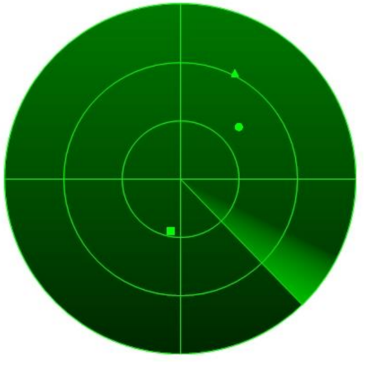
# Learning from Demonstrations

Jur van den Berg

# Kalman Filtering and Smoothing

- Dynamics and Observation model

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + \mathbf{w}_t, \quad \mathbf{w}_t \sim N(\mathbf{0}, Q)$$
$$\mathbf{y}_t = C\mathbf{x}_t + \mathbf{v}_t, \quad \mathbf{v}_t \sim N(\mathbf{0}, R)$$



- Kalman Filter:
  - Compute $\left(X_t \mid Y_0 = \mathbf{y}_0, \ldots, Y_t = \mathbf{y}_t\right)$
  - Real-time, given data so far
- Kalman Smoother:
  - Compute $\left(X_t \mid Y_0 = \mathbf{y}_0, \ldots, Y_T = \mathbf{y}_T\right), \quad 0 \le t \le T$
  - Post-processing, given all data

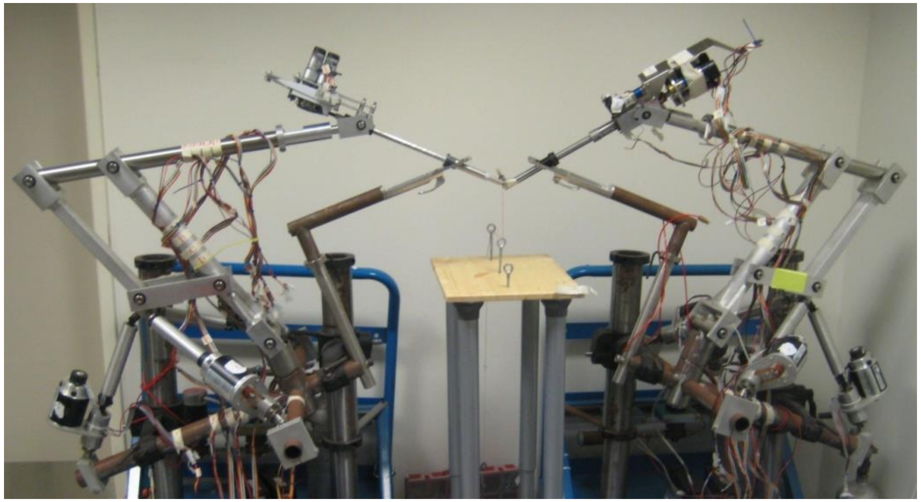
# EM Algorithm

$$\begin{aligned} \mathbf{x}_{t+1} &= A\mathbf{x}_t + \mathbf{w}_t, \quad \mathbf{w}_t \sim N(\mathbf{0}, Q) \\ \mathbf{y}_t &= C\mathbf{x}_t + \mathbf{v}_t, \quad \mathbf{v}_t \sim N(\mathbf{0}, R) \end{aligned}$$

- Kalman smoother:
  - Compute distributions $X_0, \ldots, X_t$ given parameters $A$, $C$, $Q$, $R$, and data $\mathbf{y}_0, \ldots, \mathbf{y}_t$.
- EM Algorithm:
  - Simultaneously optimize $X_0, \ldots, X_t$ and $A$, $C$, $Q$, $R$ given data $\mathbf{y}_0, \ldots, \mathbf{y}_t$.

# Learning from Demonstrations

- Application of EM-algorithm
- Example:
  - Autonomous helicopter aerobatics
  - Autonomous surgical tasks (knot-tying)



# Motivation

- Learning an ideal "trajectory" of system
- Human provides demonstrations of ideal trajectory
- Human demonstrations imperfect
- Multiple demonstrations implicitly encode ideal trajectory
- **Task:** infer ideal trajectory from demonstrations

# Acquiring Demonstrations

- Known system dynamics (*A*, *B*, *Q*)
- Observations with known sensors (*C*, *R*)
  - Inertial measurement unit
  - GPS
  - Cameras

$$\begin{aligned} \mathbf{x}_{t+1} &= A\mathbf{x}_t + B\mathbf{u}_t + \mathbf{w}_t, \quad & \mathbf{w}_t \sim N(\mathbf{0}, Q) \\ \mathbf{y}_t &= C\mathbf{x}_t + \mathbf{v}_t, \quad & \mathbf{v}_t \sim N(\mathbf{0}, R) \end{aligned}$$

- Use Kalman smoother to optimally estimate states **x** along demonstration trajectory

# Multiple Demonstrations

- *D* demonstration trajectories of duration $T^j$

$$\mathbf{d}_t^j = \begin{bmatrix} \mathbf{x}_t^j \\ \mathbf{u}_t^j \end{bmatrix} \quad j = 1, \ldots, D \quad t = 1, \ldots, T^j$$

- Hidden ideal trajectory **z** of duration $T^*$

$$\mathbf{z}_t = \begin{bmatrix} \mathbf{x}_t^* \\ \mathbf{u}_t^* \end{bmatrix} \quad t = 1, \ldots, T^*$$

# Model of Ideal Trajectory

- Main idea: *use demonstrations as noisy observations of hidden ideal trajectory*
- Dynamics of hidden trajectory

$$\mathbf{z}_{t+1} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \cdot \mathbf{z}_t + \mathbf{w}_t^*, \quad \mathbf{w}_t^* \sim N(\mathbf{0}, \begin{bmatrix} Q & 0 \\ 0 & N \end{bmatrix})$$

- Observation of hidden trajectory

$$\begin{bmatrix} \mathbf{d}_t^1 \\ \vdots \\ \mathbf{d}_t^D \end{bmatrix} = \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix} \cdot \mathbf{z}_t + s_t, \quad s_t \sim N(\mathbf{0}, \begin{bmatrix} S^1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & S^D \end{bmatrix})$$

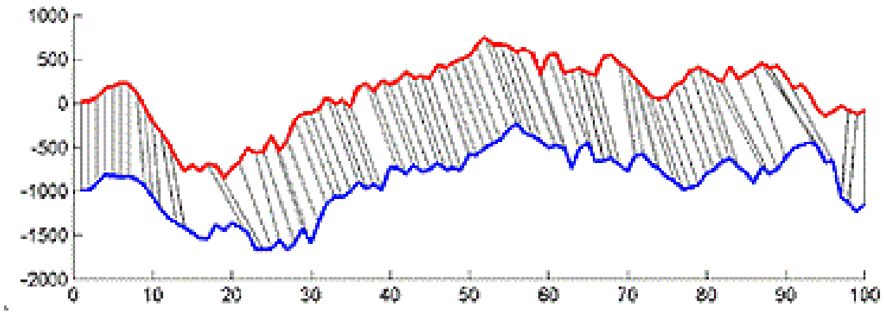
# Inferring Ideal Trajectory

- Dynamics model: Parameter $N$ controls smoothness; $A$, $B$, $Q$ known
- Observation model: Parameters $S$ encode relative quality of demonstrations

- Use EM-algorithm with Kalman smoother to simultaneously optimize $\mathbf{z}$ and $S$ (and $N$).
  - Initialize S with identity matrices

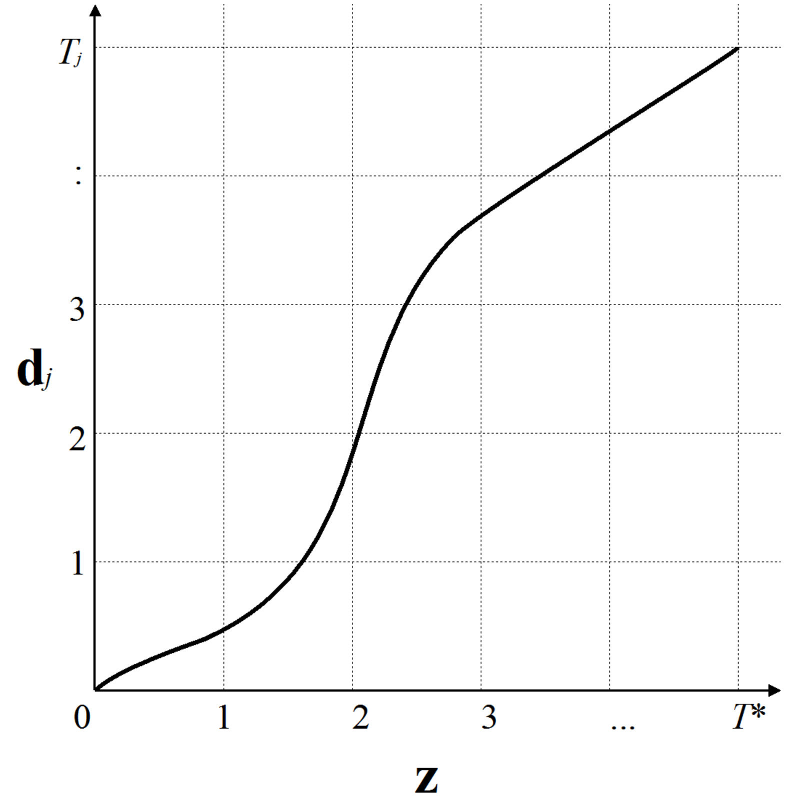
# Time Warping

- But, this assumes demonstrations are of equal length and uniformly paced
- Include *Dynamic Time Warping* into EM-algorithm




- Such that demonstrations map temporally

# Time Warping




- For each demonstration *j*, we have function $\tau^j(t)$
- Maps time *t* along **z** to time $\tau^j(t)$ along $\mathbf{d}^j$
- Adapted observation model:

$$\begin{bmatrix} \mathbf{d}^1_{\tau^1(t)} \\ \vdots \\ \mathbf{d}^D_{\tau^D(t)} \end{bmatrix} = \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix} \cdot \mathbf{z}_t + \mathbf{s}_t, \quad \mathbf{s}_t \sim N(\mathbf{0}, \begin{bmatrix} S^1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & S^D \end{bmatrix})$$

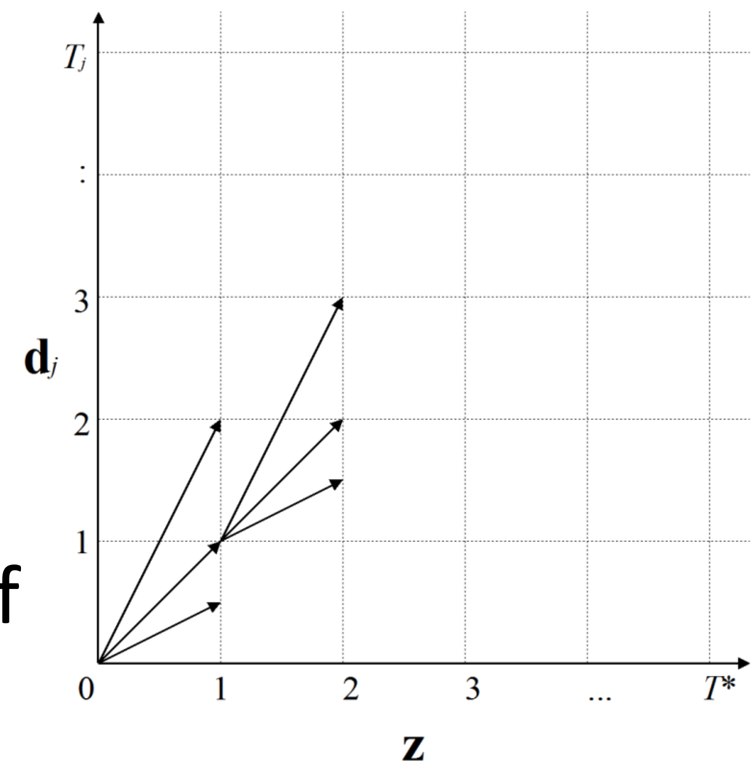
# Learning Time Warping

- $\tau^j(t)$ is (initially) unknown
- Assume (initially):
  - $T^* = (T^1 + \ldots + T^D) / D$
  - $\tau^j(t) = (T^j / T^*)\, t$
- Adapted EM-algorithm:
  - Run Kalman smoother with current S and $\tau$
  - Optimize S by maximizing likelihood
  - Optimize $\tau$ by maximizing likelihood (Dynamic Time Warping)

# Dynamic Time Warping

- Match demonstration *j* with **z**
- Assume that demonstration moves locally
  - twice as slow as **z**
  - same pace as **z**
  - twice as fast as **z**
- Dynamic Programming to find optimal "path"
- Cost function: likelihood of

$$\mathbf{d}^{j}_{\tau^{j}(t)} = \mathbf{z}_t + \mathbf{s}_t, \quad \mathbf{s}_t \sim N(\mathbf{0}, S^{j})$$

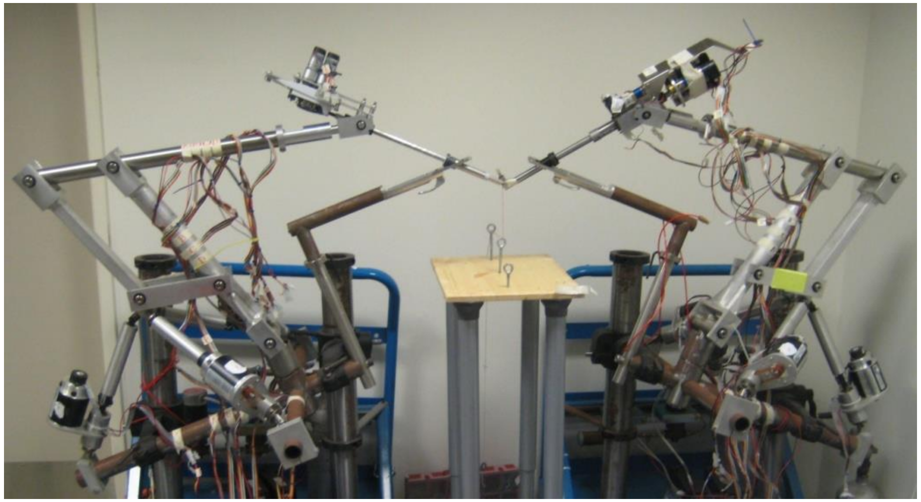


# Example: Helicopter Airshow

- Thesis work of Pieter Abbeel
- Unaligned demonstrations:
  - Movie
- Time-aligned demonstrations:
  - Movie
- Execution of learnt trajectory
  - Movie

# Example Surgical Knot-tie

- ICRA 2010 Best Medical Robotics Paper Award
- Video of knot-tie



# Conclusion

- Learning from demonstrations
- Includes Dynamic Time Warping into EM-algorithm