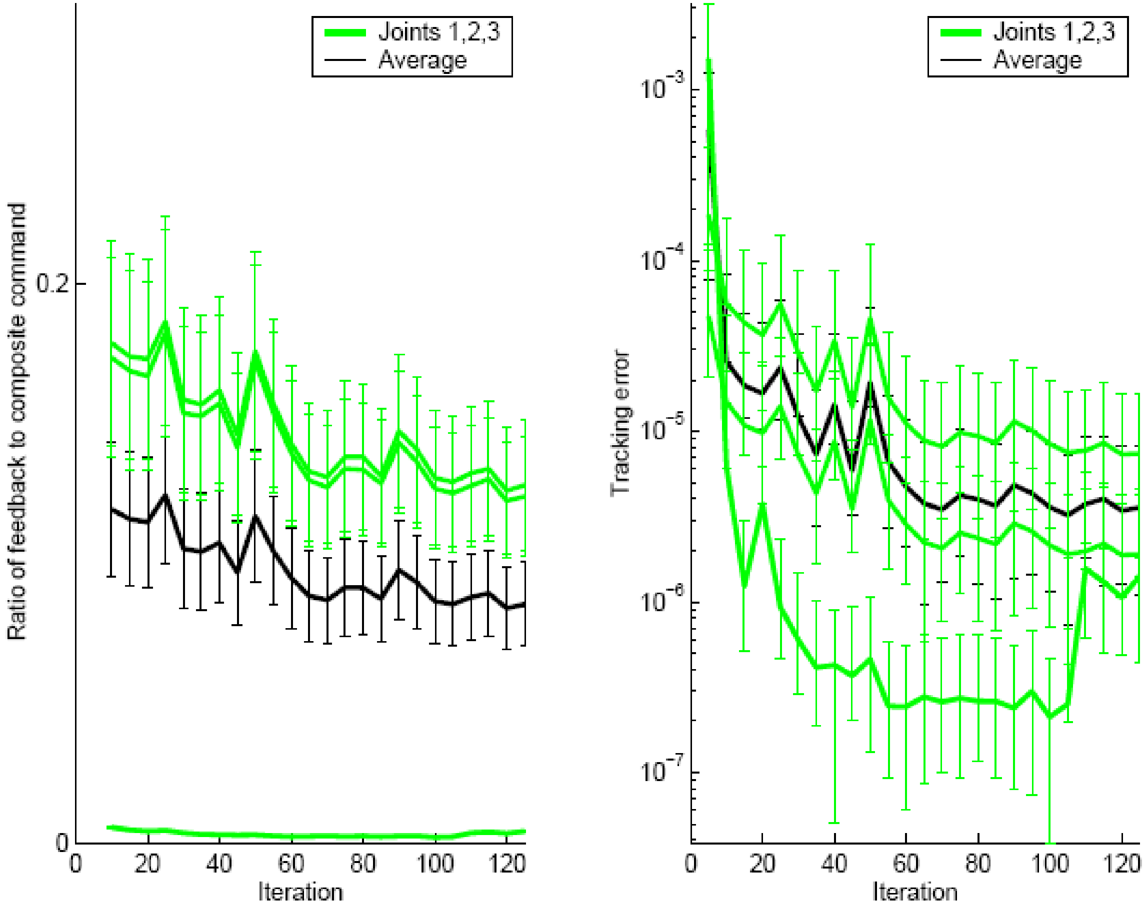
- Higher dimensional problem but context estimation is not required anymore

Results using the tactile augmented model



- DLR arm
- Training stops at iteration 80

Results using the autoregressive model

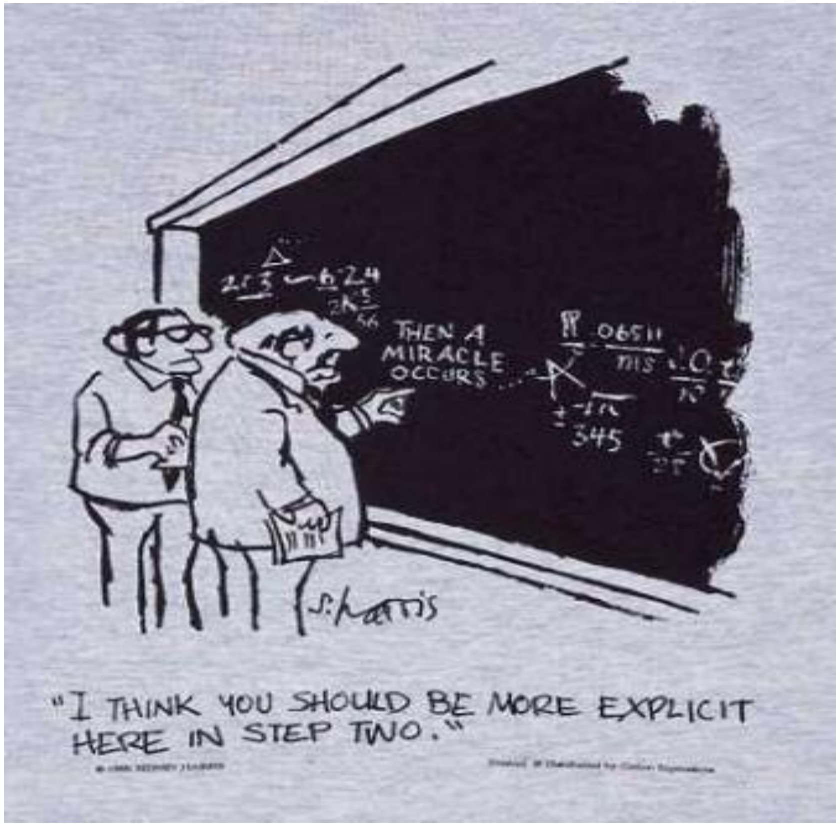


- Artificial arm 3DoF arm
- Training stops at iteration 100

Summary

- Learning of single model for stationary and non-stationary dynamics
- Multiple models for a set of recurring dynamics
- Continuous latent variable model for a range of contexts
  - Linear state space model for a scenario when the nature of the context is known
  - Nonlinear state space model when the nature of the context is not known
- Augmented model with variables that convey contextual information indirectly and are observed

Thank you!



- Questions?
- Comments?