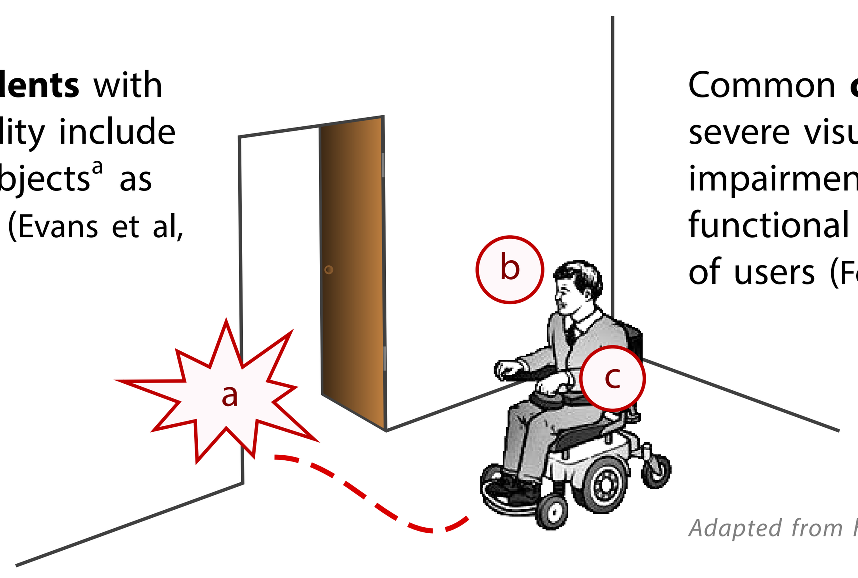
Markov Framework for Power Wheelchair Driving

Tim Yang[1–3] Seth Hutchinson[4–6] Yih-Kuen Jan[1–3]

[1] Rehabilitation Engineering Lab, [2] Kinesiology & Community Health, [3] Computational Science & Engineering, [4] Coordinated Science Lab, [5] Beckman Institute, [6] Electrical & Computer Engineering

Purpose

Common **accidents** with powered mobility include running into objects[a] as well as people (Evans et al, 2007)



Adapted from http://www.ada.gov/

Common **causes** include severe visual[b] and motor[c] impairments, preventing functional steering in 40% of users (Fehr et al, 2000)

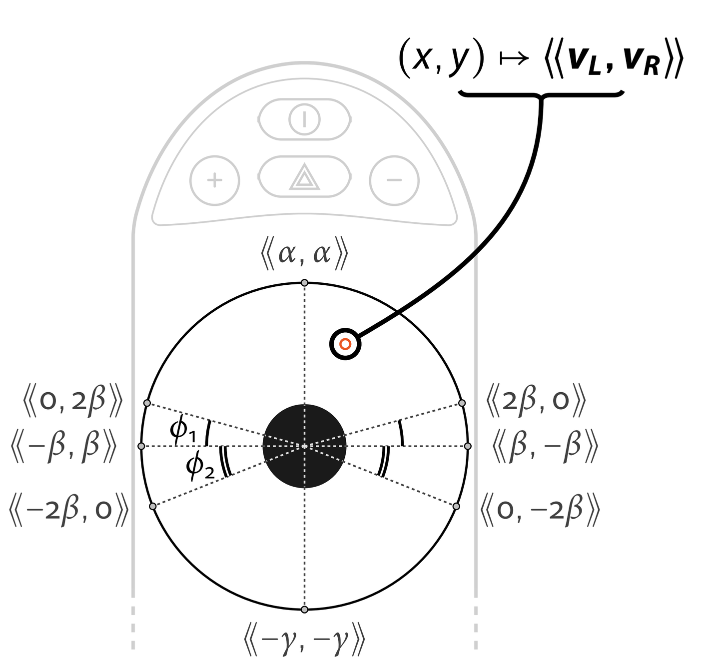
- Powered mobility assessments are needed to ensure the safety of potential users and their surroundings
- Existing assessments are either subjective or descriptive
- To better characterize the dynamic, stochastic, and nonlinear nature of human driving (Nechyba & Xu, 1997), our goal was **to develop a quantitative framework for powered mobility**

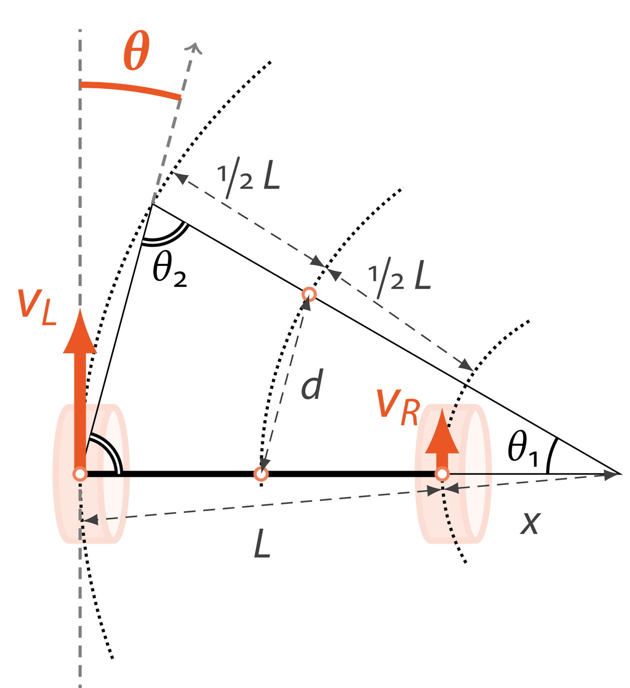
Approach

Driving	Markov Model	Methodology
Position	States $S = \{s_0, s_1, \ldots, s_n\}$	Location nodes represented as a **3-branch, 5-depth tree**
Direction	Actions $A(s) = \{a_0, a_1, \ldots, a_k\}$	Heading vectors based on wheel velocity and holonomic **differential steering**
Uncertainty	Transitions $P(s'\|s, a)$	Human motor uncertainty simulated by a reference **model with 20% variability**
Environment	Rewards $R(s)$	Infrared depth imaging captured by a retrofitted **Xbox 360 Kinect sensor**

- We constructed a tree-based **Markov decision process** (MDP)
- We assessed 2 able-bodied volunteers driving a Permobil C400 power wheelchair (custom retrofitted with a Kinect and Arduino)

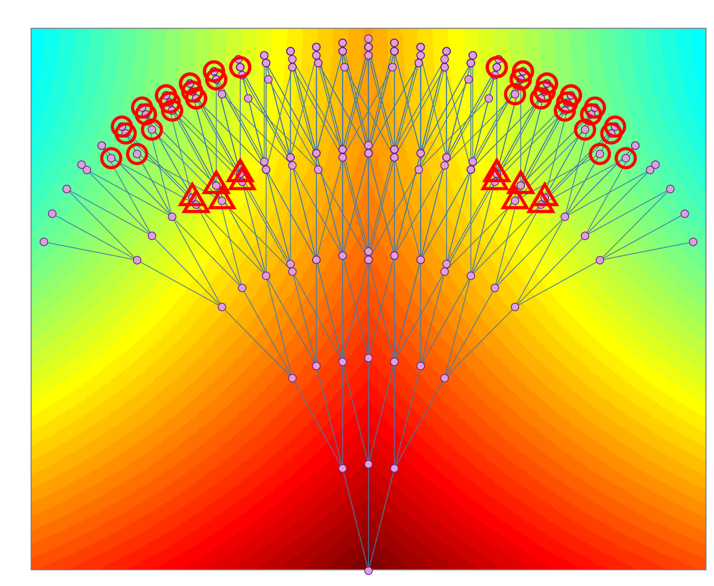
Heading Vectors ↦ Actions $A(s)$





- **Joystick excursions** were intercepted with an Arduino
- Each excursion coordinate (x, y) was mapped into a **wheel velocity** tuple $\langle\langle v_L, v_R \rangle\rangle$
- Movement was calculated based on differential steering
- Each **heading vector** angle θ was computed from $\langle\langle v_L, v_R \rangle\rangle$ via rigid body mechanics

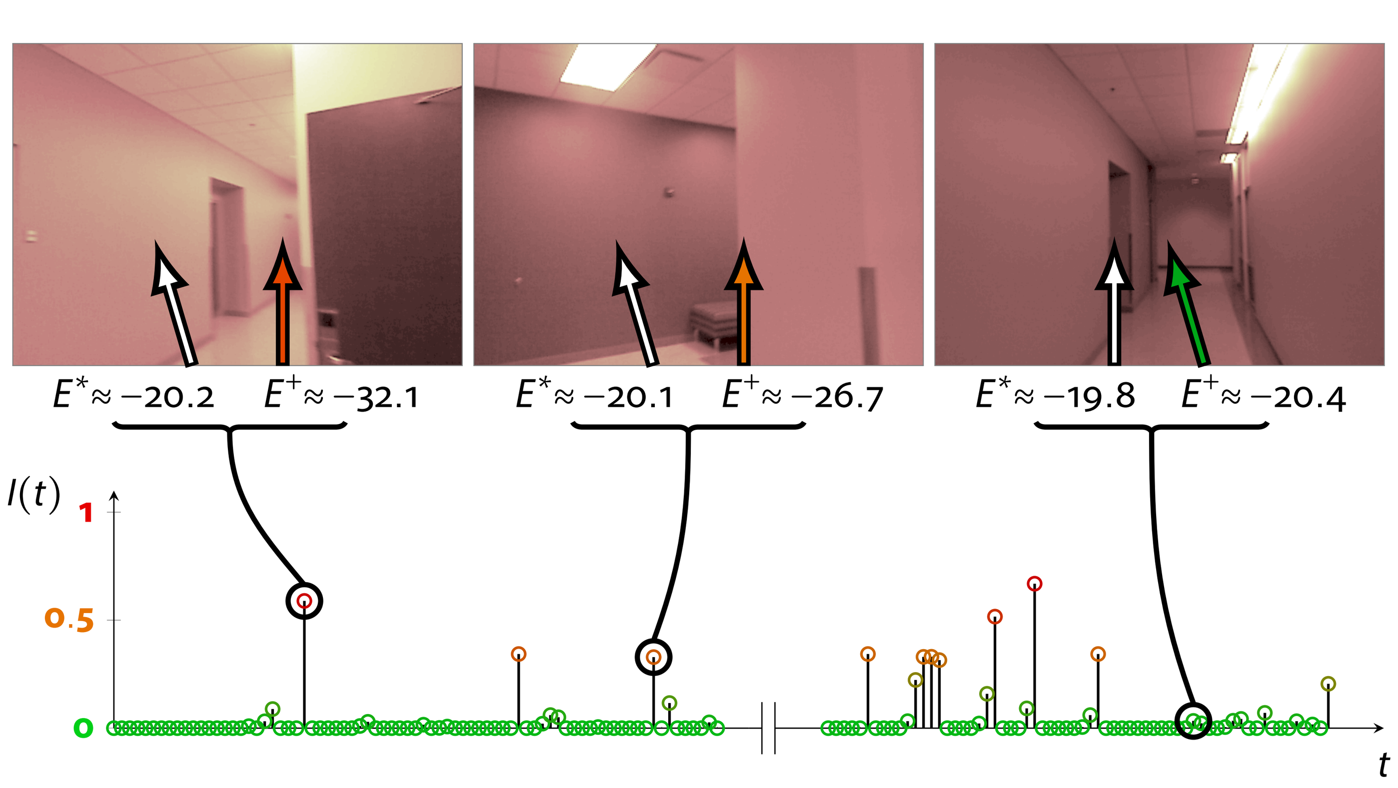
Environment ↦ Rewards $R(s)$



$$R(s) = \lambda\sqrt{(x_{max} - |s_x|)^2 + (y_{max} - |s_y|)^2}$$

- Each depth frame was transformed into a point cloud
- **Obstacles were mapped** by transposing the state tree onto the point cloud
- Obstacles were penalized with negative rewards ($\lambda < 0$)
- $R(s)$ was scaled for **proximity** and **directness of path** to the root node (0, 0)

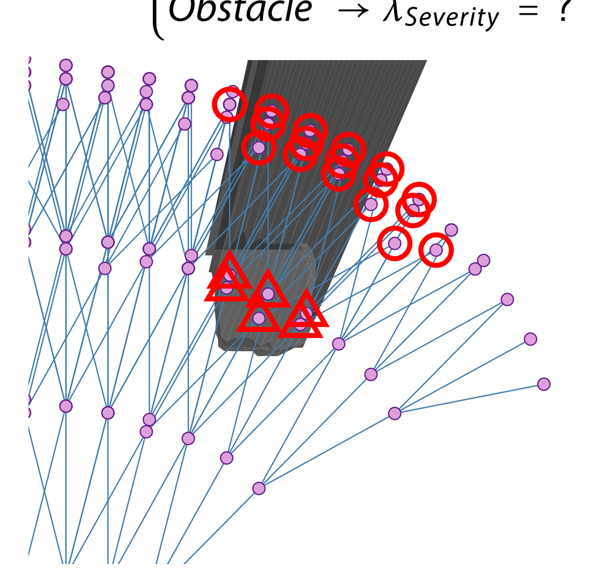
Expected Utilities $E(t)$ ↦ Assessment

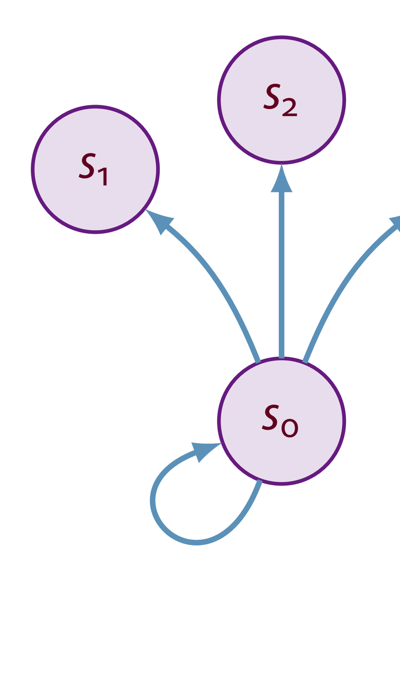


- Suboptimal decisions ($E^+ \neq E^*$) were quantified as a **risk index $I(t)$**
- $I(t) = |E^+(t) - E^*(t)| \times |E^*(t)|^{-1}$, where $E^+(t)$ and $E^*(t)$ were the observed and optimal expected utilities

Future Work

$$Blob \overset{?}{=} \begin{cases} Objective \to \lambda_{Desirability} = ? \\ Obstacle \to \lambda_{Severity} = ? \end{cases}$$





$$\begin{aligned} P(s_0|s_0, a_0) &= ? \\ P(s_0|s_0, a_1) &= ? \\ P(s_0|s_0, a_2) &= ? \\ P(s_1|s_0, a_0) &= ? \\ P(s_1|s_0, a_1) &= ? \\ &\ldots \\ P(s_3|s_0, a_2) &= ? \end{aligned}$$

- **Improve context awareness of $R(s)$**
- Add blob classification to account for both obstacles and objectives
- **Experimentally model $P(s'|s_0, a)$**
- Use individualized transition functions for clinical training and real-time assistance

Computational Science and Engineering 2014 Annual Meeting