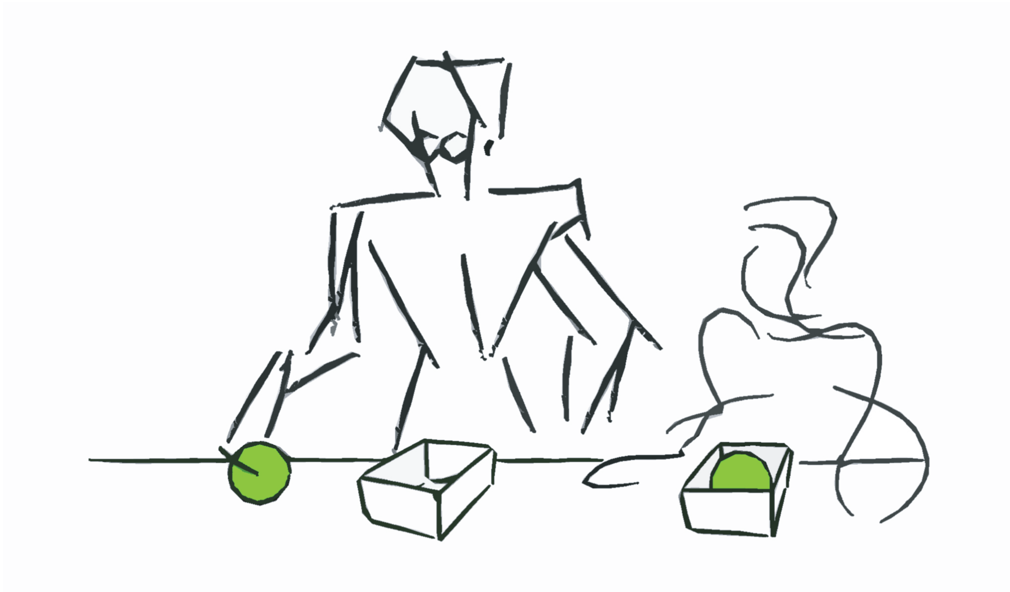
Incremental learning of motion primitives for full body motions

Dana Kulić

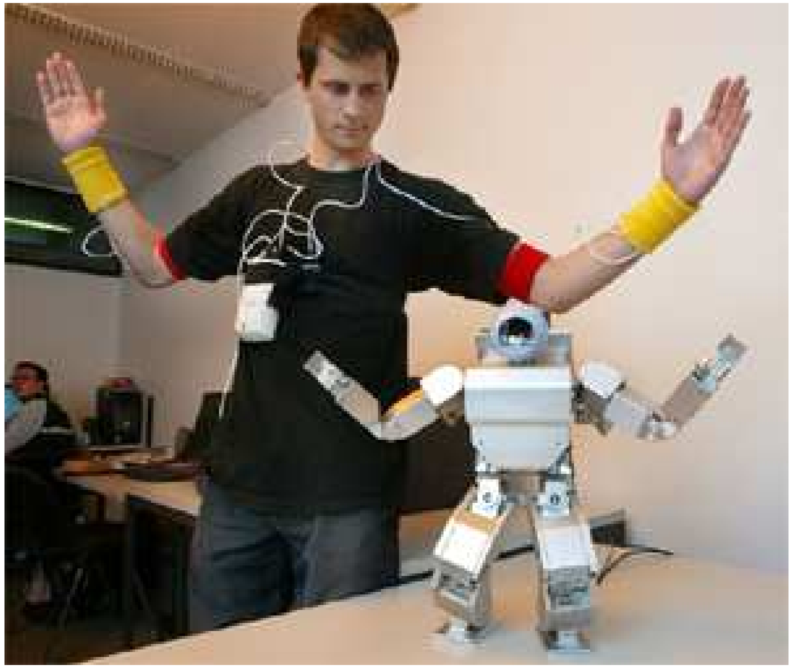
Department of Electrical and Computer Engineering, University of Waterloo, Canada

Learning from Observation for Humanoids



- Learn to accomplish tasks by observing a human teacher, rather than programming or trajectory planning
- Take advantage of similar structure between human and robot
- Suitable for non-expert demonstrators

Related Work



[Calinon and Billard 2007] HOAP at EPFL

[Ikeuchi et al. 2004] HRP-2 at AIST

Limitations of the current approaches

- Motions are specified manually by the designer
- In learning systems, motions are segmented and clustered a-priori
- Off-line, one-shot training
- No further learning during the execution stage

Desired System

- A robot that cohabits with humans, and learns incrementally over a lifetime of observations

- A robot that accumulates knowledge and improves performance over time

- Fully autonomous, on-line, continuous learning

System Requirements:

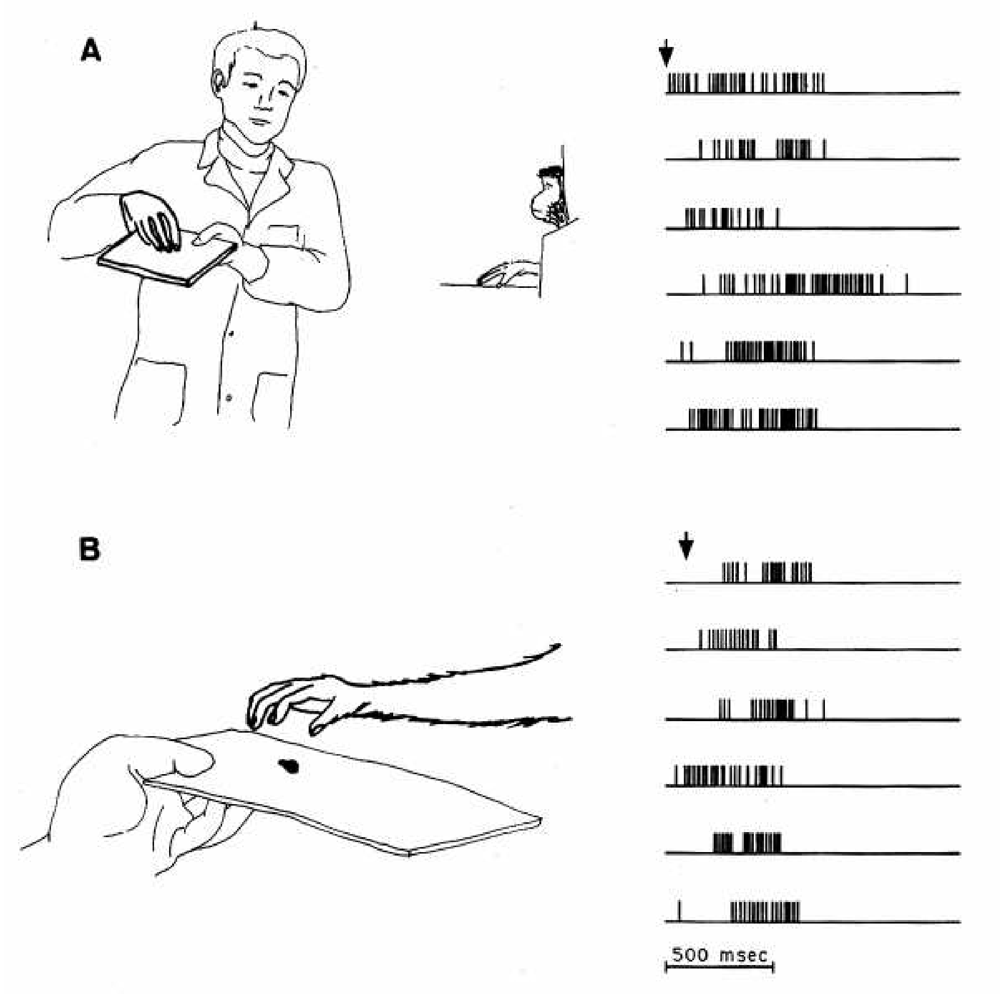
- Autonomous Motion Segmentation
- Autonomous, On-line Motion Clustering
- Autonomous Knowledge Management with fast Retrieval

Talk Outline

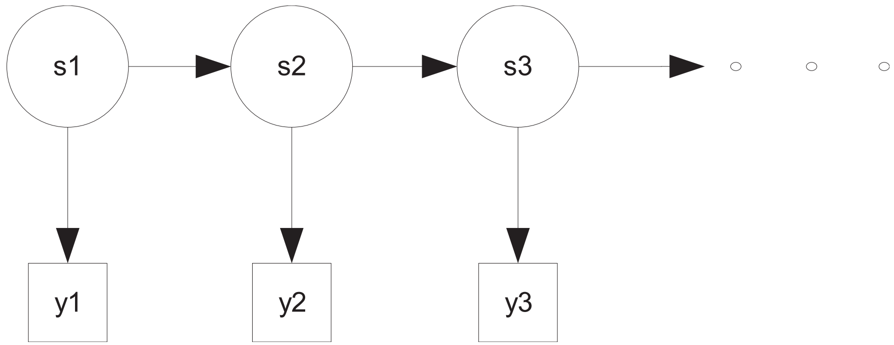
- Robot Learning from Observation
  - Representing full-body Motion
  - On-line Segmentation
  - On-line Clustering and Organization
  - Combining Segmentation and Clustering
  - Learning the sequencing of motion primitives
  - Incremental Memory Consolidation
- Conclusions and Directions for Future Work

Learning from Observation - Mirror Neurons

The same neural structure is used for both recognition and generation [Rizzolatti et al. 2001]



Motion Representation by Hidden Markov Models [Inamura et al. 2004]



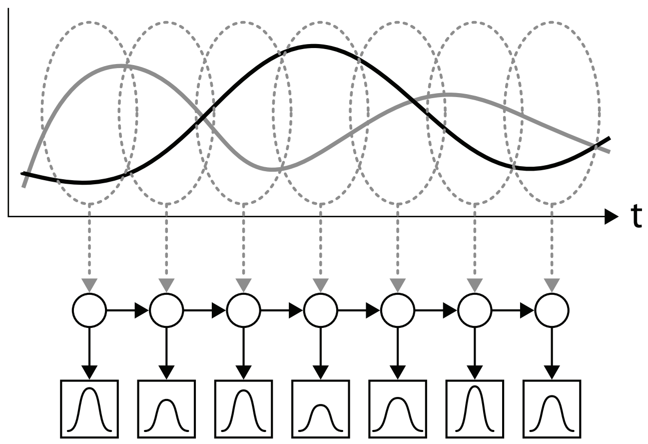
- Stochastic model capturing both spatial and temporal variability
- Model training (learning) is implemented with the Baum-Welch Algorithm
- Once the model is trained, the same model can be used for both
  - Recognition (Forward Procedure)
  - Generation (either stochastic or deterministic)
- Factorial HMMs also used for representing motions with greater accuracy [Kulić et al. IROS 2007]

On-line Segmentation

- Want to segment with no a-priori knowledge of the motions
- Must make some assumption about the structure of the data
  - Mean velocity falls below a certain value [Pomplun and Matarić, 2000]
  - Zero velocity crossing in some dimensions [Fod et al., 2002]
  - Minimize variance [Koenig and Matarić, 2006]
  - Same motion will belong to same underlying distribution [Kohlmorgen and Lemm, 2001] [Janus and Nakamura, 2005]

Stochastic Segmentation

[Kohlmorgen and Lemm, 2001]



Embed the data into a higher-dimensional space

$$\vec{x_t} = (\vec{y_t}, \vec{y_{t-1}}, \ldots, \vec{y_{t-(m-1)\tau}})$$

Estimate the density distribution over a sliding window of length W

$$p_t(\mathbf{x}) = \frac{1}{W} \sum_{w=0}^{W-1} \frac{1}{(2\pi\sigma^2)^{d/2}} exp(-\frac{(\mathbf{x} - \vec{x_{t-w}})^2}{2\sigma^2})$$

Computing the distance between states

Can compute the distance between windows based on integrated square error between two pdfs

$$d(p_{t1}, p_{t2}) = \int (p_{t1}(x) - p_{t2}(x))^2 dx$$

$$\begin{aligned} d(p_{t1}(\mathbf{x}), p_{t2}(\mathbf{x})) = & \frac{1}{W^2(4\pi\sigma^2)^{d/2}} \\ & \sum_{w,v=0}^{W-1} [exp(-\frac{(\vec{x_{t1-w}} - \vec{x_{t1-v}})^2}{4\sigma^2}) \\ & - 2exp(-\frac{(\vec{x_{t1-w}} - \vec{x_{t2-v}})^2}{4\sigma^2}) \\ & + exp(-\frac{(\vec{x_{t2-w}} - \vec{x_{t2-v}})^2}{4\sigma^2}) \end{aligned} \quad (1)$$

Defining a stochastic model over the data

Define an HMM over a set of sliding windows.

Observation Function:

$$p(p_t(\mathbf{x})|s) = \frac{1}{\sqrt{2\pi\varsigma}} exp(-\frac{d(p_s(\mathbf{x}), p_t(\mathbf{x}))}{2\varsigma^2})$$

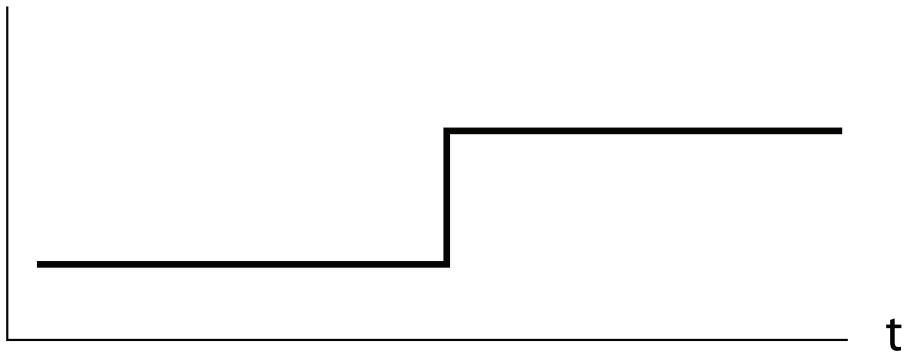
State Transition Model:

$$a_{ij} = \begin{cases} \frac{k}{k+N-1} & \text{if } i = j; \\ \frac{1}{k+N-1} & \text{if } i \neq j. \end{cases}$$

Segmentation via the Viterbi Algorithm

Find the optimum state sequence for the specified HMM, given the actual data sequence

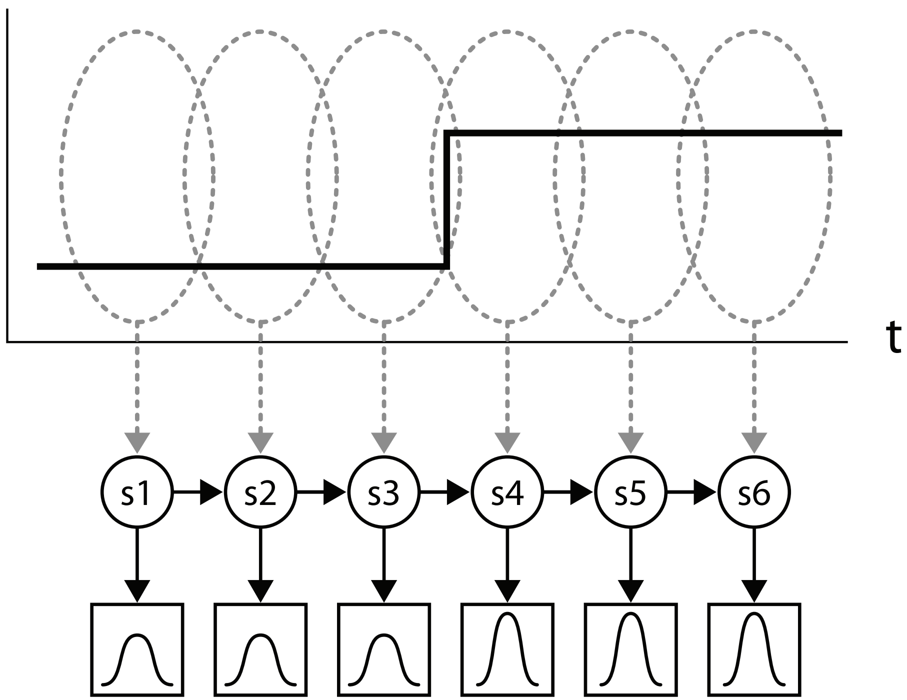
A simple example:



Segmentation via the Viterbi Algorithm

Find the optimum state sequence for the specified HMM, given the actual data sequence

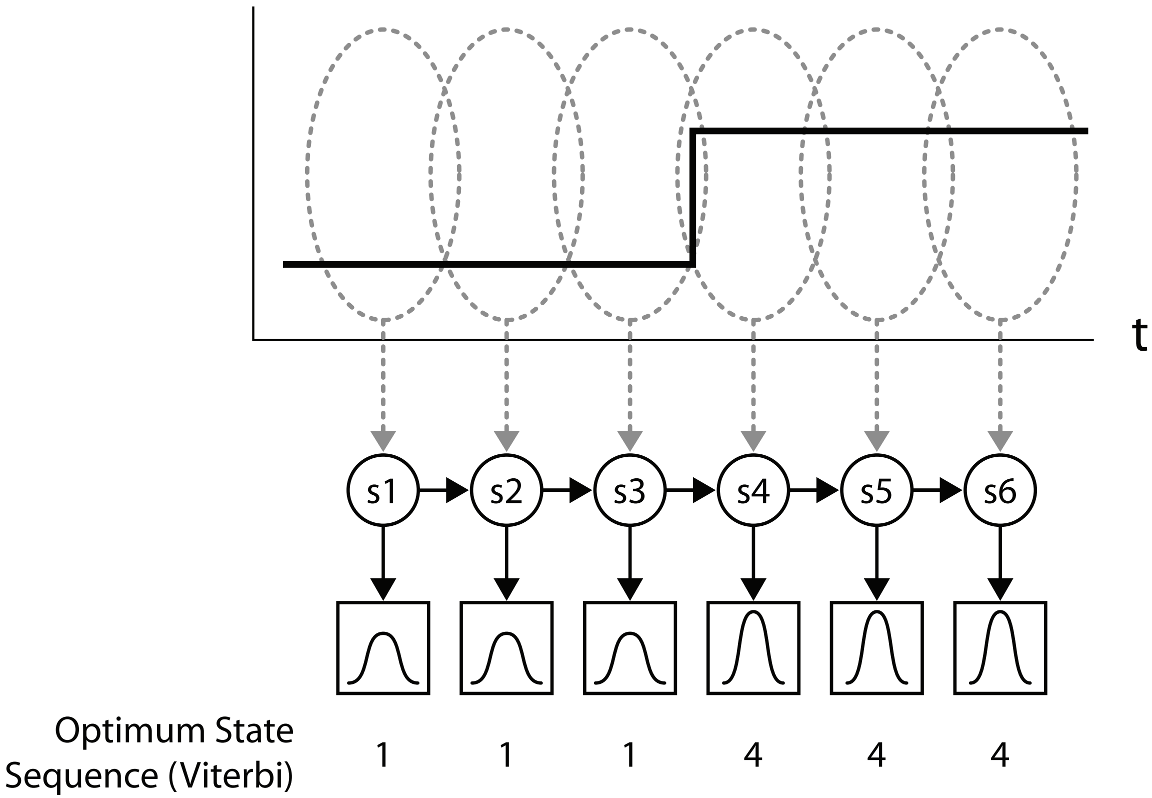
A simple example:



Segmentation via the Viterbi Algorithm

Find the optimum state sequence for the specified HMM, given the actual data sequence

A simple example:



Improving the Segmentation

[Kulić Nakamura IROS 2008]

Bias state transition model towards known states

$$a_{ij} = \begin{cases} \frac{k}{C} & \text{if } i = j; \\ \frac{1}{C} & \text{if } i \neq j \text{ and } i \in \mathcal{S}_t; \\ \frac{K_s}{C} & \text{if } i \neq j \text{ and } i \in \mathcal{S}_p. \end{cases}$$

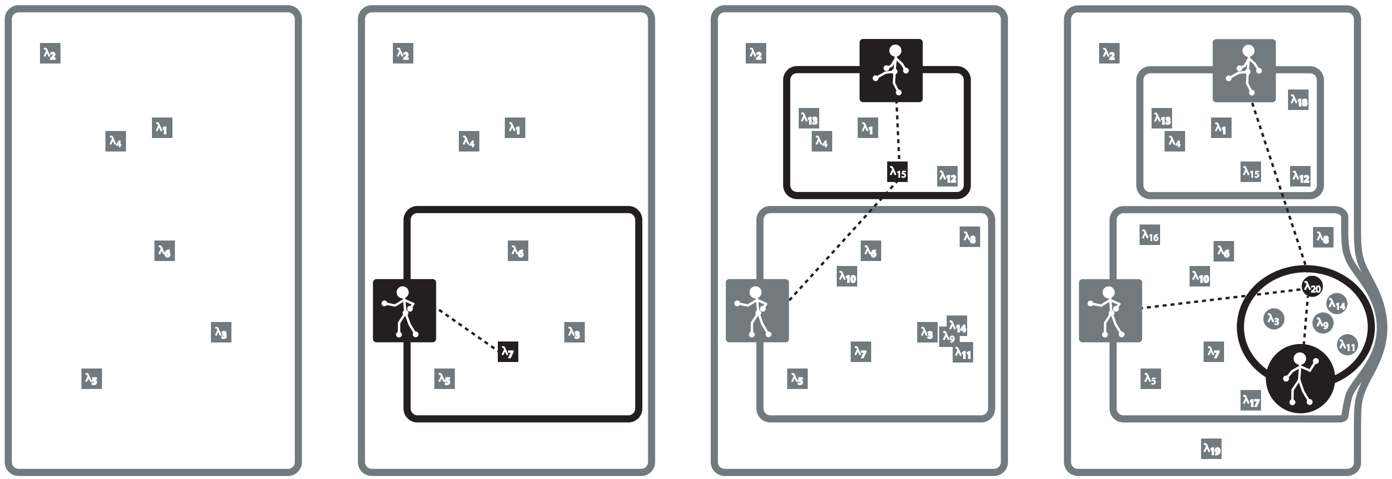
Modify pdf based on active joints in the known state

$$\begin{aligned} D_w(p_{t1}(\mathbf{x}), p_{t2}(\mathbf{x})) = \frac{1}{L^2(4\pi\sigma_k^2)^{d/2}} \\ \sum_{i,j=0}^{L-1} [exp(-\frac{W(x_{t\vec{1}-i} - x_{t\vec{1}-j})^2}{4\sigma_k^2}) \\ - 2exp(-\frac{W(x_{t\vec{1}-i} - x_{t\vec{2}-j})^2}{4\sigma_k^2}) \\ + exp(-\frac{W(x_{t\vec{2}-i} - x_{t\vec{2}-j})^2}{4\sigma_k^2}) \quad (2) \end{aligned}$$

On-line clustering and hierarchy formation

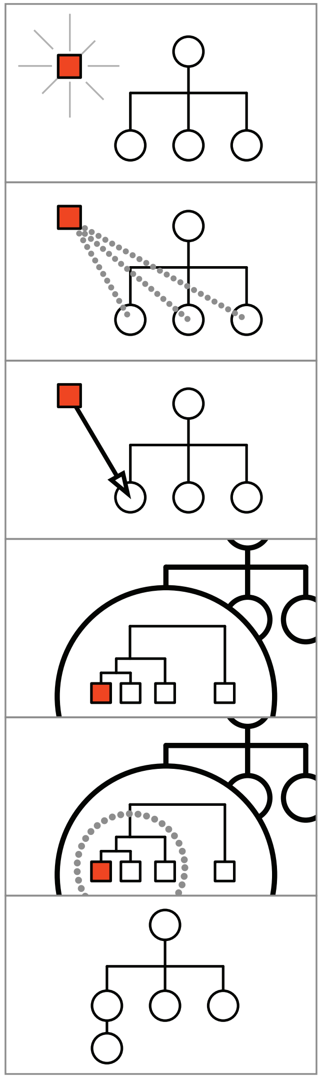
[Kulić et al. ISRR 2007, IJRR 2008]

- Use HMM representation to abstract motion patterns as they are perceived
- Cluster individual motion patterns incrementally, based on intra-model distances
- Use formed clusters to form group models
- Autonomously select appropriate model type, based on model distances in the considered region of the motion space



Algorithm Pseudo-Code

Following observation of each motion sequence:



Step1 Encode observation sequence O_i into an HMM λ_i

Step2 Calculate the distance between λ_i and each existing behavior group model λ_{Gj}

Step3 Place λ_i into the closest group G_c

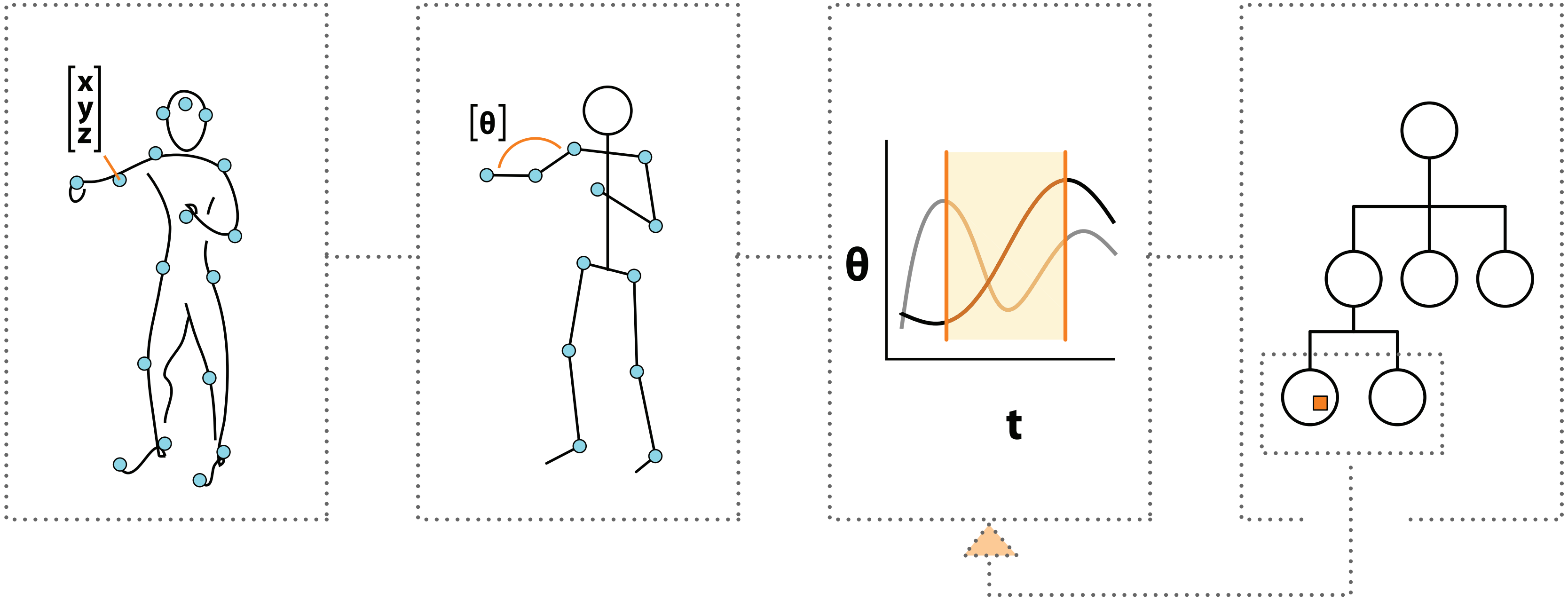
Step4 Cluster all exemplars within G_c

Step5 If a sub-group forms, form a new node G_n, containing the exemplars of the cluster

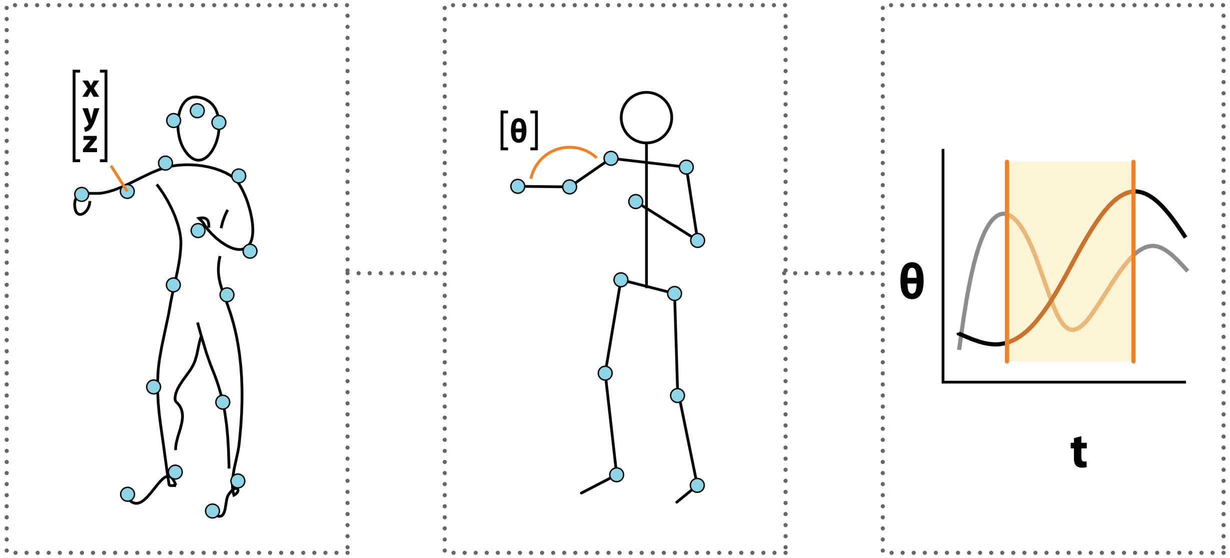
Step6 Using the observation sequences from the exemplars in G_n, form the new sub-group model λ_{Gn}

Combining segmentation and Clustering

[Kulić et al. ICRA 2008]

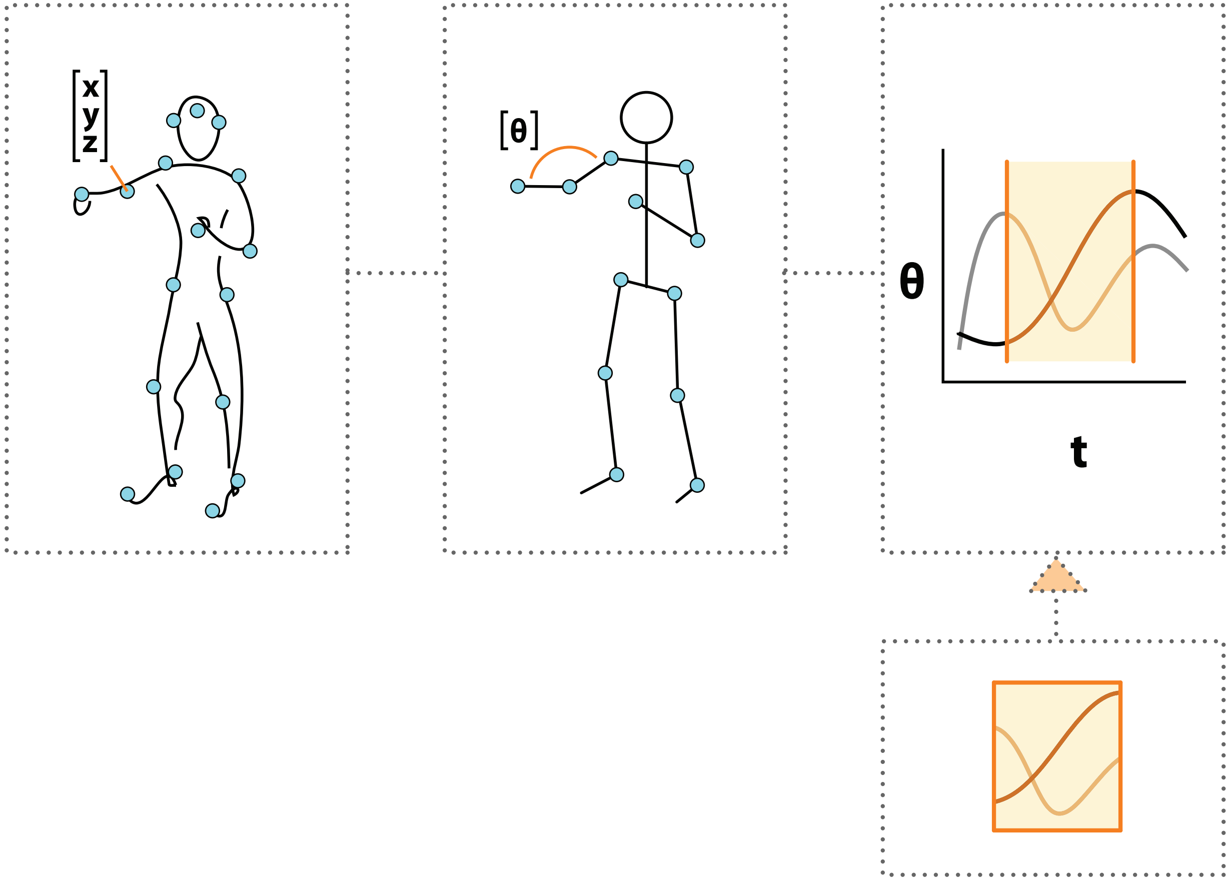


Experiments



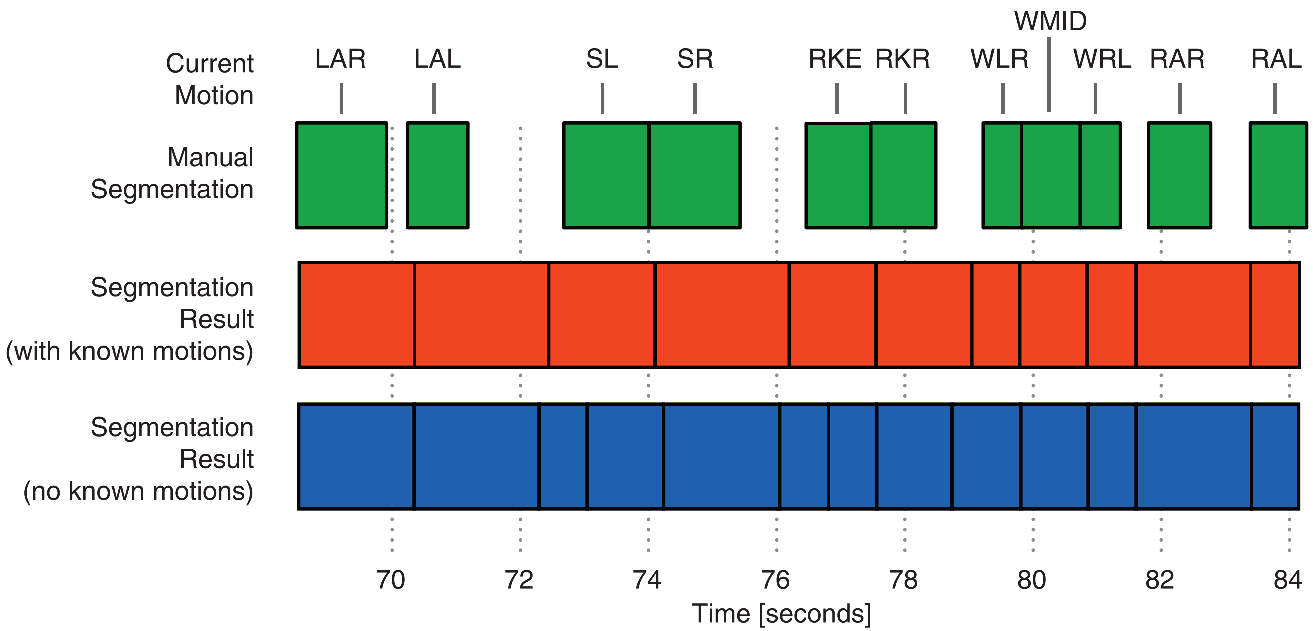
- 4 minutes of continuous whole body motion data of a single subject from motion capture data
- data is converted to a 20DoF humanoid model by online inverse kinematics
- First, test the basic segmentation algorithm, with no known states, and compare with manual segmentation

Testing the Segmentation



- Next, test the improvements obtained through adding known motions
  - Provide manually extracted primitives as exemplars

Segmentation Results

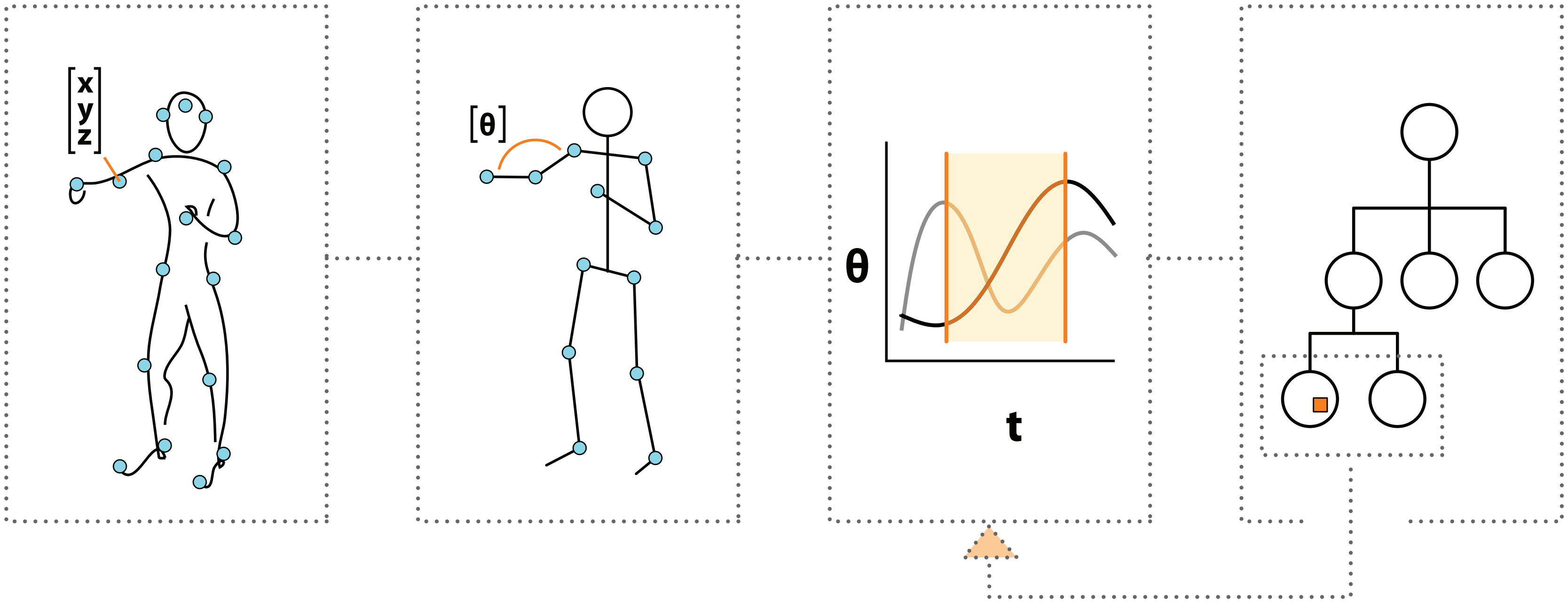


Algorithm	Correct	False Pos	False Neg
Basic	128	65	43
Scaffolded (with Squat and Kick)	139	59	32

- Worst performance occurs at switching points where few joints are moving

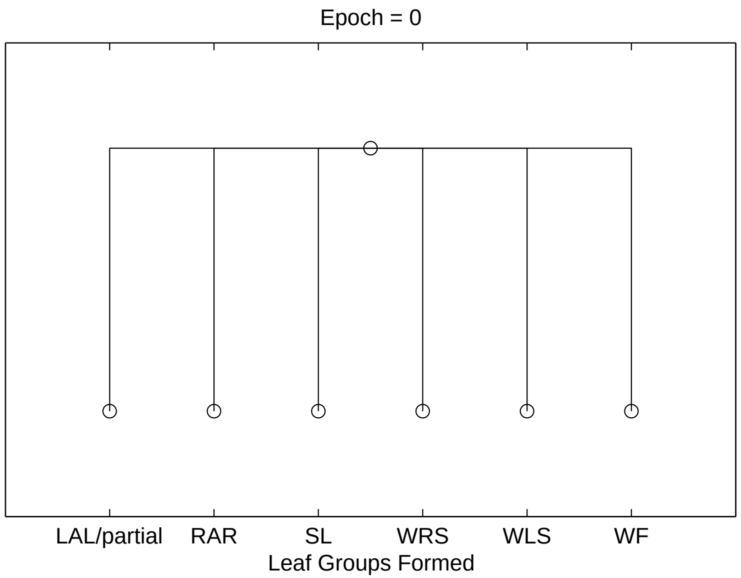
Sample Video

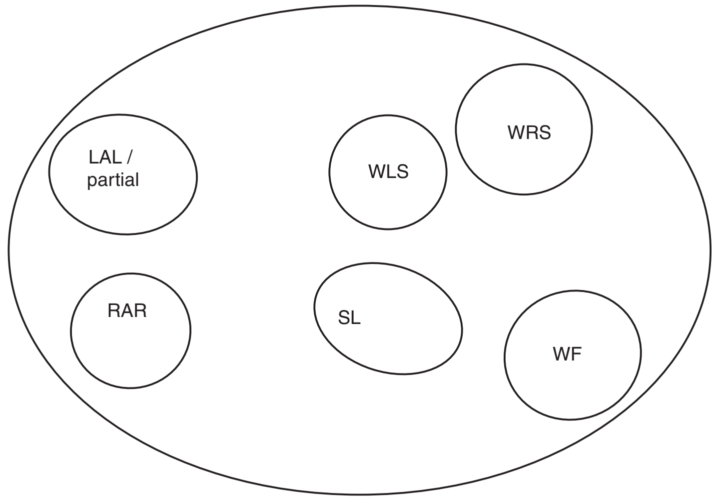
Testing the Combined Segmentation and Clustering



- Present the complete 4min sequence and apply segmentation
- The leaf nodes of the resulting tree are used to scaffold the segmentation
- To facilitate analysis, 4min sequence is presented repeatedly (epochs), and new exemplars are added to the segmentation module at the end of each epoch

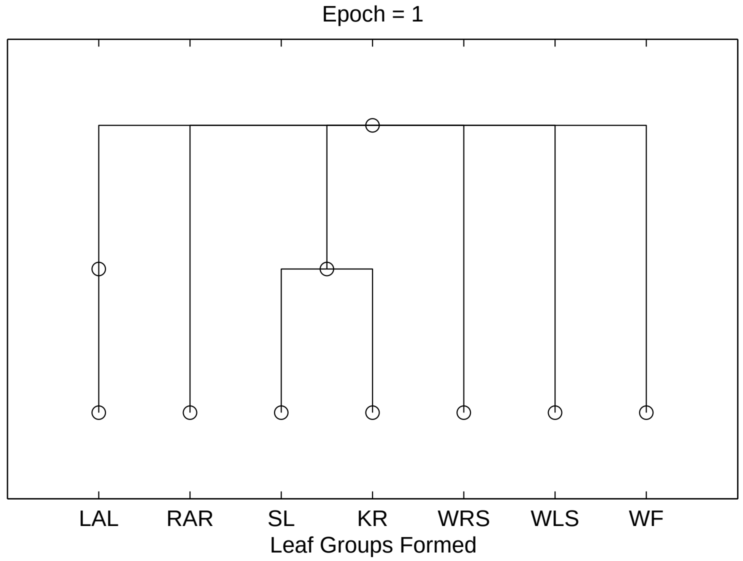
After Epoch 1

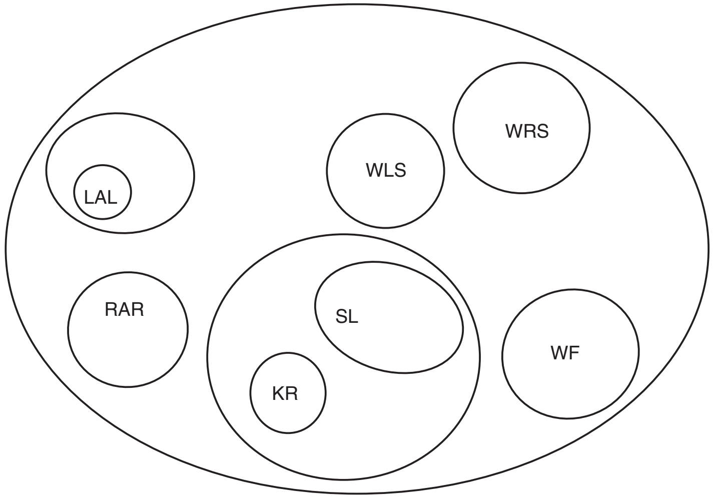




Example Extracted Motion: Right Arm Raise

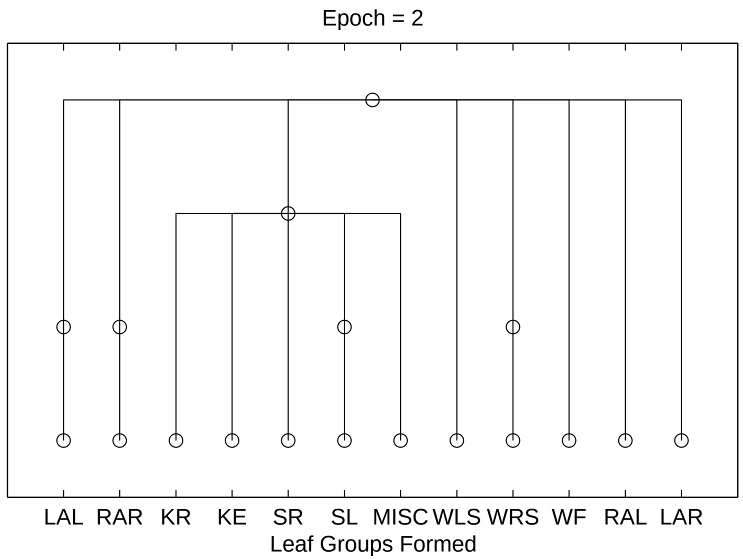
After Epoch 2

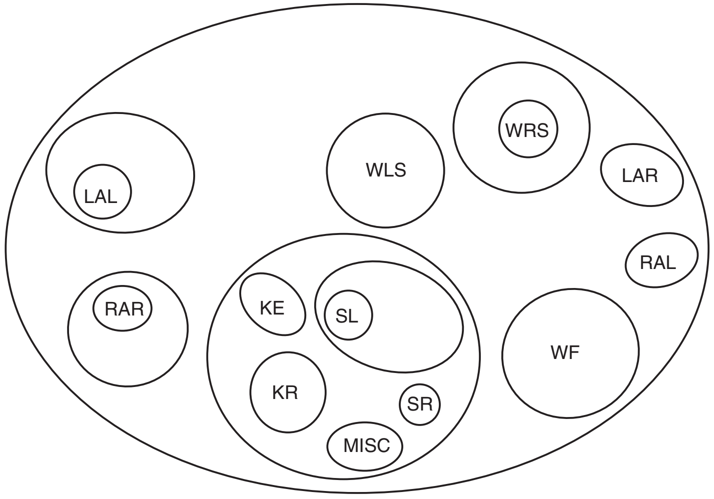




Example Extracted Motion: Left Arm Lower

After Epoch 3

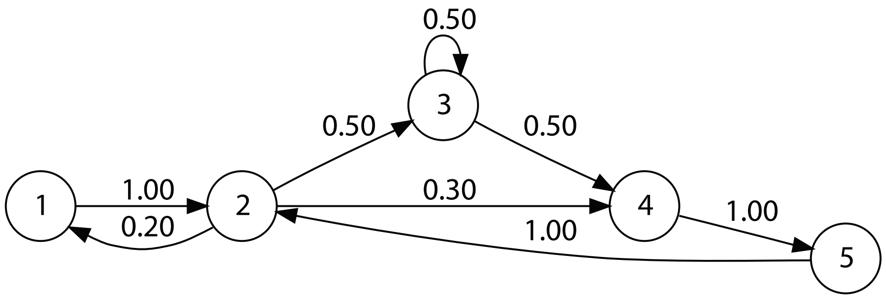




Example Extracted Motion: Kick Extend, Squat Raise

Motion Primitive Graph

[Kulić et al. Humanoids 2008]



- At the same time as learning the motion primitives, learn the transition rules between primitives
- Each node in the motion primitive graph represents a motion primitive, while each edge represents an observed transition between two motion primitives
- Each time a new motion primitive is abstracted by the clustering algorithm as a leaf node, a corresponding node is added to the motion primitive graph.
- Each time a transition is observed between two known motions, the edge count is updated

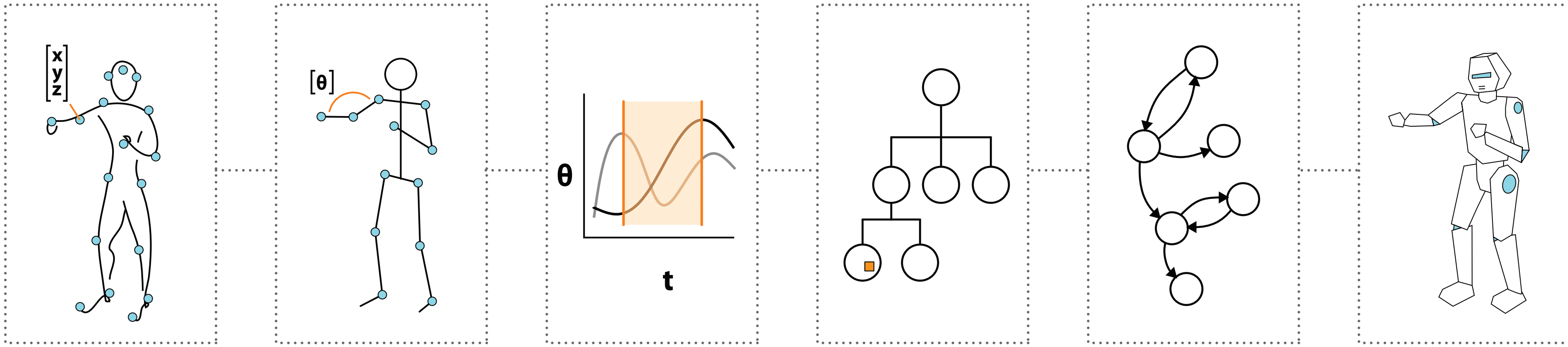
Experiments with a Humanoid Robot

Collected 16 min of continuous whole body motion data (26 different motion types) of a single subject from motion capture data

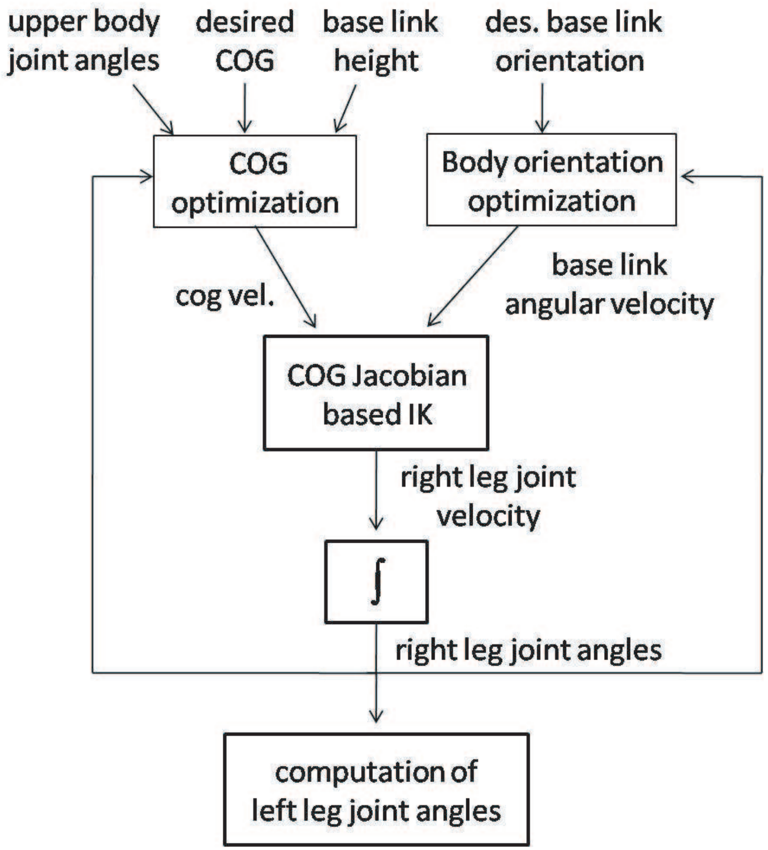
data is converted to a 32DoF humanoid model by online inverse kinematics

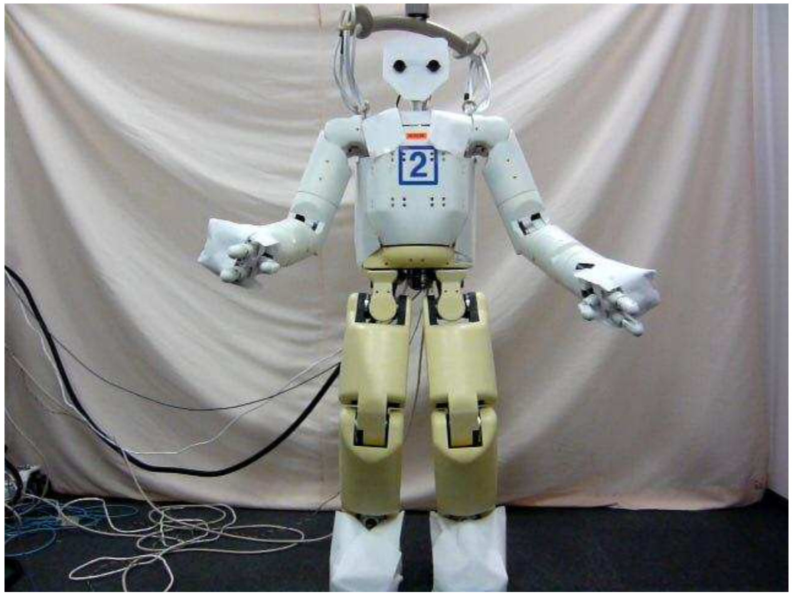
online feed to automated segmentation, clustering and motion graph extraction

Data Flow Diagram

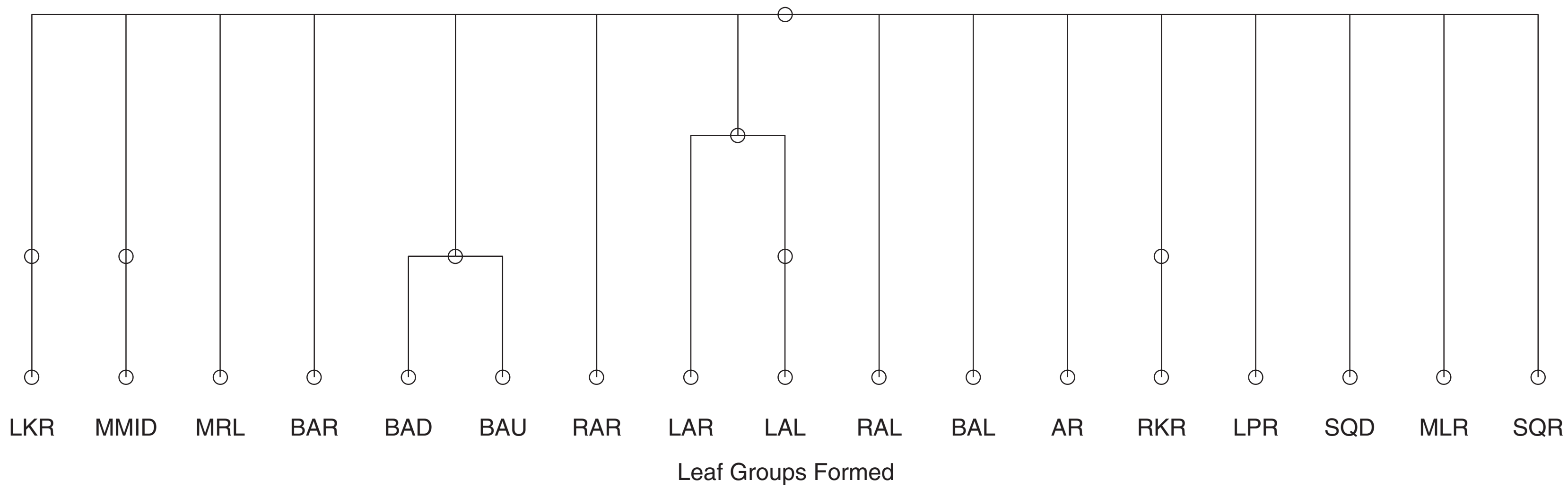


Robot Hardware and Control System

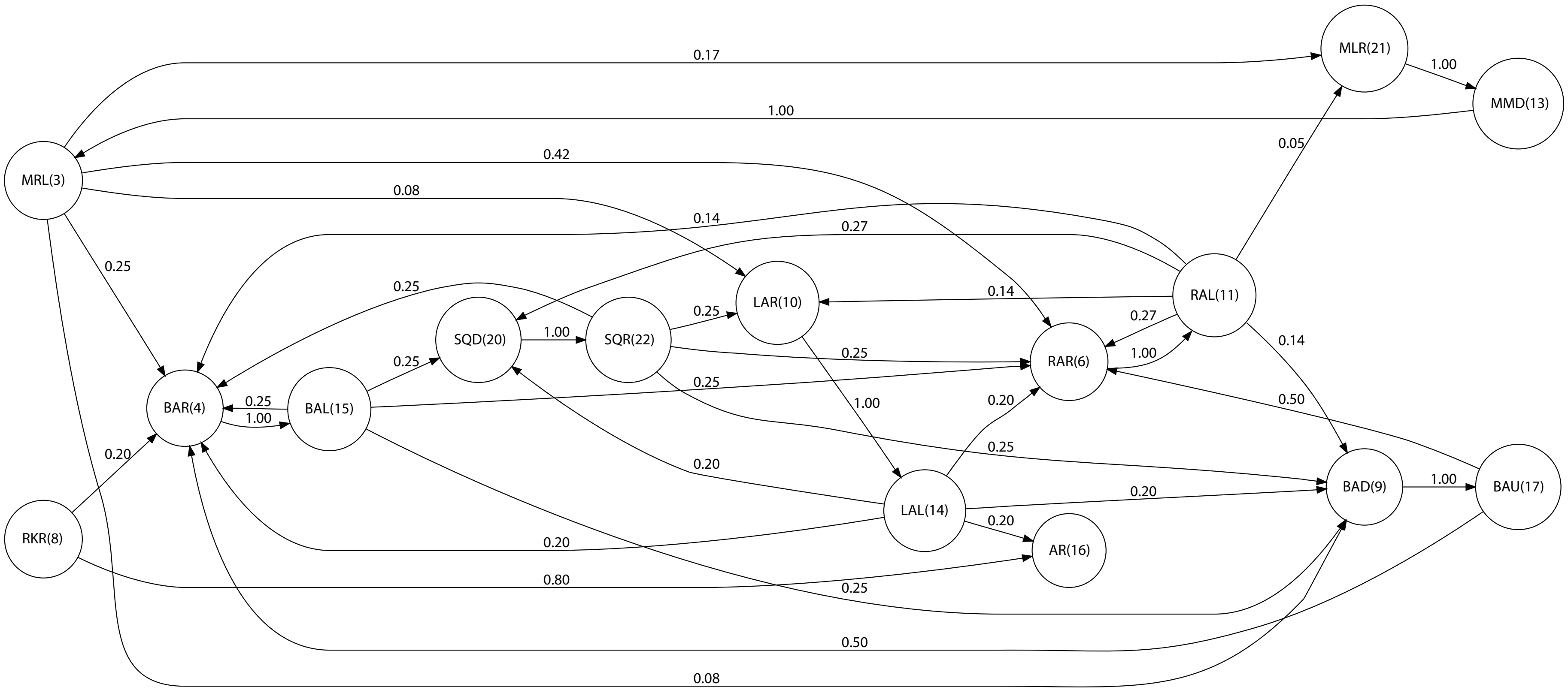




The Extracted Motion Primitive Tree

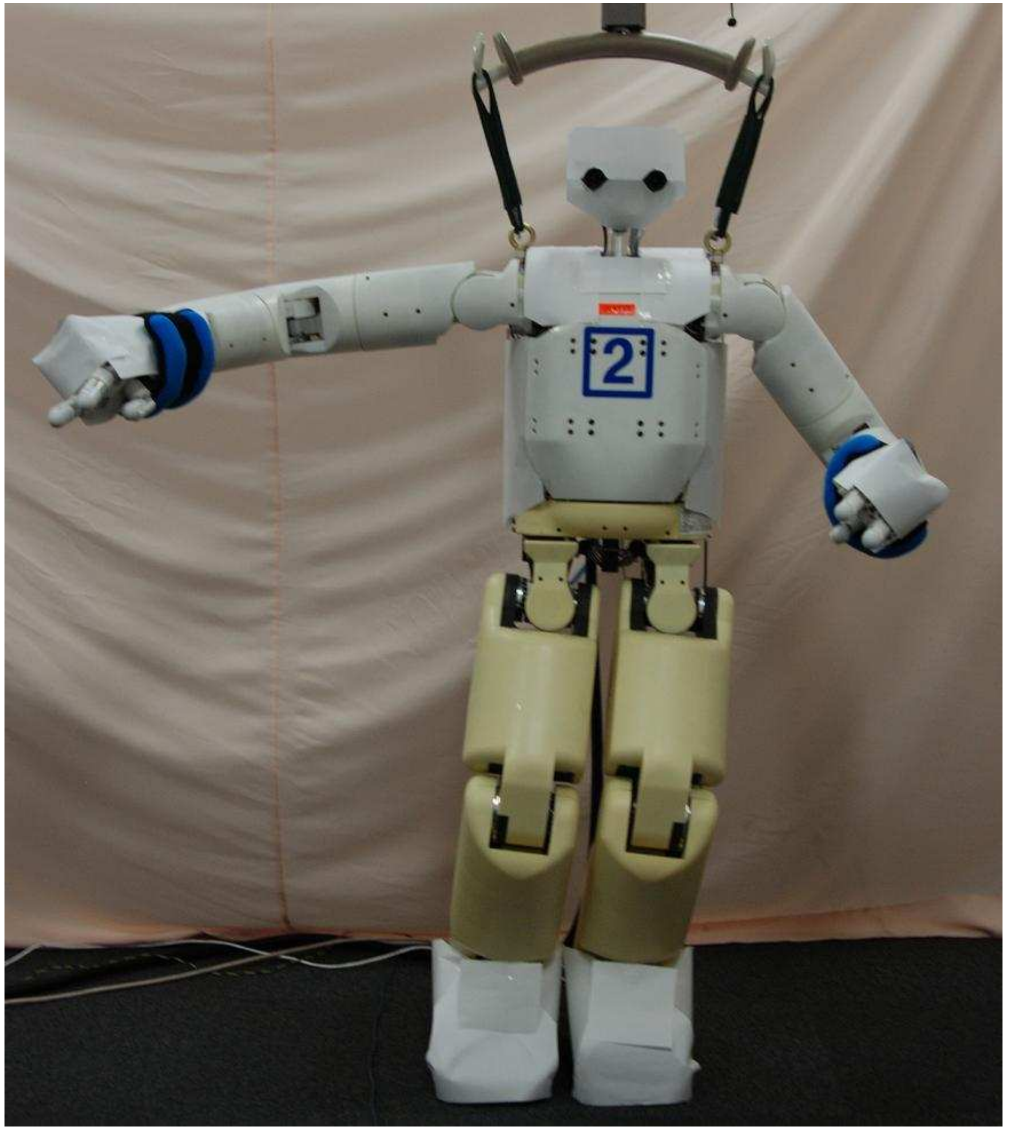


The Extracted Motion Primitive Graph



- Due to current hardware limitations of the robot, motions involving foot raising are manually removed from the graph

Robot Motion Generation



Video of Experiment

Summary on Automated Segmentation

- Autonomous, on-line segmentation of full body motion data, by building an HMM over a window of previous observations, and finding the optimum state sequence [Kohlmorgen and Lemm]
- Input segments into automated incremental clustering algorithm for motion primitive extractions
- Improve segmentation results by scaffolding with known motion primitives obtained from the clustering
- As more motions become known, motion model and segmentation results become more accurate
- At the same time, learn the transition model of the motion primitives by constructing a motion primitive graph

Memory Consolidation

- Want to re-examine the performance of the clustering algorithm
- What type of errors can occur during clustering, and how do these errors depend on the algorithm parameters?
  - false negative errors
  - tree structure errors
- Both errors occur due to the incremental nature of the algorithm, where not enough information is available at the start of the algorithm to identify the correct segmentation boundary

Memory Consolidation in Biological Systems

- How is motor memory formed in biological systems?
- Following learning, the motor memory does not remain constant, but changes over time - *memory consolidation* [Stickgold, 2005] [Krakauer and Shadmehr, 2006] [Shadmehr and Holcomb, 1997] [Diekelmann and Born, 2007]
- Two Complementary Consolidation Processes:
  - Stabilization Stage (the waking stage)
  - Sleep-dependant Stage (occurs during sleep)
- During the sleep-dependent stage, brain imaging studies show that brain regions active during memory formation are repeatedly reactivated - *rehearsal* [Ogata, 2005]
- During the sleep-dependent stage, evidence of system-level reorganization of memory

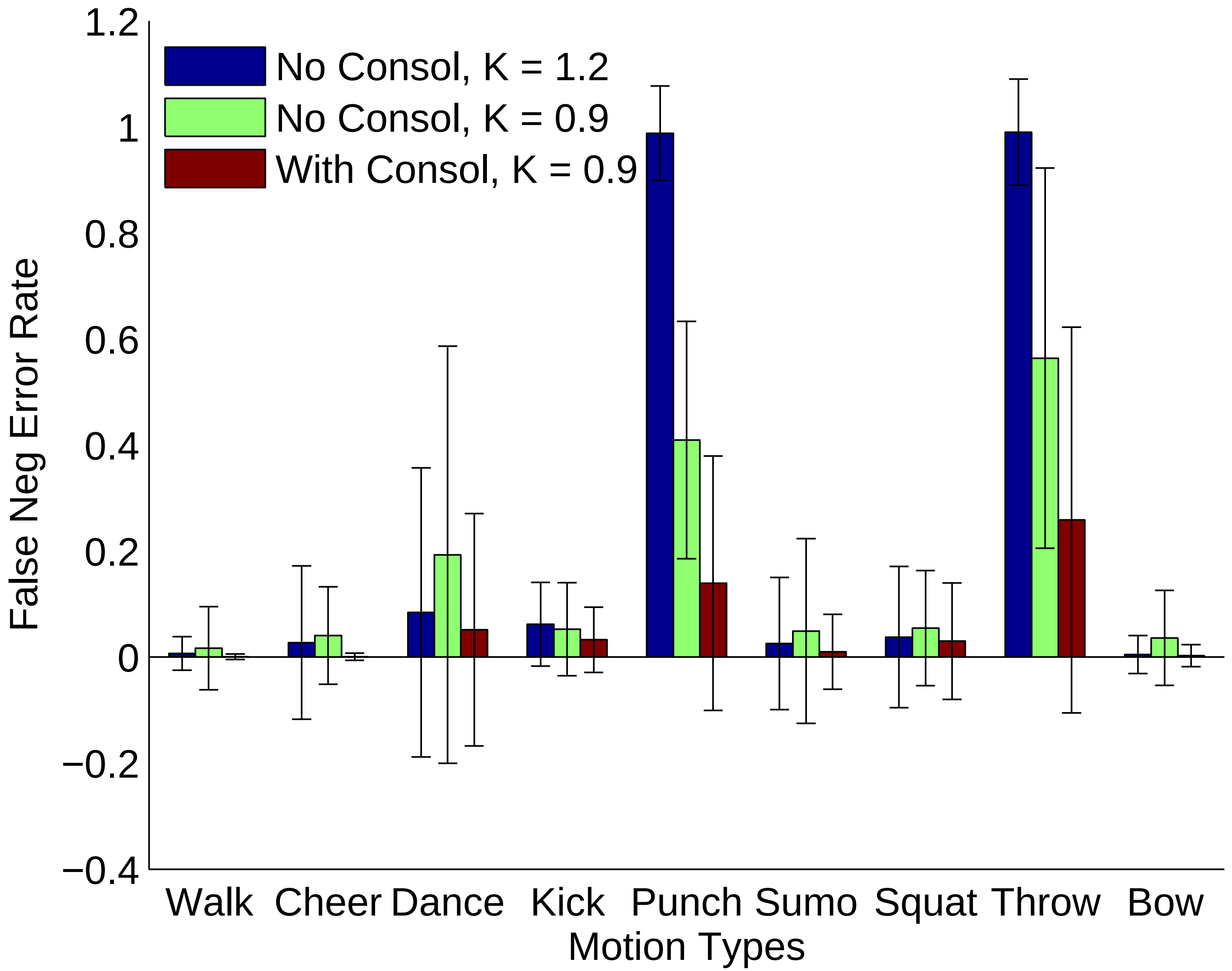
Using Memory Consolidation for Motion Learning [Kulić Nakamura EpiRob 2008]

- In previous work, focused on getting high accuracy at the leaf nodes by using adaptable models [Kulić et al., 2007]
- However, this may not always be the best approach
  - Results in flat tree structure
  - Delays node formation
- Alternate approach: form nodes quickly (with lower K_{cutoff}), and correct any errors later using memory consolidation
- Repeatedly apply clustering procedure on the same data at a later time - *rehearsal*
- As more data become available, initial mistakes can be corrected in an incremental, on-line fashion, analogous to memory consolidation in biological systems
- Two levels of consolidation: individual motion level, node level

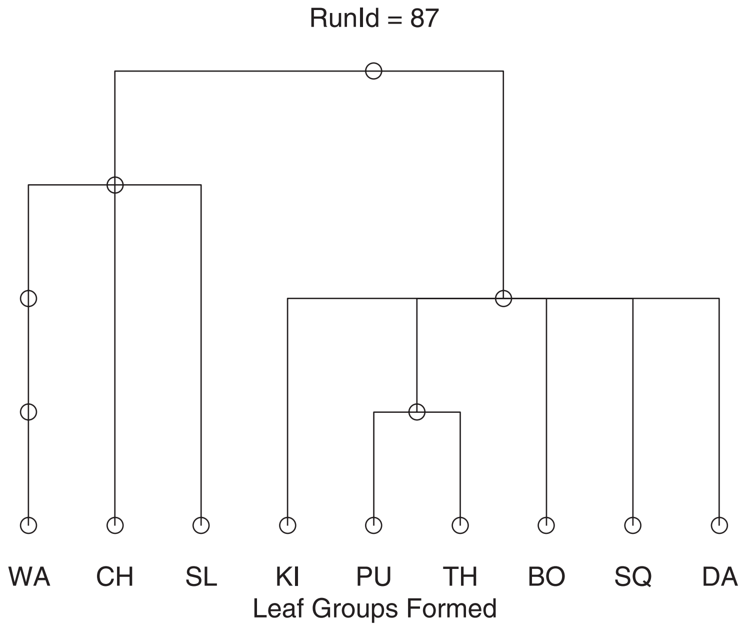
Experiments

- Tested on a motion database of 137 motions (9 types: KIck, PUnch, THrow, WAalk, Sumo Leg raise, CHeer, SQuat, BOw and DAnce)
- Data obtained from motion capture studio, converted to 20DoF humanoid model
- Motions are pre-segmented and presented in random order
- Tested with no consolidation and with consolidation
- Consolidation was executed after each 10 exemplars

Results: False Negative Errors, Comparison



Results: Tree Structure Errors



K_{cutoff}	Consolidation	Mean Error	Mean Depth
1.2	No	2.11	2.08
0.9	No	1.96	2.94
0.9	Yes	1.21	3.81

Summary

1. Motion primitives are autonomously segmented by building an HMM over a window of previous observations, and finding the optimum state sequence over the model
2. Segmented motions are incrementally clustered and organized into a hierarchical tree structure, with the leaf nodes representing the most detailed representation
3. At the same time, learn the transition model of the motion primitives by constructing a motion primitive graph
4. The algorithm is able to autonomously extract motion primitives from a continuous data stream, and is robust to segmentation and real-time measurement errors
5. Generated motion graph can then be used to generate extended motion sequences composed of motion primitives

Current and Future Work

- Getting away from motion capture and using simpler sensors
  [Kulić et al. ICRA 2009]
- Examining system performance for higher accuracy kinematic models
  [Kulić Nakamura IROS 2009]
- Including additional learning modalities: learning from practice and interaction with the teacher
  [Kulić et al. RO-MAN 2009]
- Applications for rehabilitation and sports training
  [Kulić et al. EMBC 2009]
- Incorporating interaction with the environment
- Selecting the correct task representation
- Planning with motion primitives
- Learning complex behaviors from the motion primitives

The End

Questions?

Additional Questions or Comments?
Email: dkulic@ece.uwaterloo.ca

Copies of publications can be obtained from:
http://ece.uwaterloo.ca/∼dkulic