Participant Identification in Haptic systems using HMM

Yednek Asfaw, Mauricio Orozco, Shervin Shirmohammadi,
Abdulmotaleb El Saddik, Andy Adler

SITE 2005

Outline

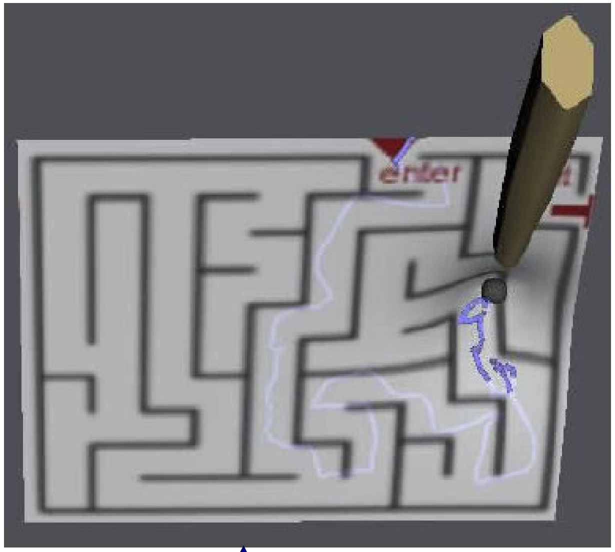
- Biometrics in Haptic systems
- Hidden Markov Model
- System design and Training
- Participant Identification
- Results
- Discussion

User identification in Haptics

- Common Biometric systems use:
  - Fingerprint, Face, Iris, Voice…
- All based on unique features of individuals
- Haptic systems introduce:
  - Sense of touch, force and hand kinetics in human-computer interface
  - Possible unique features could be associated with each user
  - Continuous authentication during the life time of a task

User identification in Haptics

- Haptic-Based application
  - 3D elastic membrane maze
- Using Reachin Display system
  - Phantom, Display, and stereo glasses



User identification in Haptics

- Objective is to investigate:
  - Is it possible to model small portions of a task using HMMs applied to raw sensor data?
  - Is there a strong connection between the user and the model to allow for identification?
- Important to know:
  - HMM structure
  - Number of states
  - Number of output parameters

Hidden Markov Models

- Using segmented training data:
  - Train an HMM that is the most likely set of transition probabilities
- Using previously unseen data:
  - Classify it to a particular HMM based on output parameters
- Theory of training and applying HMMs
  - Baum-Welch algorithm for training
  - Forward-Backward algorithm for testing

Baum-Welch algorithm

- Given initial estimate of the optimized Hidden Markov Model $\lambda = (A, B, \pi)$
- Generate a new estimate $\lambda_1 = (A_1, B_1, \pi_1)$ such that:

$$\Pi_i \; P(\lambda_1 | \, O(n)) \geq \Pi_i \; P(\lambda | \, O(n))$$

- Maximized via the EM algorithm using the entire training data set

Forward-Backward algorithm

- Given a HMM model ($\lambda_1 = (A_1, B_1, \pi_1)$) for each user
  - Determine the probability of a data set belonging to a model - $P(O \mid \lambda)$
- probability of observing the partial sequence $o_1, \ldots, o_t$ and resulting in state i at time t:

$$\alpha_i(t)=P(O_1=o_1, \ldots ,O_t=o_t,Q_t=i|\lambda)$$

Forward-Backward algorithm

- $P(O \mid \lambda)$ is determined as a sum of above probability which is determined recursively:

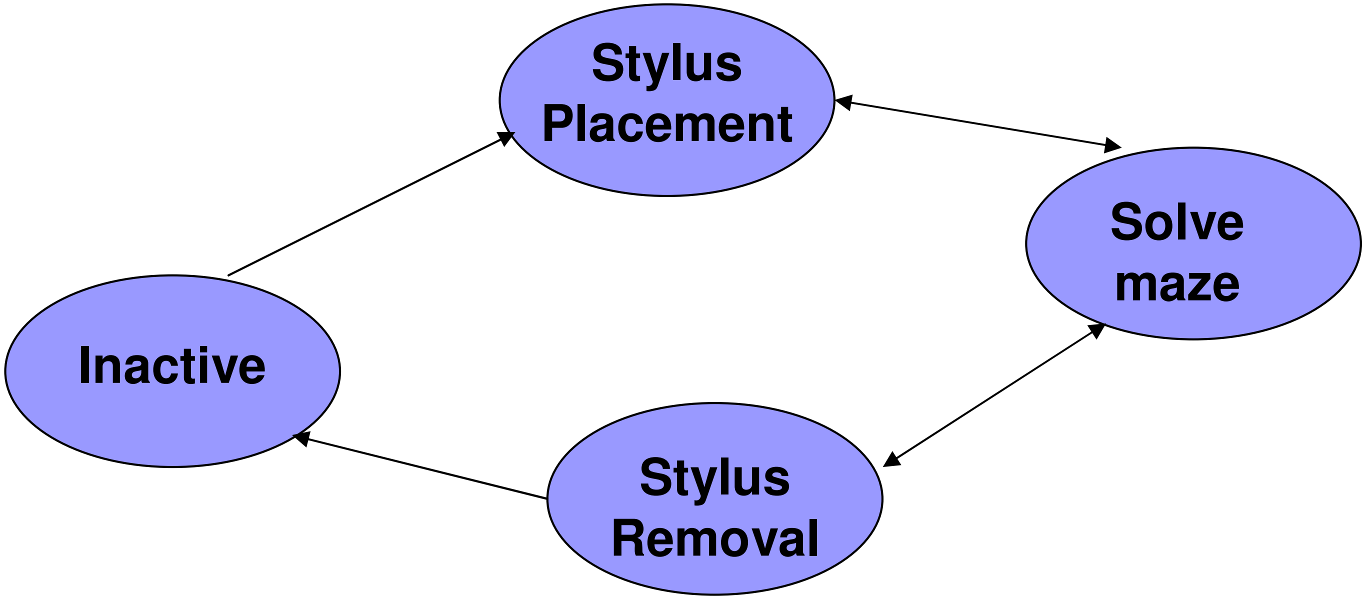
$$P(O \mid \lambda)=\Sigma_{i=1\ldots N}\,\alpha_i(T)$$

- Usually presented as the log likelihood:

$$\log(P(O \mid \lambda))$$

- Good Match is a negative value close to zero

System design and HMM training

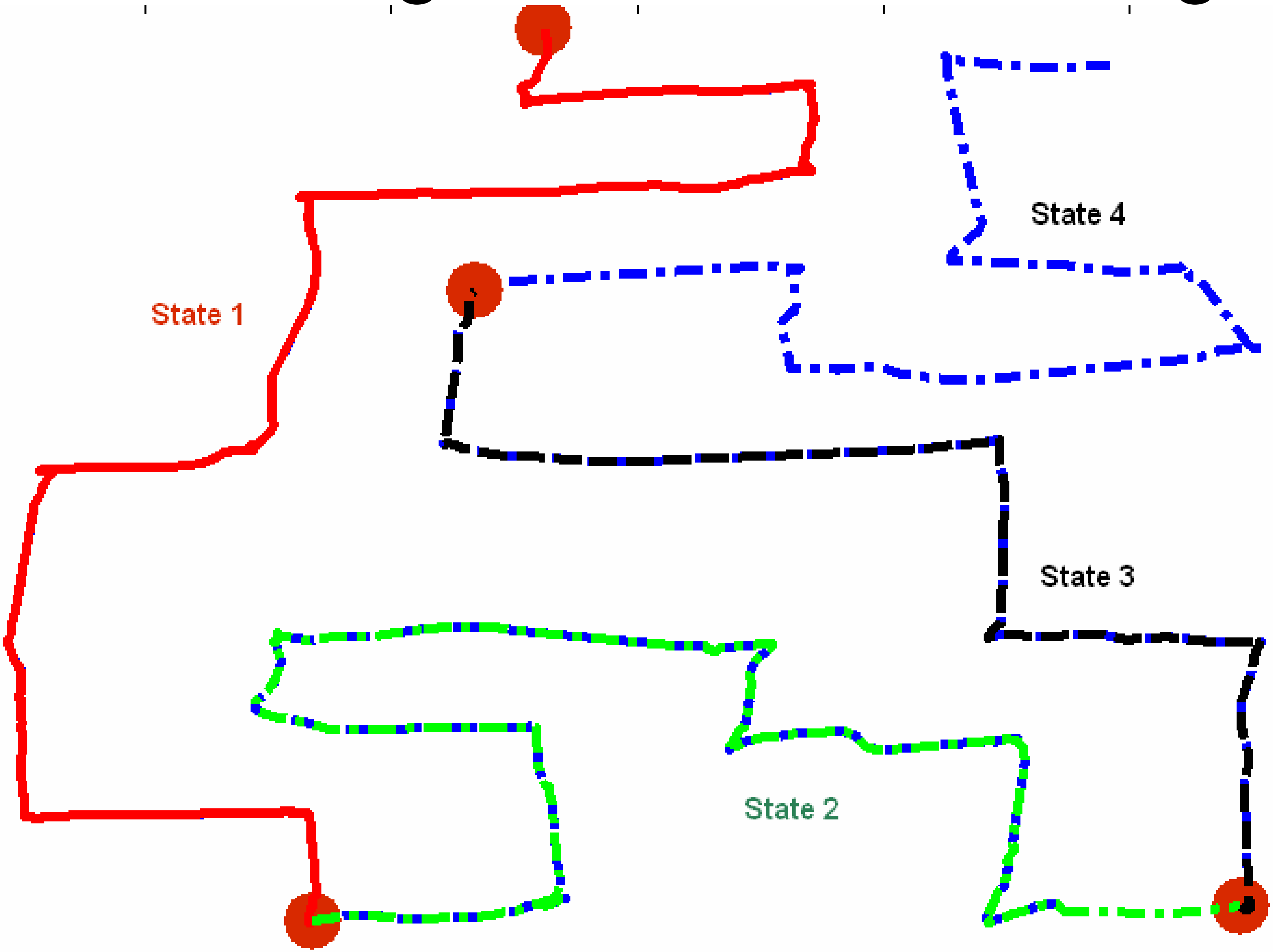


Approach based on work by Hundtofte et al (2002)
in task segmentation for remote surgical procedure

System design and HMM training

- Unable to carry out task level HMM
  - Lack of well defined protocol for users regarding the other states
  - Data set with task level segmentation did not have all possible output parameter (only pressure)
  - Potentially difficult to identify users all other task level vary greatly
- Developed HMM only within Maze Solve state

System design and HMM training



System design and HMM training

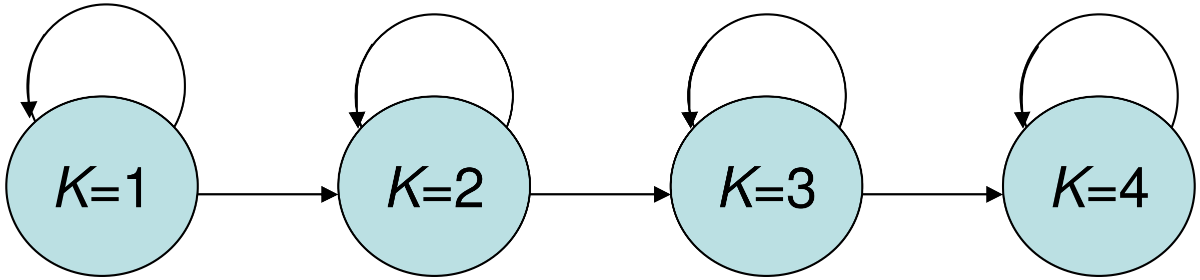
- State Topology of Maze solve
  - left-to-right transition with no state skips

Transition Probabilities (A)

P(S'|S)	S0	S1	S2	S3
S0	0.1	0.9	0	0
S1	0	0.1	0.9	0
S2	0	0	0.1	0.9
S3	0	0	0	1

Initial Probabilities(π)

S0	S1	S2	S3
1	0	0	0

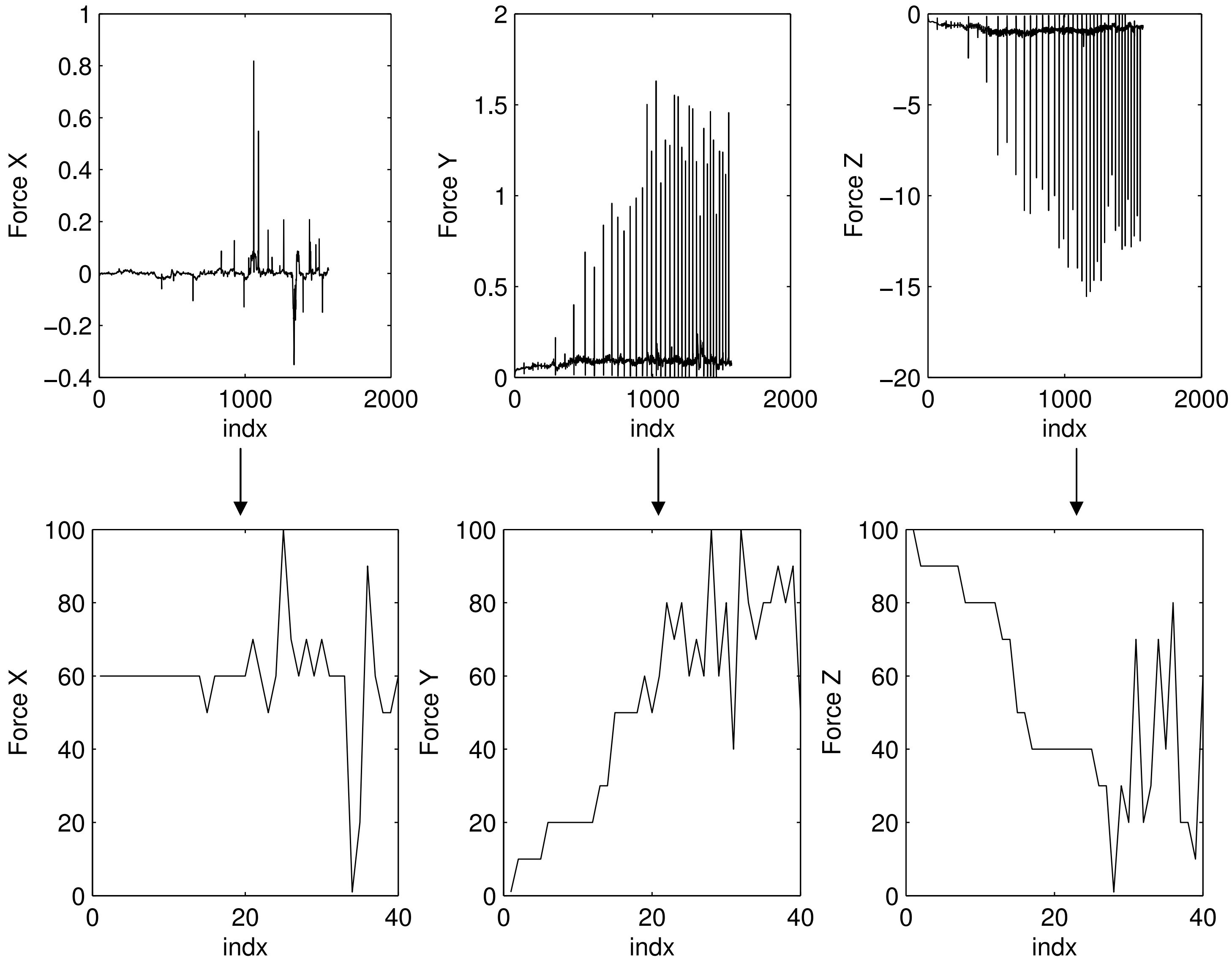


System design and HMM training

- Output Parameters:
  - Force(x,y,z) and Torque(x,y,z)
  - Segmented
    - p segments per state (N=p*4): $\Phi(k)$, k=1,2,…,N

      $\Phi(k)$= Sum (output parameter) in segment k/length of segment k
  - Normalized and quantized:
    - $\Phi^*(k)=Q[\Phi(k)]$,   k = 1,2, …, N

System design and HMM training

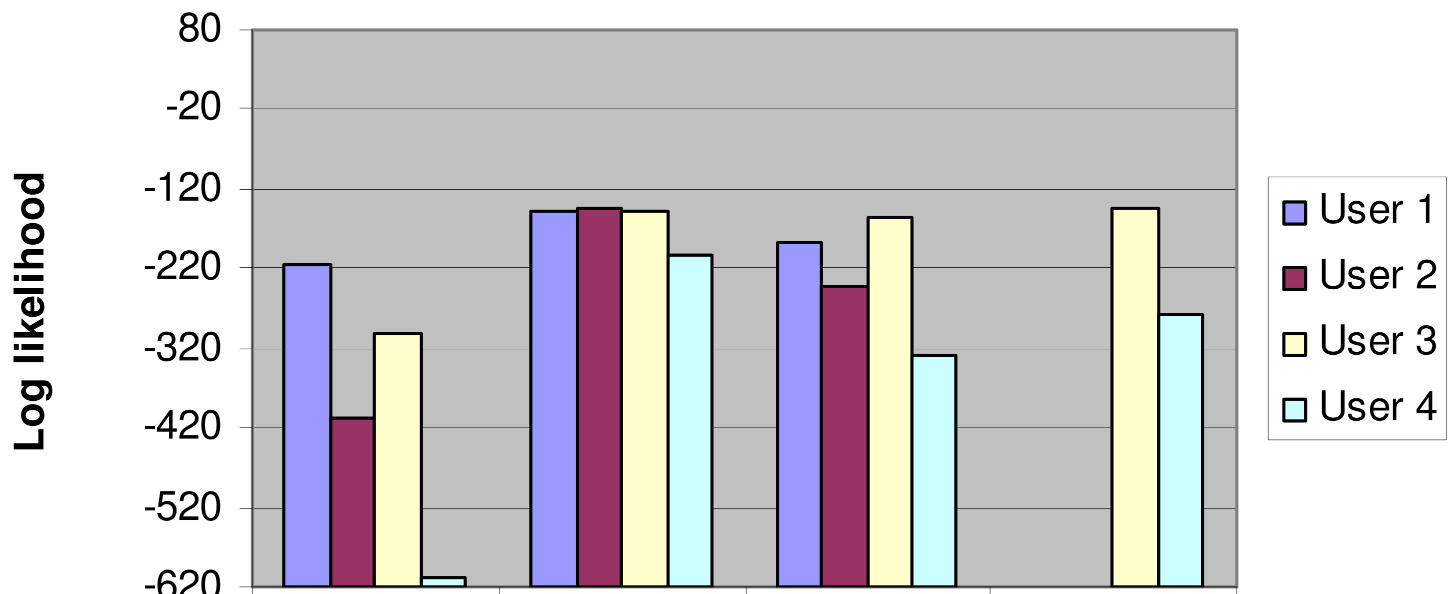


System design and HMM training

- The model for each user determined based on Baum-Welch algorithm:
  - $\lambda = (A, rand(B), \pi)$
  - 6 output parameters of 6 training data sets
    - 36 output sequence of length N (p*4)
    - 11 different symbols (O=1,10,20..100)
- The model can be tested on single and multiple parameters

Participant Identification

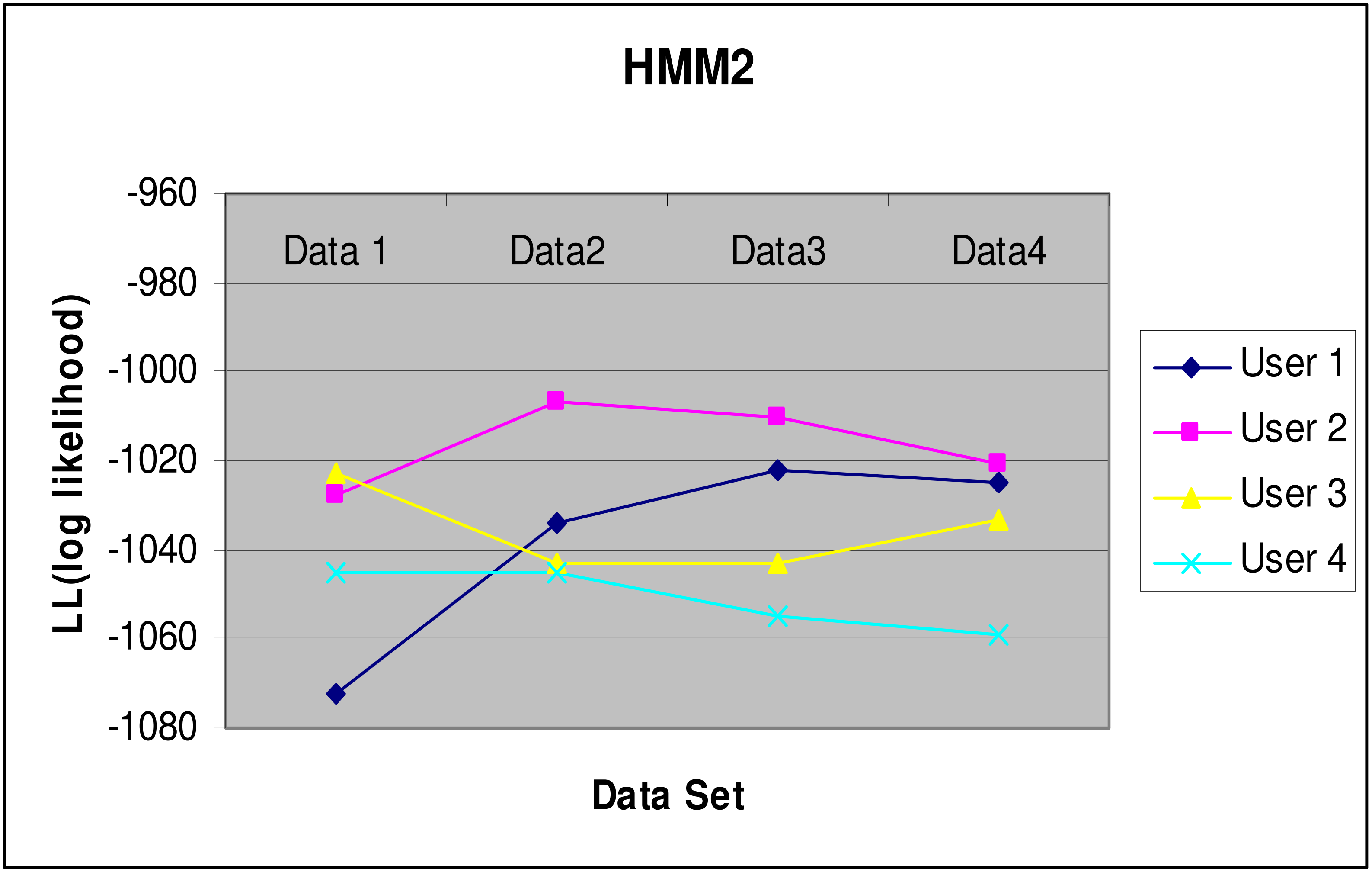
- Single parameter HMM (Torque Y) vs sum(LL)/user



	HMM1	HMM2	HMM3	HMM4
User 1	-213	-150	-186	-792
User 2	-406	-143	-243	-781
User 3	-300	-148	-157	-145
User 4	-607	-205	-329	-278

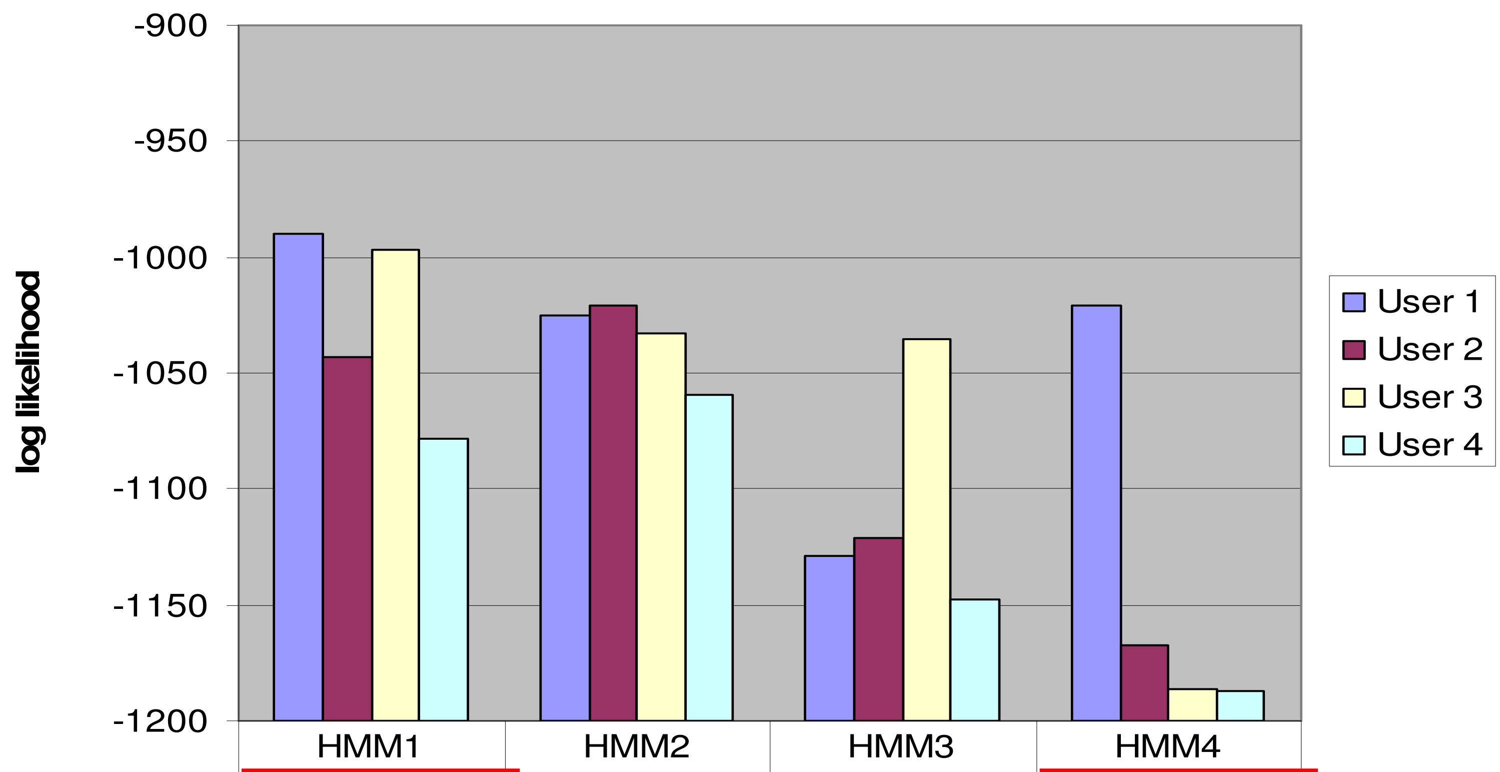
Participant Identification

- Multiple parameter HMM (All 6 output parameter)



Participant Identification

- Multiple parameter HMM (All 6 output parameter)
  - Sum of all log likelihood values for all 4 test sets



	HMM1	HMM2	HMM3	HMM4
User 1	-990	-1025	-1129	-1021
User 2	-1043	-1021	-1121	-1167
User 3	-997	-1033	-1035	-1186
User 4	-1078	-1059	-1148	-1187

Discussion and Conclusion

- Based on the observation:
  - A good HMM depends on the selection of output parameters
  - Not all parameters should be used for modeling
  - Include output parameter such as velocity, stylus angle may improve the model

Discussion and Conclusion

- Parameter selection could be based on:
  - Top performers of single parameter HMM
- Varying the number of states, the segment number per state and quantization level should be looked at
  - Wasn’t able to do this in Matlab due to memory issues

Discussion and Conclusion

- For continuous identification:
  - Average log likelihood value of several user data (between t_1 and t_2, $t_2>t_1$) would lead better detection
  - However, more susceptible to attacks with impostor adjusting the maze navigation approach with access to match score-log likelihood