Interactive Autonomous Driving through Adaptation from Participation (AfP)

Anqi Xu, Qiwen Zhang, David Meger and Gregory Dudek 2014

Presented by Nnorom Ekele
February 22, 2019
CSC2621

Supervised Approaches

Problems with supervised approaches:

- Covariance Shift
- Linear policies work well still errors.
- Error in hold-out data and training error were similar.
- We need a system that updates on the fly/ online.
- We need a system adaptive to a dynamic environment.

Introduction

What’s a better way a robot autonomy can work.

Working alongside a human.

Motivation

- Autonomous driving: A common goal shared by many is for an autonomous driving solution that can operate robustly within diverse outdoor environments.
- Existing systems have failed due to diversity of environment conditions.
- Global Positioning Systems (GPS) are inappropriate for operations in previously unseen or changing paths, suffers multipath noise interference especially near buildings.

Contributions

- Demonstration of an interactive autonomous driving system using AfP paradigm.
- Adaptation from Participation achieves the task of adjusting parameter values automatically in a robust and flexible manner.
- This paper claims shared autonomy produces better results than teleoperated and fully autonomous systems.
- Developed the Adaptive Parameter Exploration (APEX) algorithm in order to implement AfP for shared autonomy.
- This adaptive system is able to adjust to dynamic changes to task conditions.
- Applied APEX to AfP case studies on previously unseen paths.

Background

Prior Work

- AfP is an extension of Learning from Demonstration (LfD).
- LfD/ Imitation Learning: learning behaviors from demonstrations provided by a human or another robot expert with superior task knowledge.
- A key motivation for LfD is to eliminate the tedious task of manually programming behaviors for robots similar to AfP.

Prior Work

- AfP also builds on previous research in the domain of shared autonomy.
- Shared autonomy shown to give better results than tele-operated and fully autonomous vehicles.
- AfP aims to improve efficiency of a task specific robot by repeatedly adjusting its parameters.

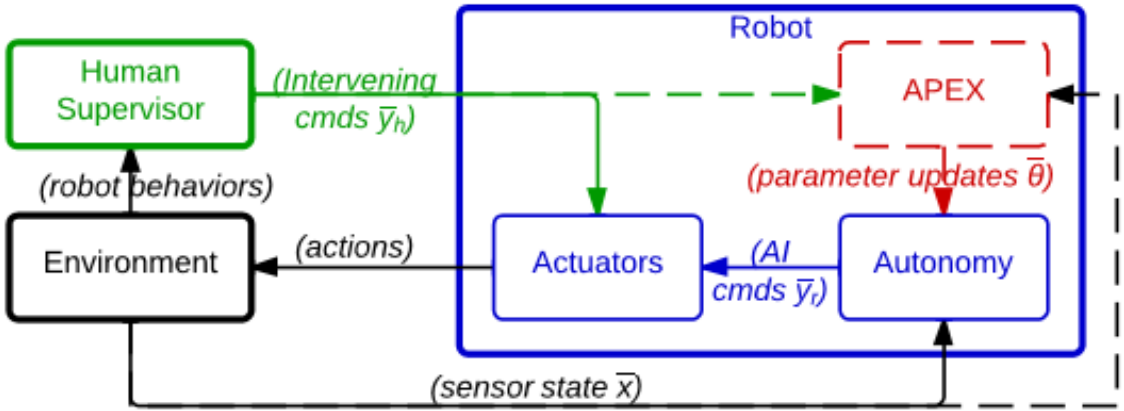
AfP vs previous LfD approaches

- AfP differs from previous LfD approaches as AfP is designed for highly dynamic situations previous LfD approaches have a single, stationary task objective.
- More so, in previous LfD approaches, the robot agent learns a novel set of task behaviors while in AfD robot agent improves the performance of an existing robot autonomy.

Adaptive Parameter Exploration

Adaptive Parameter Exploration (APEX)

- **APEX:** APEX is an algorithmic solution to the novel problem of Adaptation from Participation.



Parameter Hypothesis - Particles

- Particle - an evolving parameter hypothesis.
- APEX uses a multi-hypothesis search approach.
- Parameter optimization is between two consecutive sensor updates based on training exemplars.
- Particles involved; local search particles, random search particles and persistence particles.

Particle Types

- Local search particles – employ gradient-based search method to iteratively find numerical approximations.
- Random restart search particles – provide non-local explorations of parameter space by randomly sampling initial parameter values.
- Persistence particles – enforce temporal consistency by duplicating winning particle from previous loop iteration, and do not perform further parameter optimizations.

APEX Algorithm

Algorithm 1 APEX's main pipeline loop

$\mathscr{C}_i \leftarrow 0 \ \forall i$
loop
  wait for incoming sensor update $\bar{x}$
  if *particles are optimizing parameters* $\overline{\theta_i}$ **then**
    for all *particles* i **do**
      pause optimization
      update long-term cost $\mathscr{C}_i$
    end for
    $i^* \leftarrow argmax_i(\mathscr{C}_i)$ *// choose winning particle*
    update main pipeline's parameters $\overline{\theta}$ *with* $\overline{\theta_{i^*}}$
  end if
  $\overline{s'} \leftarrow \overline{s}$ *// save prior autonomy state*
  $\overline{y_r}, \overline{s} \leftarrow \mathbb{A}(\bar{x}, \overline{\theta}, \overline{s'})$
  if $\overline{y_h} \neq \varnothing$ **then**
    store latest exemplars $\{\bar{x}, \overline{s'}, \overline{y_h}\}$
    resume optimization for all particles
  else
    $\mathscr{C}_i \leftarrow 0 \ \forall i$
  end if
end loop

Each particle optimizes between consecutive updates

Local search, random start search and persistence particles

Update using a discount γ hyparamter

This should be an argmin for lowest cost

During manual intervention

APEX Algorithm

- Updating long term cost

$$\mathscr{C}_i \leftarrow \gamma \mathscr{C}_i + cost\left(\overline{\theta_i}\right) \qquad \gamma \in [0,1]$$

long-term cost — Parameter hypothesis — discount factor

- Updating Mean Square objective Cost

$$cost\left(\overline{\theta_i}\right) = \frac{1}{W} \sum_{w=1}^{W} \left\| \overline{y_{h,w}} - \mathbb{A}\left(\overline{x_w}, \overline{\theta_i}, \overline{s'_w}\right) \right\|^2$$

Robot autonomy

- For continuous particles

$$\overline{\theta^c} \leftarrow \overline{\theta^c} + \alpha\left(\overline{\theta^c_{i^*}} - \overline{\theta^c}\right)$$

continuous particles

γ - discount factor enforces temporal consistency and reduces the likelihood of oscillating between different winning particiles in successive iterations

W – optimizing using a sequence of W most recent training exemplars

$\overline{y_{h,w}}$ - Human intervening commands

α – learning rate ϵ [0,1]

SL-Commander vehicle

System Infrastructure - SL-Commander vehicle

An electric-powered side-by-side All-Terrain Vehicle (ATV)

1. Mechanical Platform:
   - The SL-Commander utilizes Ackerman steering and a four wheel drive system.
   - Safety feature includes a steel roll-cage for protection from roll-overs on steep inclines, as well as front and rear ventilated disc brakes for rapid stopping power.

System Infrastructure - SL-Commander vehicle

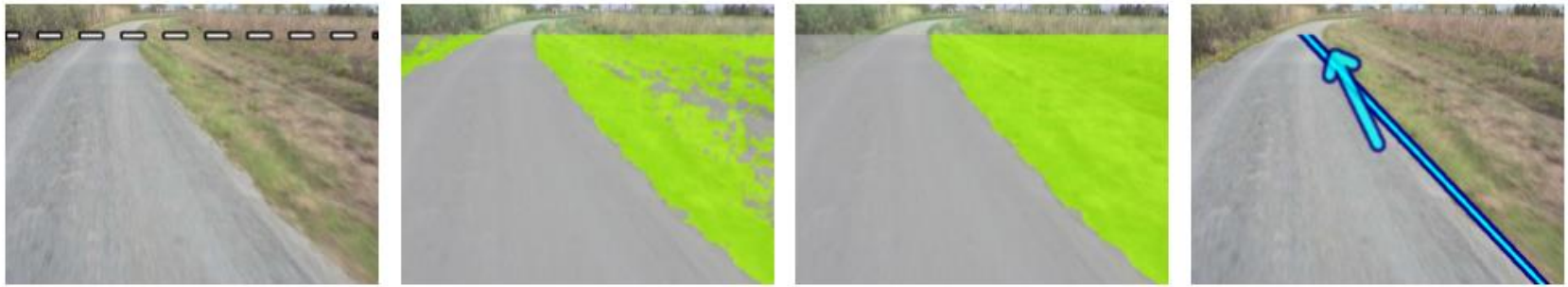
2. Actuation and Sensing:

A drive-by-wire system onboard the SL-Commander replicates many of the operations typically conducted by a human driver during manual control,including actuation of the brake and accelerator pedals, the gear shift, and also the steering wheel.

3. Software Architecture was the Robot Operating System middleware (ROS).

Vision Based Algorithm - SL-Commander vehicle

Vision Based Path Following algorithm



(a) exclude horizon (b) segment target region (c) extract boundary curve (d) fit line and map into steering

Vision Based Algorithm - SL-Commander vehicle

Vision Based Path Following algorithm

- first the top of the image is excluded to remove the horizon and image content near the vanishing point.
- next the target region is segmented using a specified image feature (e.g. HSV).
- the dominant boundary curve is then extracted from the filtered segmented image
- the resulting line fit of the boundary is mapped into a steering command (blue arrow)

Vision Based Algorithm

- Geometric information – camera pose, intrinsic and extrinsic parameters, inclination and desired boundary are tedious to obtain for the dynamic environment.
- These geometric information is required to obtain steering angle.
- Adaptive parameterized linear mapping is used instead:

$$y_r = M_1X + M_2Ø + M_3$$

y_r – robot autonomy command to actuator,

X – intersection on X axis,

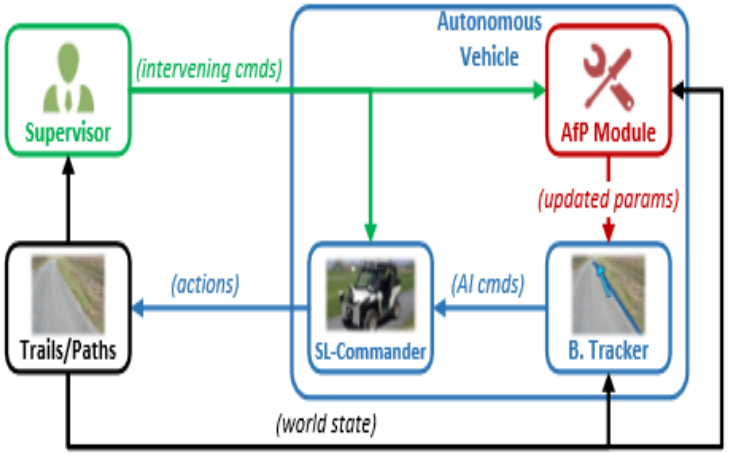
Ø – slope of line

M_1, M_2 – scaling factors (to be optimized), M_3 – bias (to be optimized as well)

Parameters of the system

As the parameters to optimize are:

- Boundary type $T_b \in$ {Edge, Strip} and
- Segmentation feature choice $T_s \in$ {Hue, Grayscale, HSV}.
- Horizon cut-off threshold H_0 which is a continuous value,
- Mapping coefficients M_1, M_2, and M_3.



Field Experiments

Test site route

Evaluation Scenario

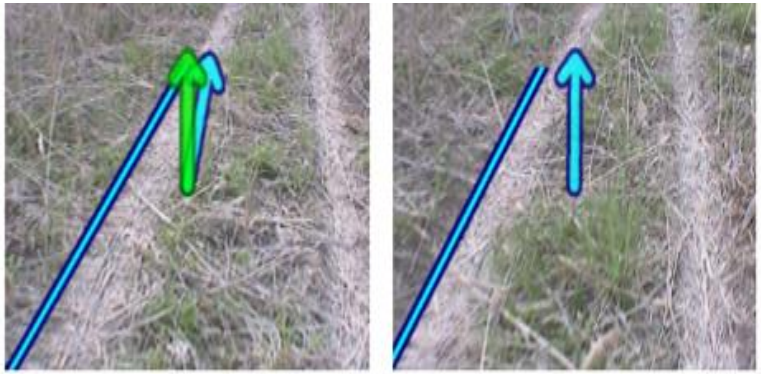
(a) (b) (c) (d)

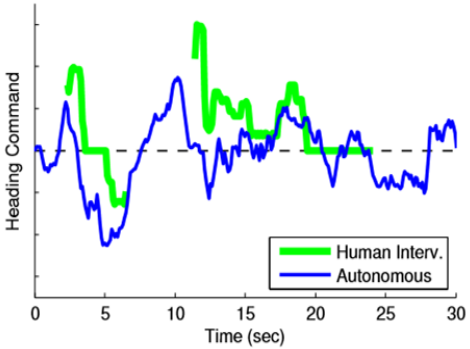
Fig. 6. Snapshots of a run through the primary test circuit: (a) the human passenger began by training the boundary tracker at 3 km/h speed to follow a previously-unseen gravel road, with the user's command shown as a green arrow; (b) shortly after, the passenger relinquished control to the autonomous system, which continued to track the path, with its autonomous steering command shown as a blue arrow; (c) the passenger incrementally ramped up the autonomous driving speed to 20 km/h after witnessing robust tracking of the gravel-grass border through diverse surroundings; and (d) the run concluded near a large tent that produced a confusing secondary boundary, although autonomous tracking of the gravel path remained unfaltering.

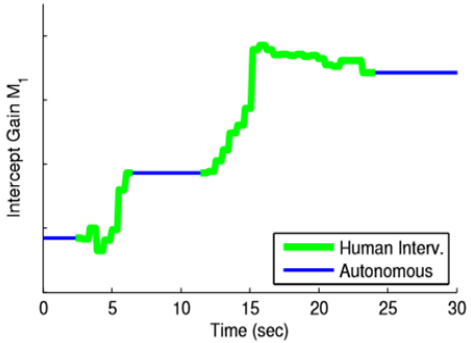
(a) (b) (c) (d)

Fig. 7. Sample segment demonstrating interactive re-learning following a change in the camera's positioning: (a) following initial manual training, the vehicle autonomously tracked the gravel pathway, with steering command shown as a blue arrow; (b) the passenger began issuing intervening commands, shown with the green arrow, in preparation to change the camera's pose while the vehicle is in motion; (c) after panning the camera to the right and downwards, this unexpected change in image perspective caused the robot's command to differ from the user's desired steering; however (d) after AfP swiftly adapted parameter settings to match the updated camera pose, the human quickly returned control back to autonomous driving system.

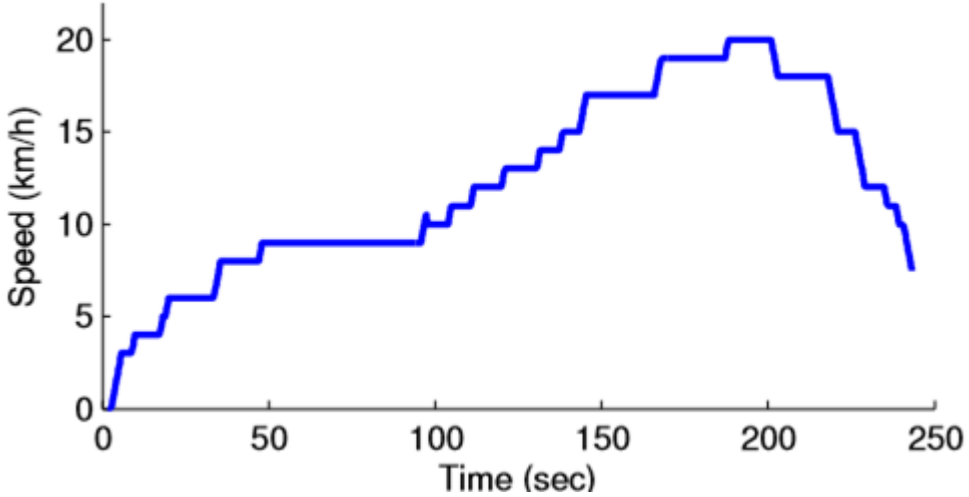
Evaluation Scenario







Progressive vehicle speed during adaptation



Major Benefits of Adaptation from Participation

- Extends Learning from Demonstration – imitation learning, adaptively improving performance.
- Dynamic adaptation to changing task objectives and conditions.
- AfP encourages continuous interactions between the robot autonomy and the human participant for better performance.
- Demonstrates ability to handle accidental perturbations to robot’s physical configuration.
- Human participants were always in full control over the vehicle.
- Largely agnostic to the underlying system – can be extended to other human robot teams.

Questions

Future Work

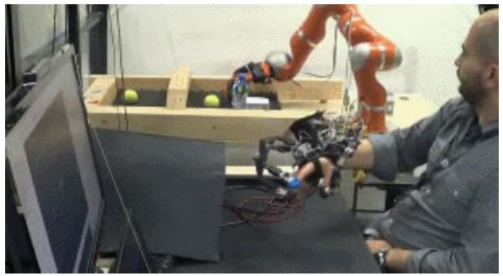
- Extend current work to explicitly modelling of user intention for different interaction periods.
- Extend current work to other shared autonomies.
- Deployment in more challenging outdoor domains like agriculture, mining and forestry.
- Capturing the user intentions for more desirable driving behavior.
- Improvement of user interface from laptop feedback to visualization by Augmented Display and replacing gamepad controller with a steering wheel for ongoing investigations.

Shared Autonomy via Hindsight Optimization

Presented by Bin Yang

22 Feb. 2019

Teleoperation



Noisy, insufficient degrees of freedom, tedious

Image credit: Javdani RSS2015 talk

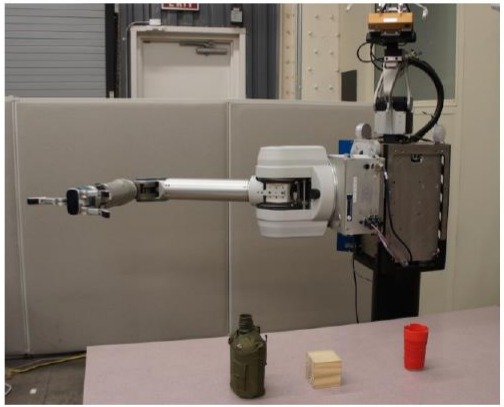
Shared Autonomy

User Input

+

Autonomous Assistance

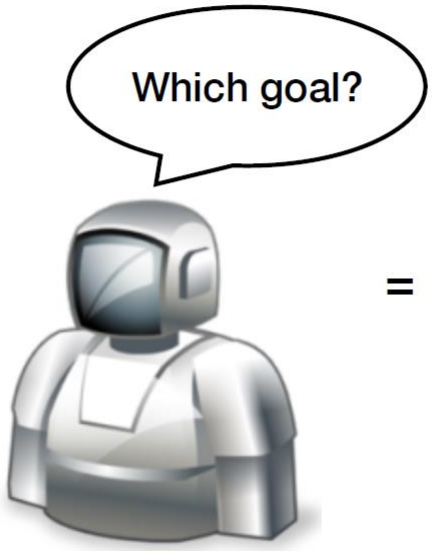
=



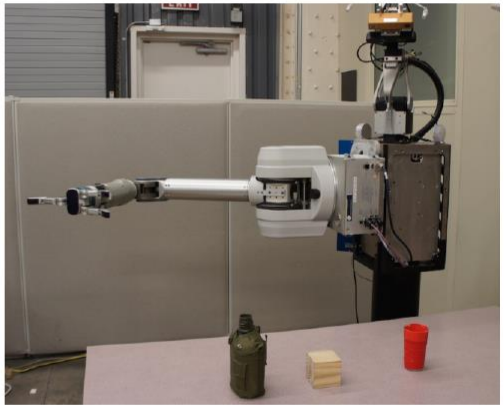
Achieve Goal

Shared Autonomy

+



=

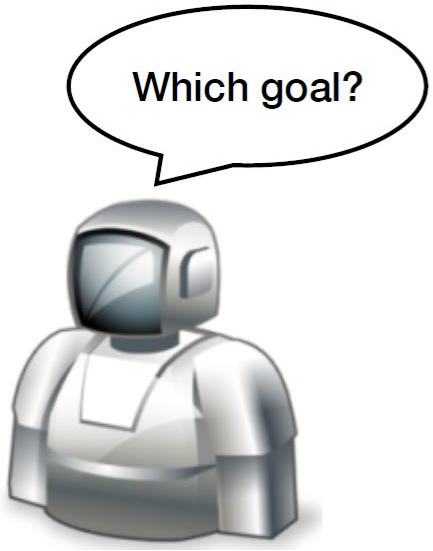


User Input **Autonomous Assistance** **Achieve Goal**

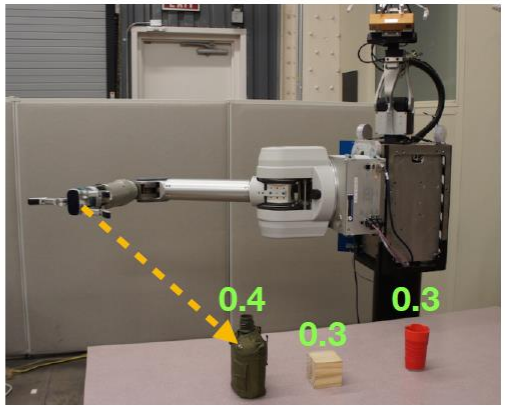
Shared Autonomy

Predict goal
Assist for single goal

[Dragan and Srinivasa 13]
[Kofman et al. 05]
[Kragic et al. 05]
[Yu et al. 05]
[McMullen et al. 14]
...



Autonomous Assistance



Achieve Goal

Shared Autonomy

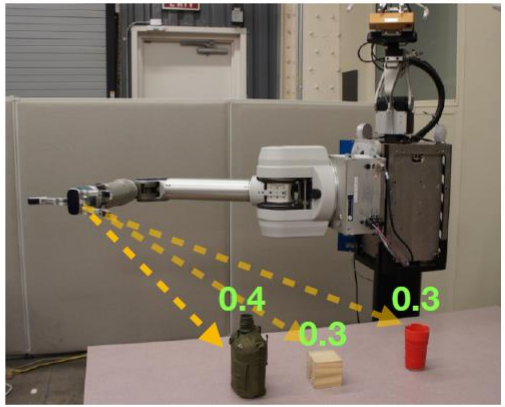
Predict goal
Assist for single goal

[Dragan and Srinivasa 13]
[Kofman et al. 05]
[Kragic et al. 05]
[Yu et al. 05]
[McMullen et al. 14]
...

Predict goal distribution
Assist for distribution

[Hauser 13]
This work!

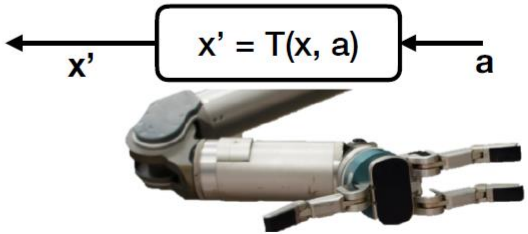
Autonomous Assistance



Achieve Goal

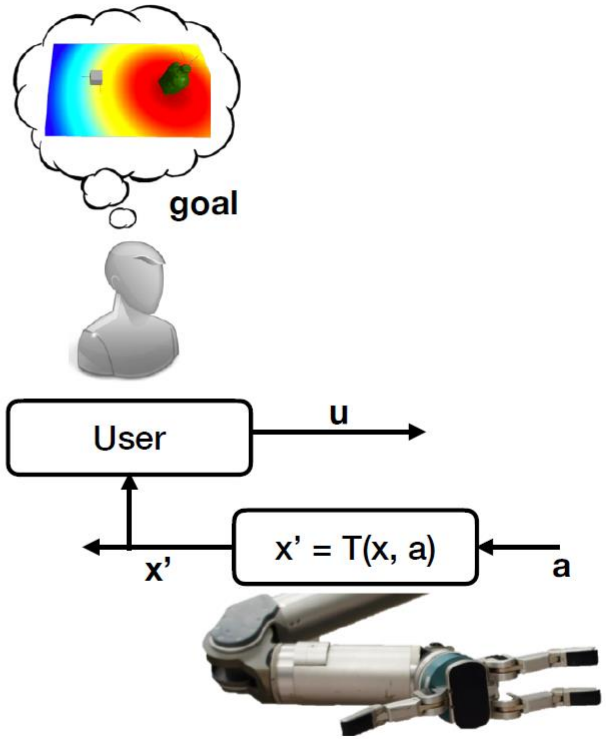
Method

- System dynamics: x’ = T(x, a)



Method

- System dynamics: x’ = T(x, a)
- User (MDP) as $(X, U, T, C_g^{\text{usr}})$
  - User policy: $\pi_g^{\text{usr}}(x) = p(u|x, g)$
  - MaxEnt IOC: $C_g^{\text{usr}} : X \times U \to \mathcal{R}$



Method

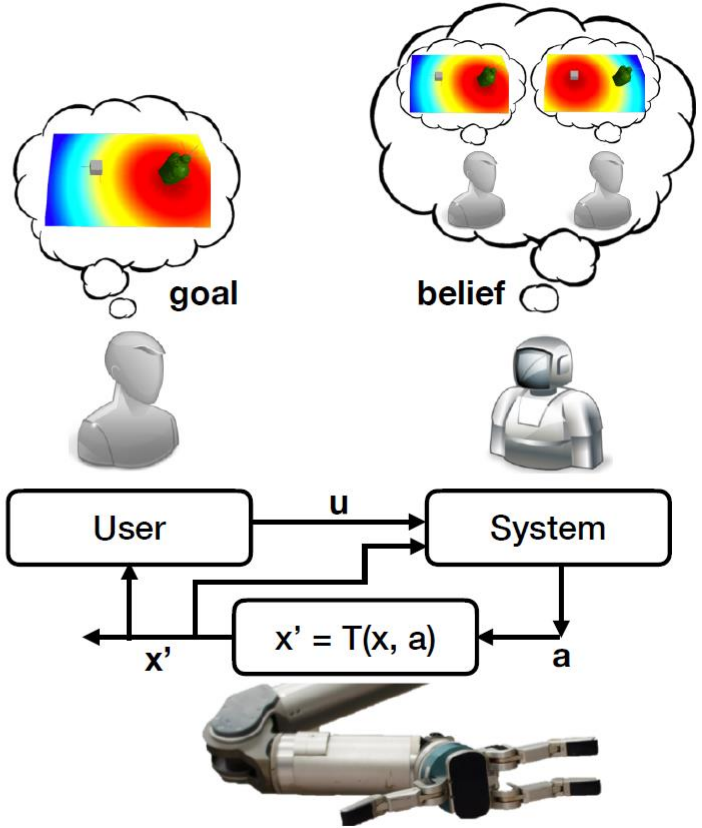
- System dynamics: x' = T(x, a)

- User (MDP) as $(X, U, T, C_g^{\text{usr}})$
  - User policy: $\pi_g^{\text{usr}}(x) = p(u|x, g)$
  - MaxEnt IOC: $C_g^{\text{usr}} : X \times U \to \mathcal{R}$

- System (POMDP) as $(S, A, T, C^{\text{rob}}, U, \Omega)$
  - Uncertainty over user's goal
  - System state: $S = X \times G$
  - Observation: user inputs U
  - Observation model Ω

$$p(g|\xi^{0\to t}) = \frac{p(\xi^{0\to t}|g)p(g)}{\sum_{g'} p(\xi^{0\to t}|g')p(g')}$$

  - Cost function $C^{\text{rob}} : S \times A \times U \to \mathcal{R}$



Hindsight Optimization

- MDP solution:

$$V^{\pi^{\mathrm{r}}}(s) = \mathbb{E}\left[\sum_t C^{\mathrm{r}}(s_t, u_t, a_t) \mid s_0 = s\right]$$

$$V^*(s) = \min_{\pi^{\mathrm{r}}} V^{\pi^{\mathrm{r}}}(s)$$

Hindsight Optimization

- MDP solution:

$$V^{\pi^{\mathrm{r}}}(s) = \mathbb{E}\left[\sum_t C^{\mathrm{r}}(s_t, u_t, a_t) \mid s_0 = s\right]$$

$$V^*(s) = \min_{\pi^{\mathrm{r}}} V^{\pi^{\mathrm{r}}}(s)$$

- POMDP solution:

$$V^{\pi^{\mathrm{r}}}(b) = \mathbb{E}\left[\sum_t C^{\mathrm{r}}(s_t, u_t, a_t) \mid b_0 = b\right]$$

$$V^*(b) = \min_{\pi^{\mathrm{r}}} V^{\pi^{\mathrm{r}}}(b)$$

Hindsight Optimization

- MDP solution:

$$V^{\pi^{\mathrm{r}}}(s) = \mathbb{E}\left[\sum_t C^{\mathrm{r}}(s_t, u_t, a_t) \mid s_0 = s\right]$$

$$V^*(s) = \min_{\pi^{\mathrm{r}}} V^{\pi^{\mathrm{r}}}(s)$$

- POMDP solution:

$$V^{\pi^{\mathrm{r}}}(b) = \mathbb{E}\left[\sum_t C^{\mathrm{r}}(s_t, u_t, a_t) \mid b_0 = b\right]$$

$$V^*(b) = \min_{\pi^{\mathrm{r}}} V^{\pi^{\mathrm{r}}}(b)$$

- HOP approximation:

$$\begin{aligned} V^{\mathrm{HS}}(b) &= \mathbb{E}_b\left[\min_{\pi^{\mathrm{r}}} V^{\pi^{\mathrm{r}}}(s)\right] \\ &= \mathbb{E}_g[V_g(x)] \end{aligned}$$

Hindsight Optimization

- MDP solution:

$$V^{\pi^r}(s) = \mathbb{E}\left[\sum_t C^r(s_t, u_t, a_t) \mid s_0 = s\right]$$

$$V^*(s) = \min_{\pi^r} V^{\pi^r}(s)$$

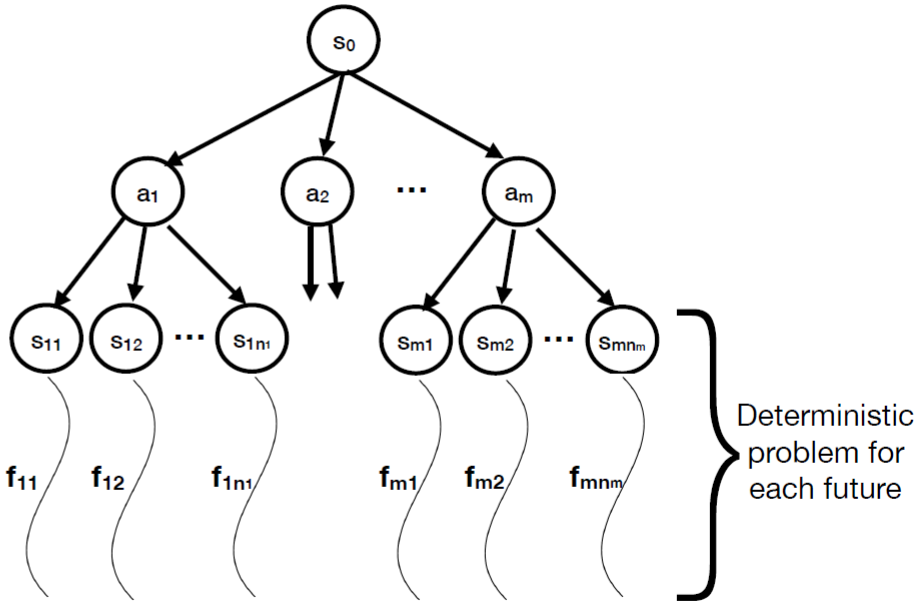
- POMDP solution:

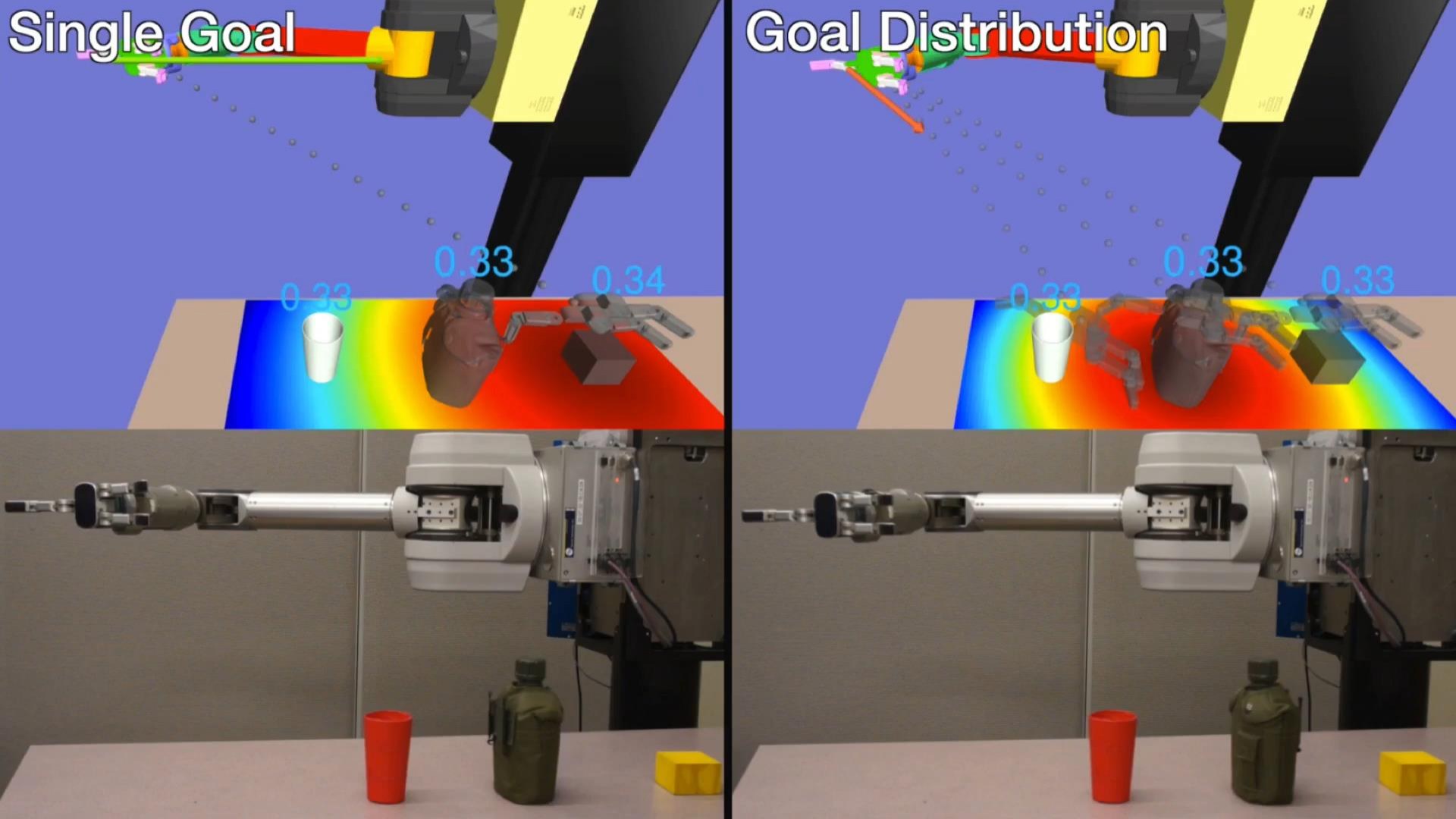
$$V^{\pi^r}(b) = \mathbb{E}\left[\sum_t C^r(s_t, u_t, a_t) \mid b_0 = b\right]$$

$$V^*(b) = \min_{\pi^r} V^{\pi^r}(b)$$

- HOP approximation:

$$V^{\mathrm{HS}}(b) = \mathbb{E}_b\left[\min_{\pi^r} V^{\pi^r}(s)\right]$$
$$= \mathbb{E}_g[V_g(x)]$$

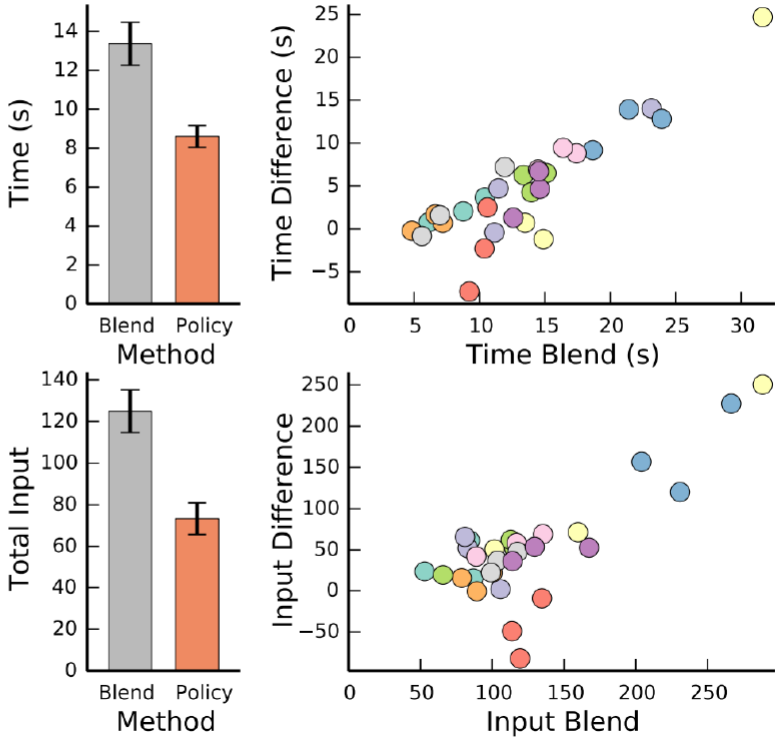



Single Goal
Goal Distribution
0.33
0.34
0.33
0.33
0.33

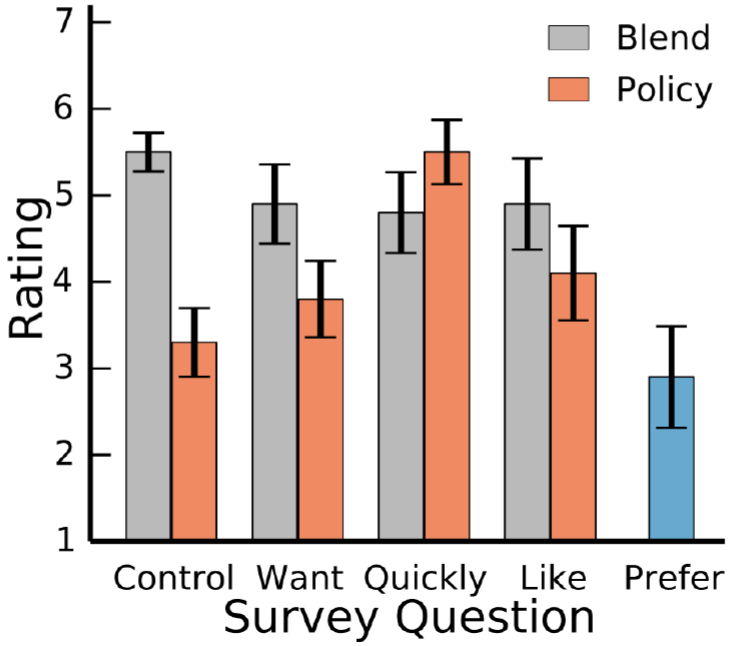
Results

Compare with method that predicts one goal, the proposed method has:

- Faster execution time
- Fewer user inputs



User Study


Blend
Policy
Rating
7
6
5
4
3
2
1
Control
Want
Quickly
Like
Prefer
Survey Question

Limitations

- Requires prior knowledge about the world:
  - a dynamics model that predicts the consequences of taking a given action in a given state of the environment;
  - the set of possible goals for the user;
  - the user's control policy given their goal.

- Suitable in constrained domains where where this knowledge can be directly hard-coded or learned.

- Unsuitable for unstructured environments with ill-defined goals and unpredictable user behavior.

References

- Javdani, S., Srinivasa, S. S., & Bagnell, J. A. (2015). Shared autonomy via hindsight optimization. *Robotics science and systems: online proceedings*, *2015*.
- RSS2015 talk: “Shared autonomy via hindsight optimization”
- Javdani, S., Admoni, H., Pellegrinelli, S., Srinivasa, S. S., & Bagnell, J. A. (2018). Shared autonomy via hindsight optimization for teleoperation and teaming. *The International Journal of Robotics Research*, *37*(7), 717-742.
- ICAPS 2015 talk: "Hindsight Optimization for Probabilistic Planning with Factored Actions”