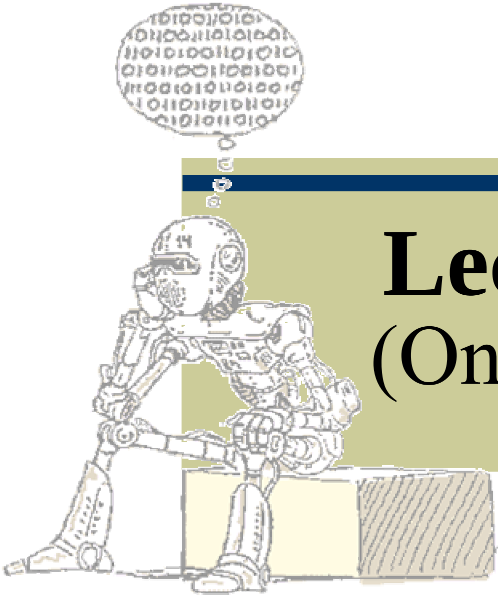


Lecture X– Putting it all together (Online Learning in High Dimensions)

Contents:

- LWPR and it’s application

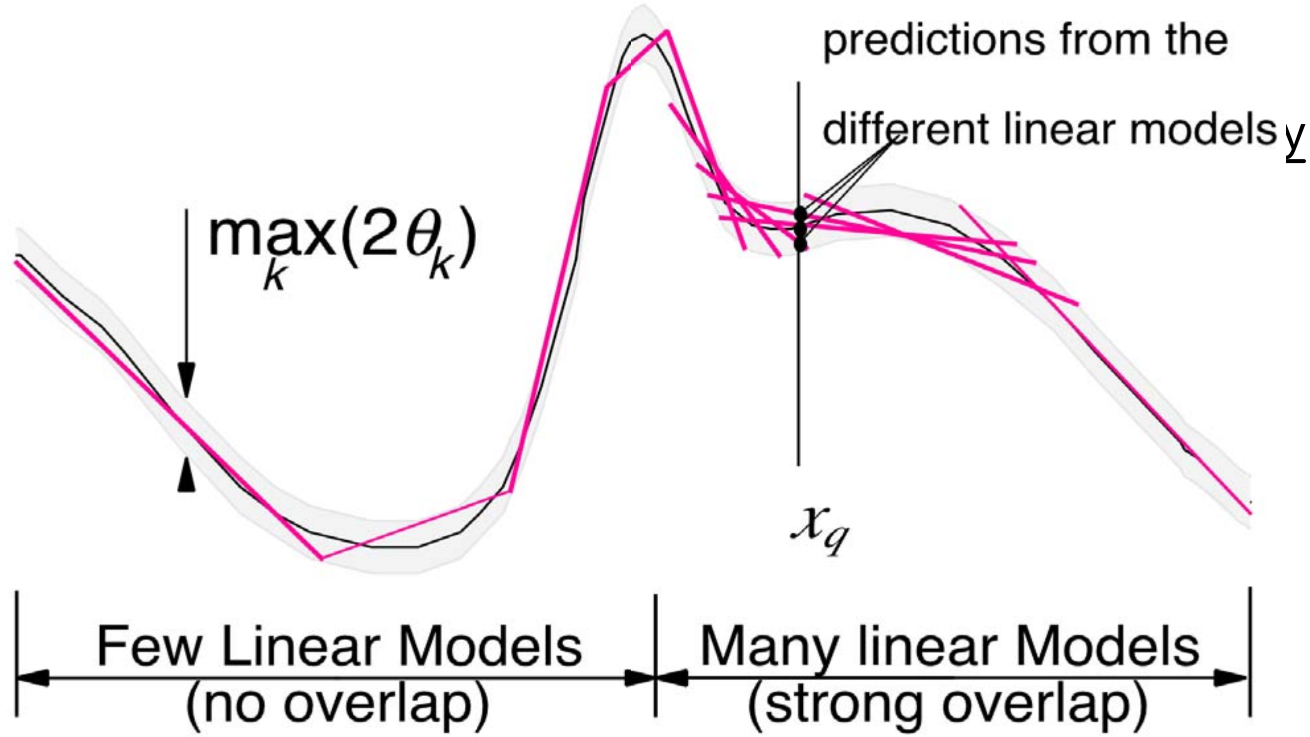
Locally Weighted Projection Regression (LWPR)

- **Projection Regression is 'cool'**
  - Computationally cheap
  - Numerically robust
  - Ideally suited for incremental addition of projections
  - Typically, fewer projections required than intrinsic dimensionality

Based on our analysis of Dimensionality reduction techniques, we will use PLS as our preprocessing step which also gives as the slope of the local linear model in each receptive field.

We then plug this into our LWR learning framework to get a very efficient incremental learning system, i.e. LWPR

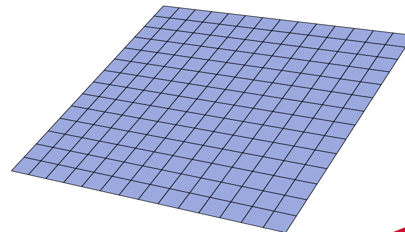
Locally Linear Models



If we can find the tangent (plane) and the region of validity from only local data, the function approximation problem can be solved efficiently

Formalization of Local Models

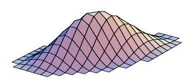
- **The Linear Model**



Open Parameters

$$\mathbf{y} = \beta_x^T \mathbf{x} + \beta_0 = \beta^T \tilde{\mathbf{x}} \quad \text{where} \quad \tilde{\mathbf{x}} = \left[\mathbf{x}^T \ 1\right]^T$$

- **The Kernel Function**



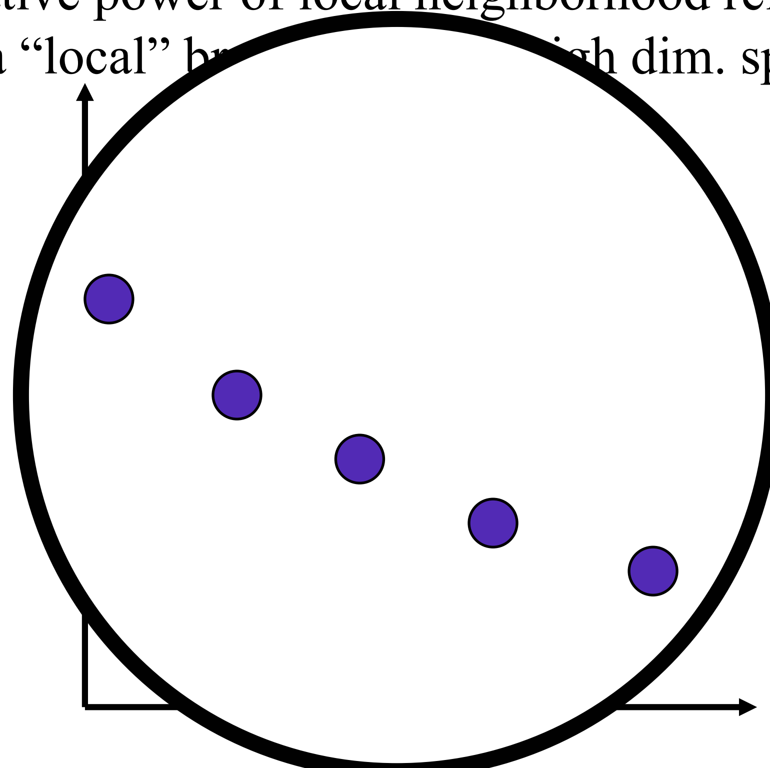
$$w = \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{c})^T \mathbf{D}(\mathbf{x}-\mathbf{c})\right) \quad \text{where} \quad \mathbf{D} = \mathbf{M}^T \mathbf{M}$$

- **The Prediction**

$$\mathbf{y} = \frac{\sum_{i=1}^{K} w_i \mathbf{y}_k}{\sum_{i=1}^{K} w_i}$$

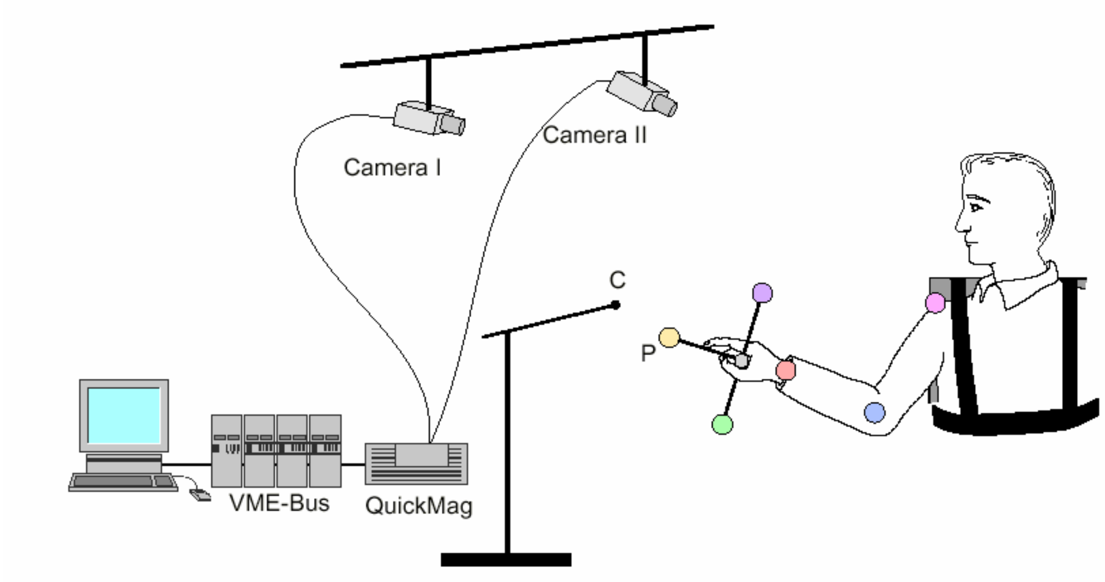
LWL in High Dimensional Spaces

- The Curse of Dimensionality
  - The power of local learning comes from exploiting the discriminative power of local neighborhood relations, but the notion of a “local” b … igh dim. spaces

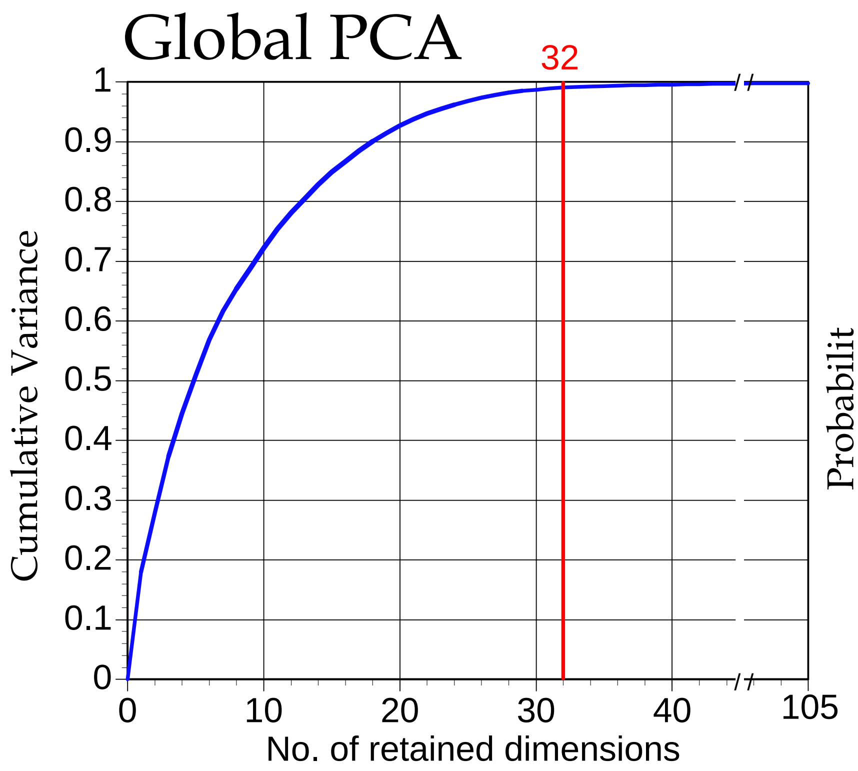


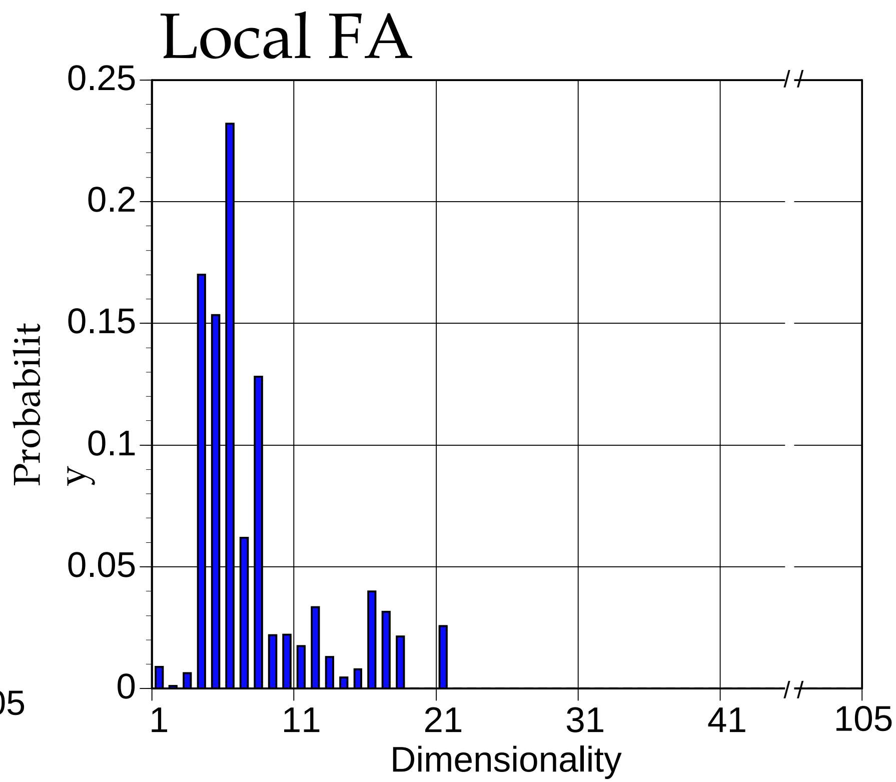
Human Movement

- Measure arm movement and full-body movement of humans and anthropomorphic robots
- Perform local dimensionality analysis with a growing variational mixture of factor analyzers



Dimensionality of Full Body Motion

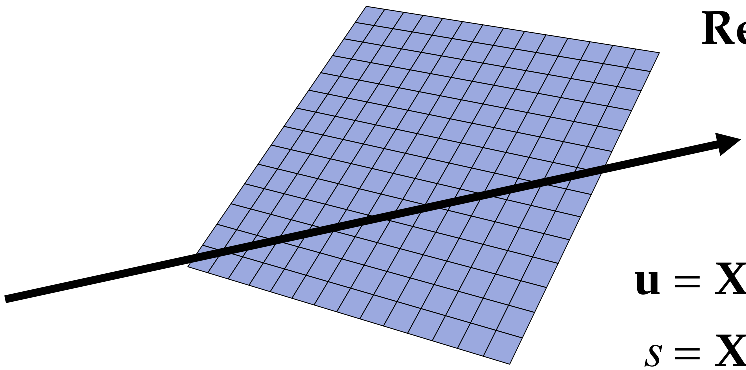




About 8 dimensions in the space formed by joint positions, velocities, and accelerations are needed to model an inverse dynamics model

Exploiting Locally Low Dim. Data

- **Use local dimensionality reduction techniques**
  - PCA
  - Factor Analysis
  - Partial Least Squares Regression



Regression Along One Dimensional Projections

$$\mathbf{u} = \mathbf{X}^T y$$

$$s = \mathbf{X}\mathbf{u}$$

$$\beta = \frac{s^T y}{s^T s}$$

Note: for spherical input distributions, PLS performs optimally with just one projection

Iterate multiple times on residual error

Learning the distance metric

$$\mathbf{D}^{n+1} = \mathbf{D}^{n} - \alpha \frac{\delta J}{\delta \mathbf{D}}$$

$$J = \frac{1}{W} \sum_{i=1}^{M} \sum_{k=1}^{r} \frac{w_i y_{\mathbf{res},k+1,i}^2}{\left(1 - w_i \frac{s_{k,i}^2}{s_k^T W s_k}\right)^2} + \gamma \sum_{i,j=1}^{N} D_{ij}^2$$

Regularization on the distance metric

Incremental Dimensionality Reduction

Incremental PLS

Initialize: $\mathbf{z}_0 = \mathbf{x} - \mathbf{x}_0^{n+1}$, $res_0 = y - \beta_0^{n+1}$

For $i = 1{:}r$,

a) $\mathbf{u}_i^{n+1} = \lambda\, \mathbf{u}_i^n + w\, \mathbf{z}_{i-1}\, res_{i-1}$

b) $s_i = \mathbf{z}_{i-1}^T \mathbf{u}_i^{n+1}$

c) $SS_i^{n+1} = \lambda\, SS_i^n + w\, s_i^2$

d) $SR_i^{n+1} = \lambda\, SR_i^n + w\, s_i^2\, res_{i-1}$

e) $SZ_i^{n+1} = \lambda\, SZ_i^n + w\, \mathbf{z}_{i-1}\, s_i$

f) $\beta_i^{n+1} = SR_i^{n+1} / SS_i^{n+1}$

g) $\mathbf{p}_i^{n+1} = SZ_i^{n+1} / SS_i^{n+1}$

h) $\mathbf{z}_i = \mathbf{z}_{i-1} - s_i \mathbf{p}_i^{n+1}$

i) $res_i = res_{i-1} - s_i \beta_i^{n+1}$

j) $SSE_i^{n+1} = \lambda\, SSE_i^n + w\, res_i^2$

Sufficient statistics

Lambda = Forgetting factor

Incremental Metric Adjustment

Incremental Distance Metric Update

$$\mathbf{M}^{n+1} = \mathbf{M}^{n} - \alpha \frac{\partial J}{\partial \mathbf{M}} \quad \text{with } \mathbf{D} = \mathbf{M}^{T}\mathbf{M}\text{, where } \mathbf{M} \text{ is upper triangular}$$

$$W^{n+1} = \lambda W^{n} + w$$

$$E_k^{n+1} = \lambda E_k^{n} + w\, res_{k-1}^{2}$$

$$\mathbf{H}_k^{n+1} = \lambda \mathbf{H}_k^{n} + \frac{w\, s_k\, res_{k-1}}{1-h_k}, \quad \text{where} \quad h_k = w \frac{s_k^2}{SS_k^{n+1}}$$

$$\mathbf{R}_k^{n+1} = \lambda \mathbf{R}_k^{n} + \frac{w^2\, res_{k-1}^{2} s_k^2}{1-h_k}$$

$$\frac{\partial J}{\partial \mathbf{M}} \approx \frac{\partial w}{\partial \mathbf{M}} \sum_{i=1}^{p} \sum_{k=1}^{r} \frac{\partial J_{1,r,i}}{\partial w} + \frac{w}{n\, W^{n+1}} \frac{\partial J_2}{\partial \mathbf{M}}$$

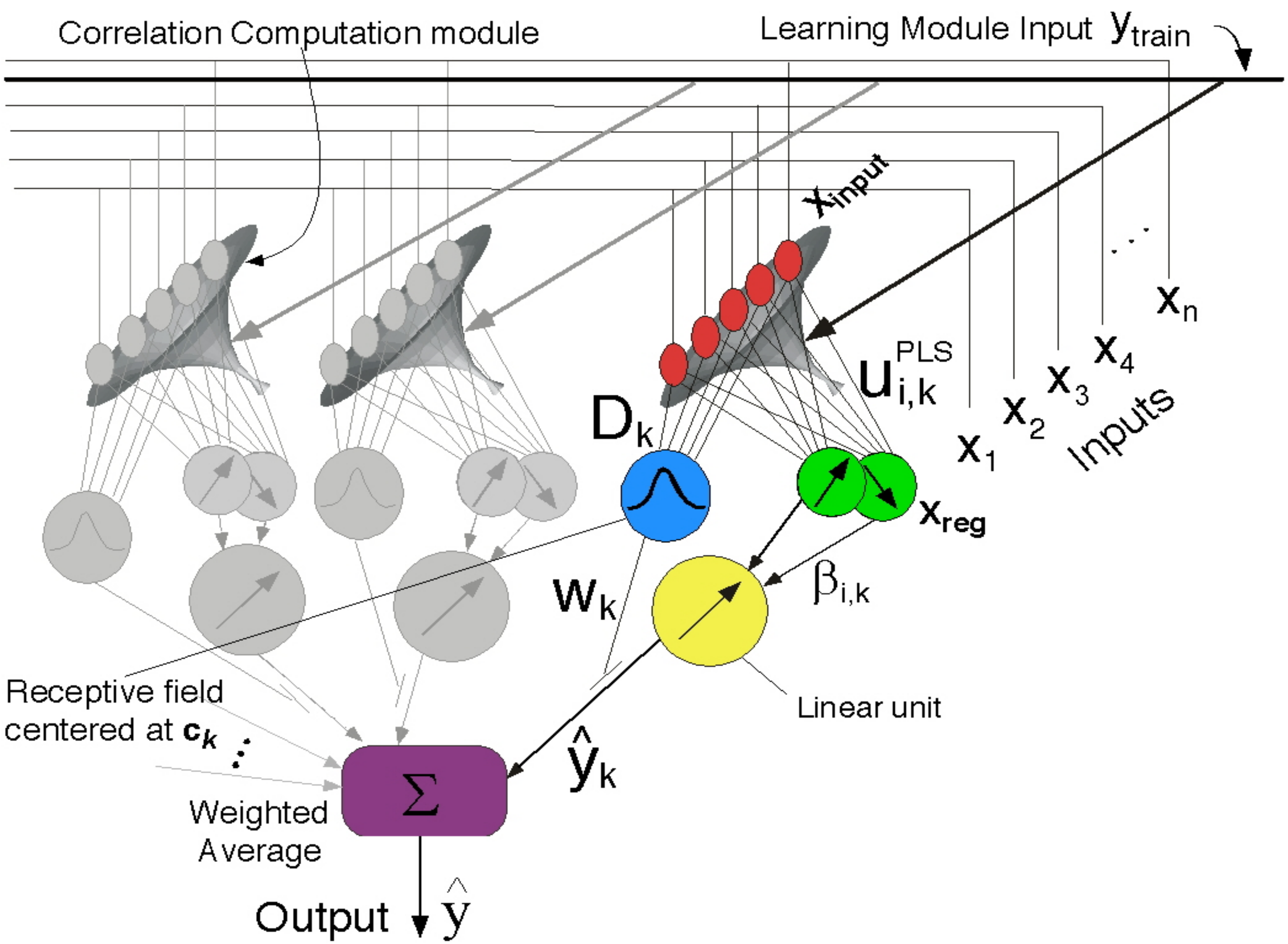
where:

$$\frac{\partial w}{\partial M_{rl}} = -\frac{1}{2} w (\mathbf{x}-\mathbf{c})^{T} \frac{\partial \mathbf{D}}{\partial M_{rl}} (\mathbf{x}-\mathbf{c}), \quad \frac{\partial J_2}{\partial M_{rl}} = 2\gamma \sum_{i,j=1}^{n} D_{ij} \frac{\partial D_{ij}}{\partial M_{rl}}$$

$$\frac{\partial D_{ij}}{\partial M_{rl}} = \delta_{lj} M_{ri} + \delta_{il} M_{rj} \quad (\delta \text{ is the Kronecker operator})$$

$$\sum_{i=1}^{p} \sum_{k=1}^{r} \frac{\partial J_{1,r,i}}{\partial w} \approx \sum_{k=1}^{r} \left[-\frac{E_k^{n+1}}{\left(W^{n+1}\right)^2} + \frac{1}{W^{n+1}} \left(res_{k-1}^{2} - 2\frac{res_k s_k}{SS_k^{n+1}} \mathbf{H}_k^{n} - 2\left(\frac{s_k}{SS_k^{n+1}}\right)^2 \mathbf{R}_k^{n} \right) \right]$$

Processing Units of LWPR

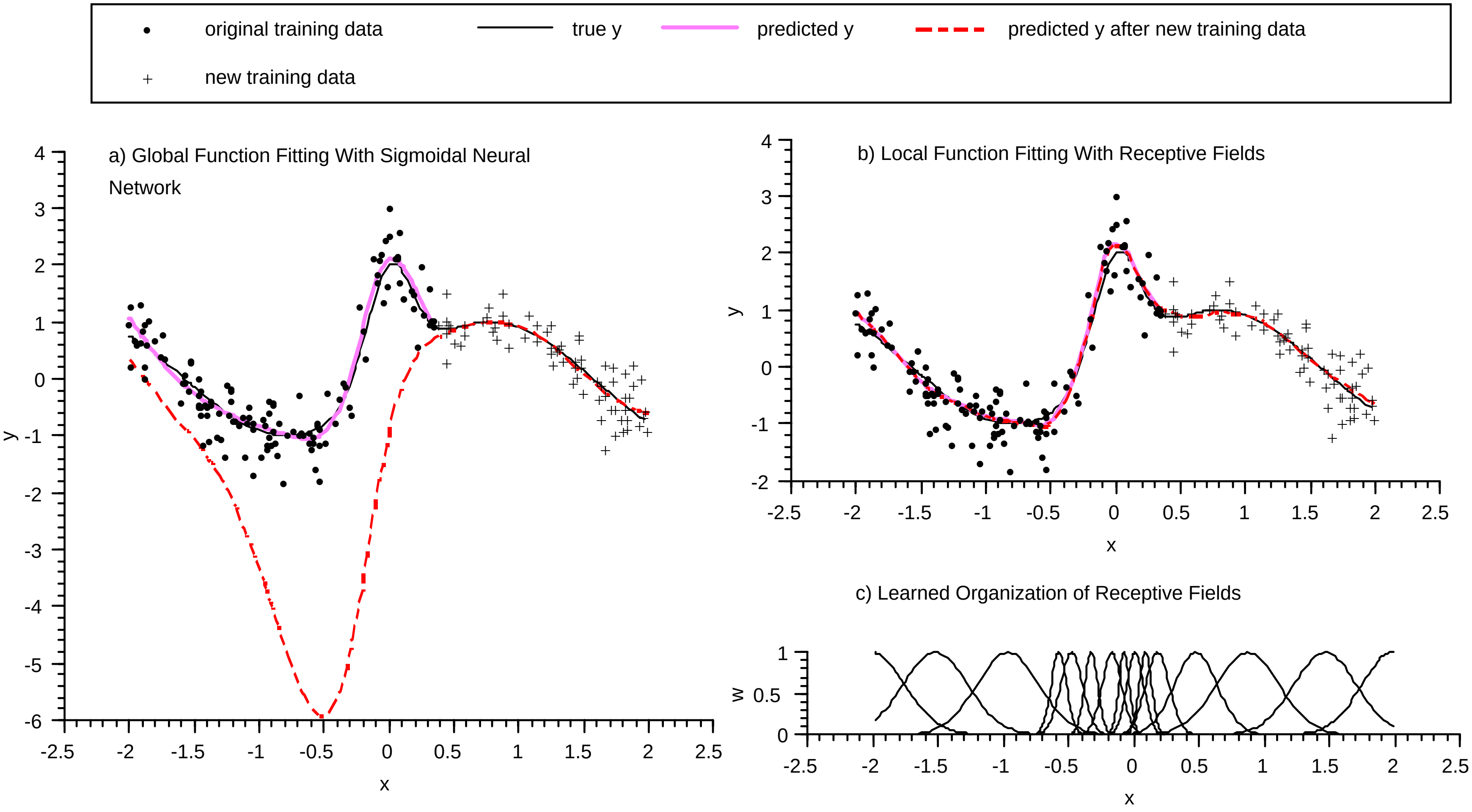


Summary – LWPR algorithm

- Fit the non-linear function with sum of ***weighted local linear models***
- In each of these local models, do a ***local dimensionality reduction*** for regression (orthogonal projections)
- Learn the ***locality of the models*** incrementally

Key point: *No competition while learning individual local models (unlike mixture models)*

Example: 1D Fitting

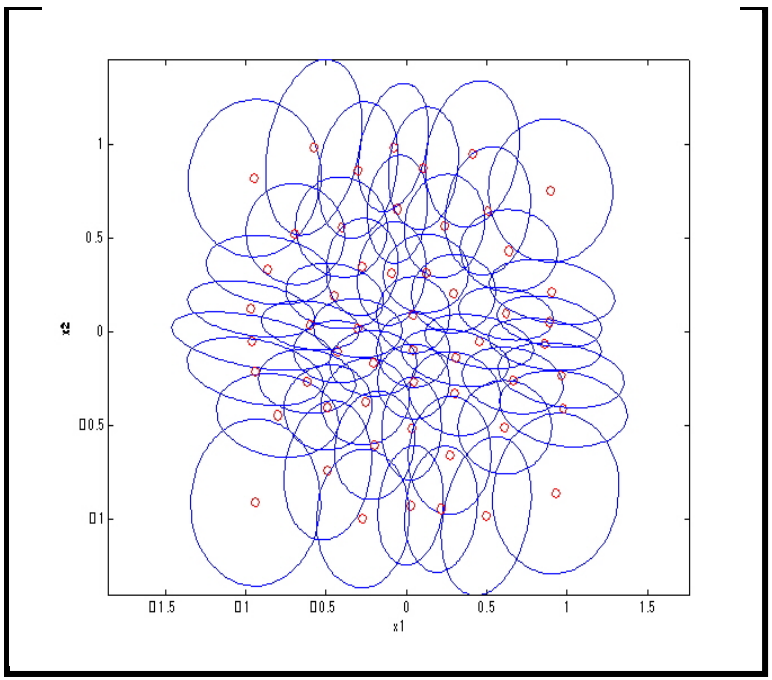


Empirical Evaluations (Cross Data)

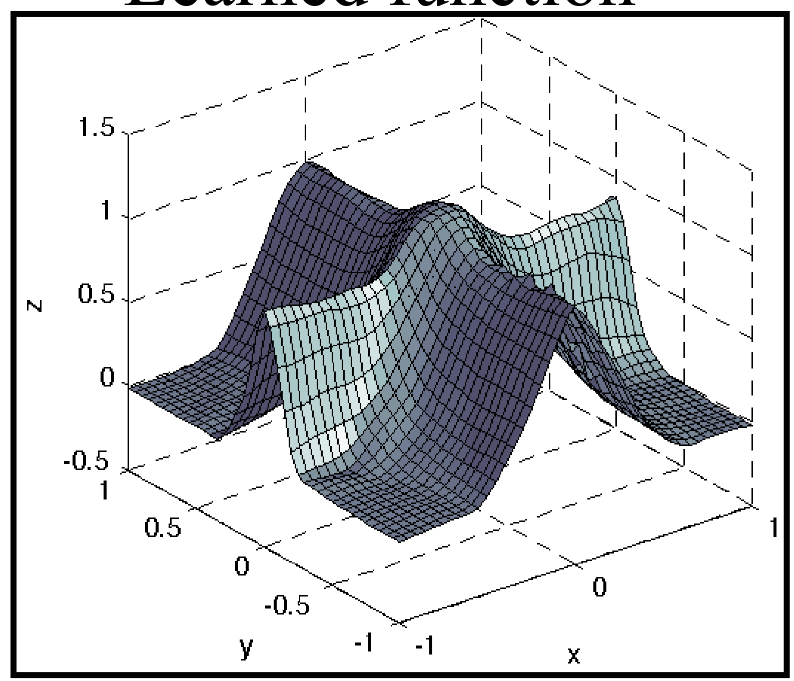
Input Dimensionality = 2 (+ 8 or 18 redundant dimensions.)

Noise ~ N(0,0.01) # training data = 500

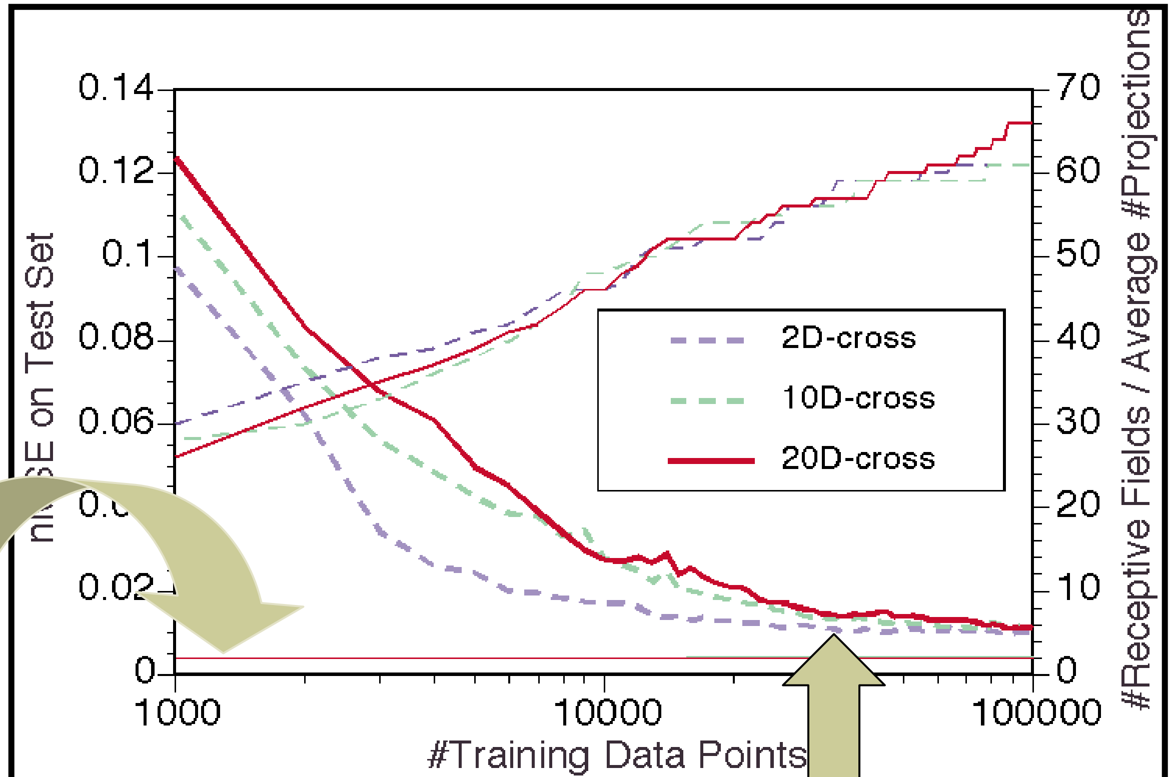
Learned Receptive Field



Learned function



Empirical Evaluations (Cross Data)



Correctly detects dimensionality of 2

Low nMSE achieved inspite of redundant inputs

Inverse Dynamics Learning (7 DOF)

- **Inverse dynamics of a 7DOF anthropomorphic**

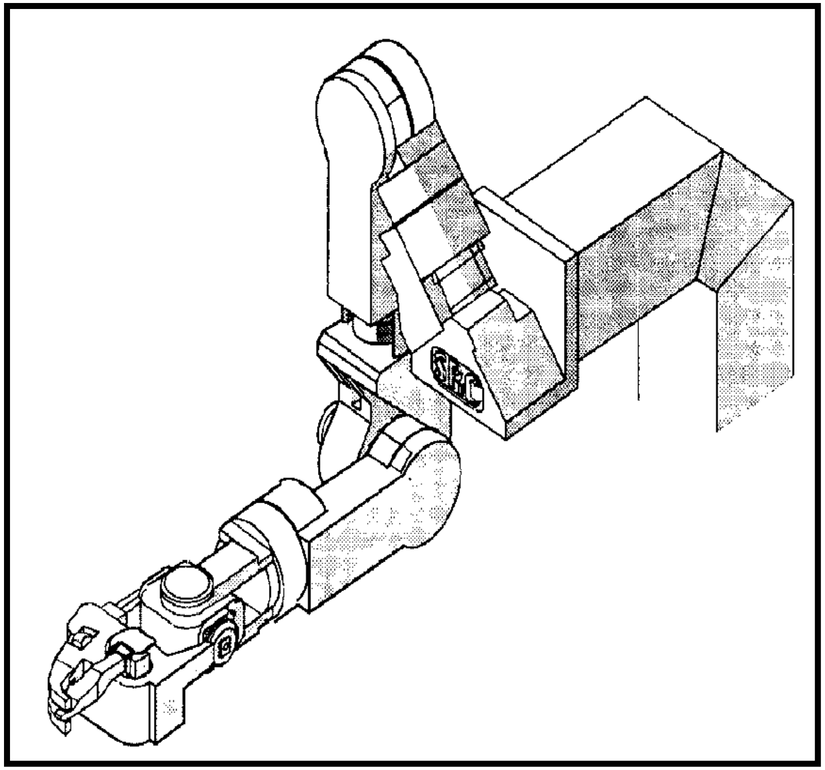
$$\tau = f(\theta, \dot{\theta}, \ddot{\theta})$$

$$f : \Re^{21} \rightarrow \Re^{7}$$

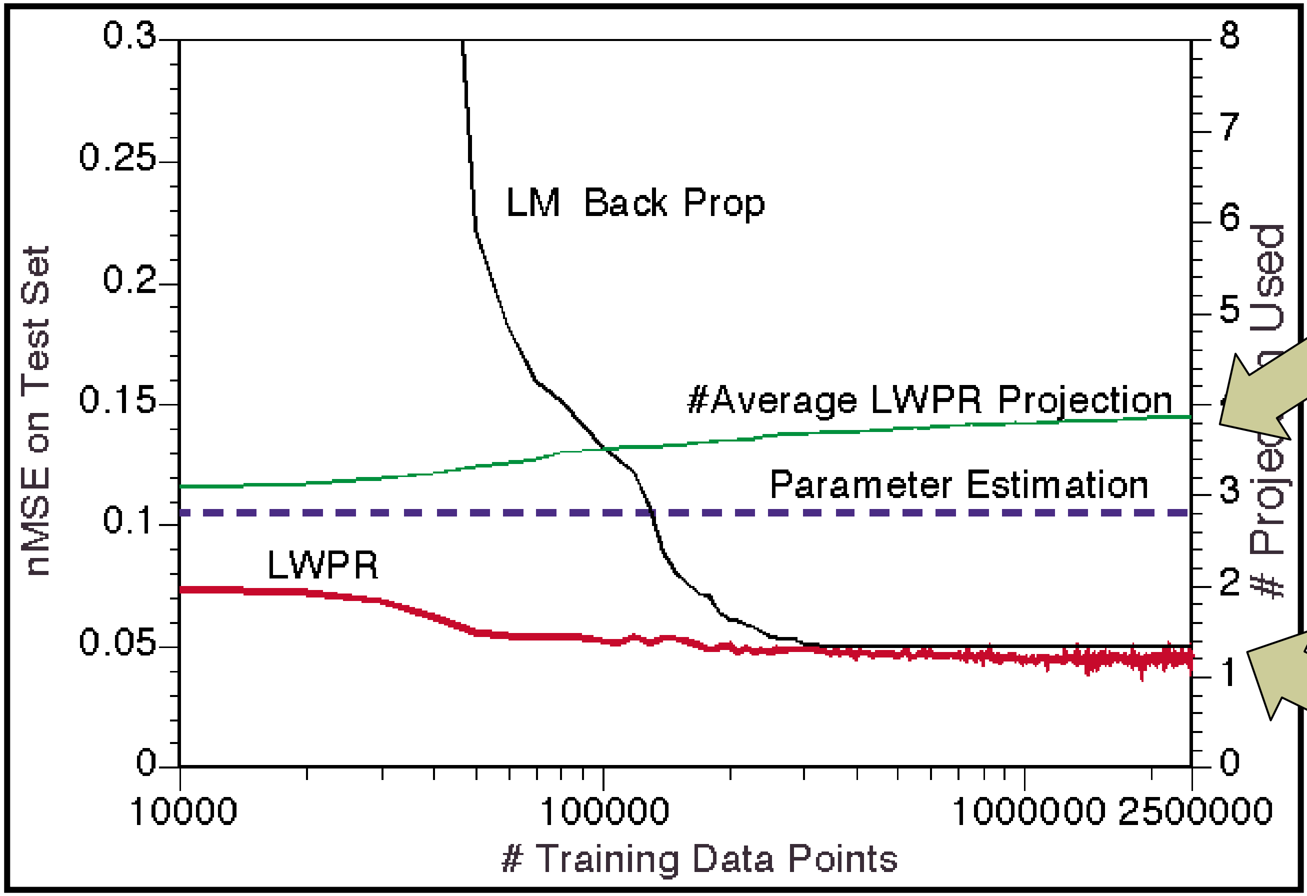
We add an additional 29 redundant input dimensions to verify the redundancy handling ability of our algorithm

$$f : \Re^{50} \rightarrow \Re^{7}$$

SARCOS dexterous arm



Learning results & comparison

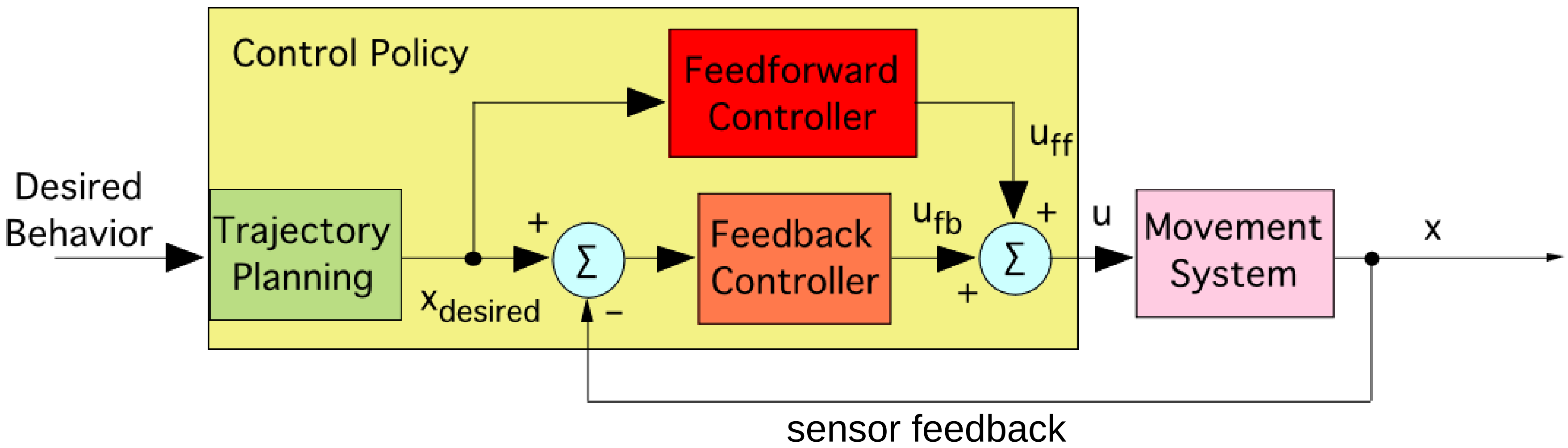


An average of only 4 projections used by LWPR

(input dim =50)

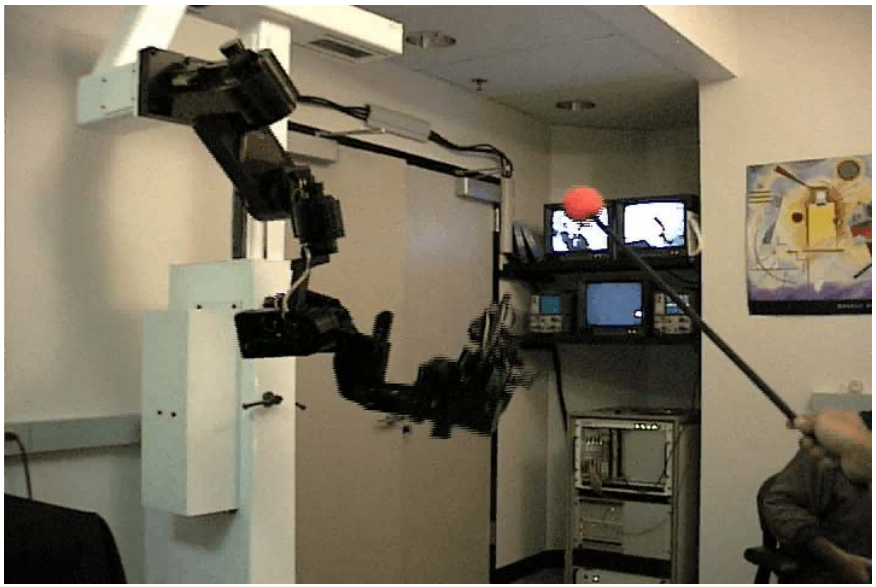
Excellent nMSE & fast convergence

Learning Feedforward Controller

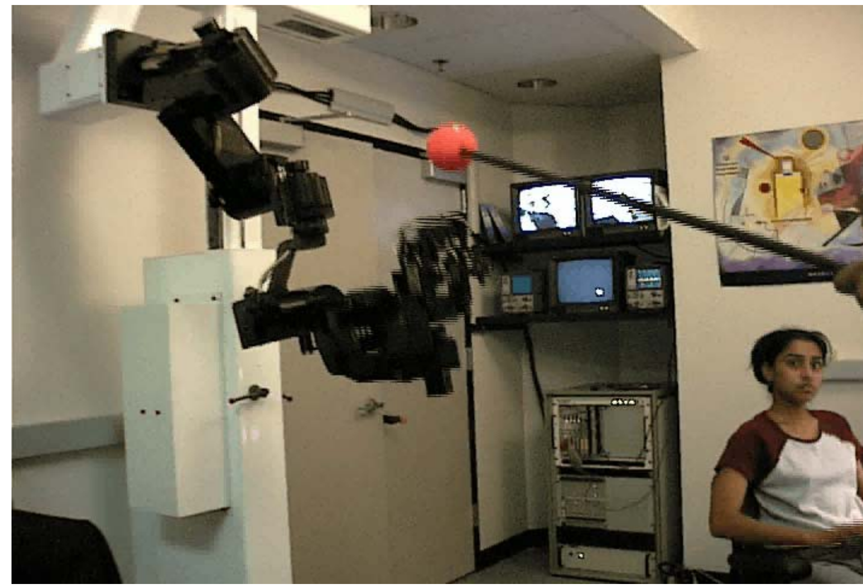


Online Learning with 7DOF Robot Arm

PD control – no dynamics model

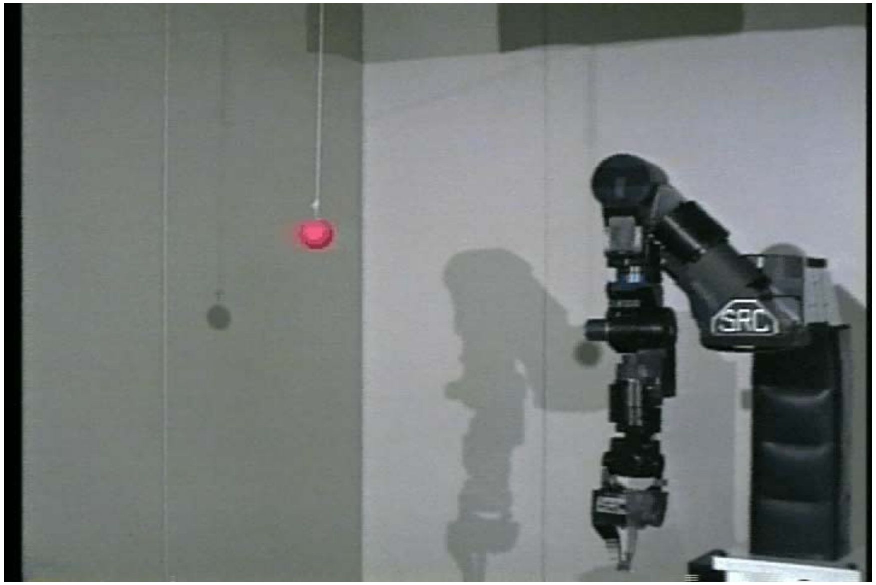


Learning the Dynamics Model

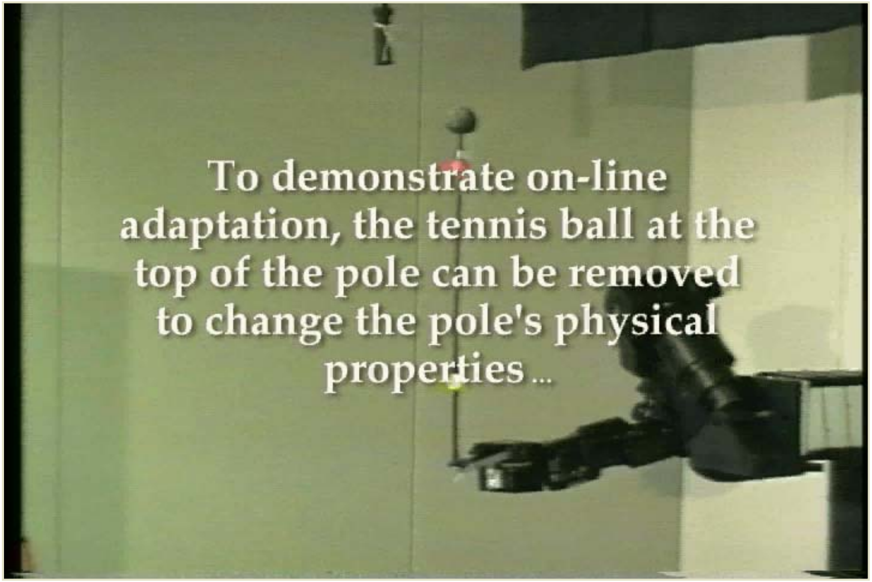


Online learning with Robot Arm (II)

Reaching for targets & pole balancing

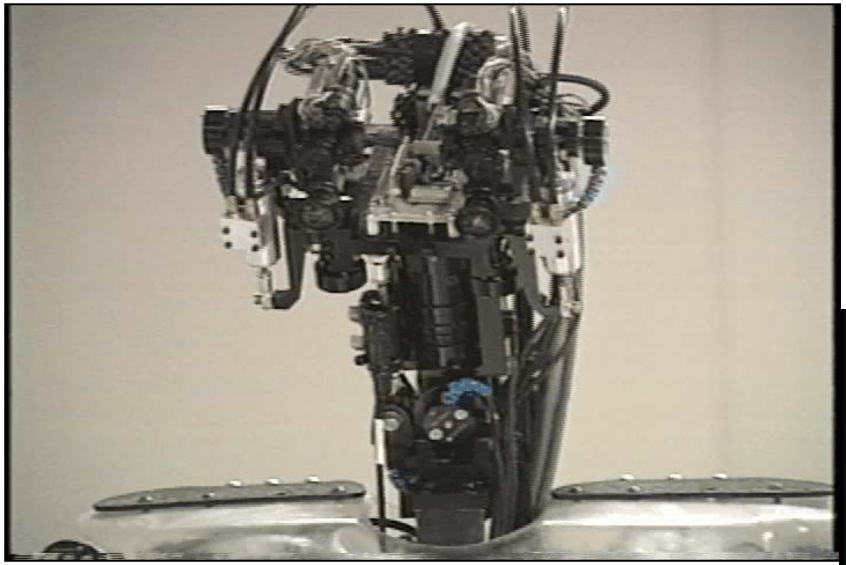


Adapting to changed dynamics



Real time learning with the Humanoid

Online Dynamics Adaptation



$$f : \Re^{90} \rightarrow \Re^{30}$$

Real Time Visual Attention