Online Adaptive Teleoperation via Motion Primitives for Mobile Robots

Xuning Yang · Ayush Agrawal · Koushil Sreenath · Nathan Michael

the date of receipt and acceptance should be inserted later

Abstract Assistive teleoperation aims to help operators control robotic systems with ease. In this work, we present a novel adaptive teleoperation approach that is amenable to mobile systems using motion primitives for long-duration teleoperation, such as exploration using mobile vehicles or walking for humanoid systems. We first describe teleoperation using motion primitives, which are dynamically feasible and safe local trajectories based on a kinematic or dynamic model. We take a predict-and-adapt approach to assistive teleoperation, whereby adaptation is based on the predicted user intent. By representing the operator as an optimizing controller, a probabilistic distribution can be constructed for the available future actions based on some reward function. Adaptation is provided in the form of subsampling, which tailors the set of available actions based on the likelihood of action selection. We describe the framework for general systems and delineate the extrapolation to ground, air, and legged mobile robots, and demonstrate generalizability of this framework on two systems via simulation and experimentation; namely, a quadrotor micro air vehicle, and a simulated 3D humanoid system. Both systems show provably better performance in teleoperation by measures of behavioral entropy.

The work of A. Agrawal and K. Sreenath is supported in part by NSF Grant IIS-1526515.

X. Yang · N. Michael
The Robotics Institute, Carnegie Mellon University
Pittsburgh, PA 15213, USA
E-mail: {xuning, nmichael}@cmu.edu

A. Agrawal · K. Sreenath
Department of Mechanical Engineering, University of California
Berkeley, CA 94720, USA
E-mail: { ayush.agrawal, koushils }@berkeley.edu

1 Introduction

We present a novel task-agnostic user-independent adaptive teleoperation framework for mobile robots using motion primitives. Teleoperated mobile robots are often used in dynamic environments, subject to fluctuations in dynamics. For long-duration maneuvers such as exploration using mobile vehicles and walking for humanoid systems, information regarding the environment and operator proficiency is often not available; and the task is usually not well defined. Our framework provides a novel approach in assisting operators in teleoperation that is task-agnostic as it does not require the operator to provide a well-defined task in the form of a defined goal or descriptive behavior; and user-independent as it does not require prior information on the operator's skill level and teleoperation style. The proposed approach emphasizes three key properties: 1) generalizability of adaptive teleoperation to mobile systems that are amenable to motion primitives, while preserving feasibility and safety; 2) task-agnostic in inferring operator intent in order to provide predictive assistance in teleoperation, and 3) retention of full control for the operator during assistance.

This paper extends the core result of [60] to systems to which motion primitive based methods are readily applied. First, we elaborate the trajectory-based teleoperation scheme that robustly and reliably allows joystick-based control of robots with continuous and hybrid dynamics using motion primitives. We additionally show that motion primitives can be constructed with known and unknown dynamics of the robot. Second, we describe adaptative teleoperation using motion primitives. User intent is represented by probabilistic distribution over the set of available actions, which is translated into assistance by adapting the available range of motion for the operator at the next time step. The proposed strategy is generalizable to systems that are amenable to continuous input motion primitives. By providing a mapping

from inputs to a set of possible motion primitives, we move the user intent inference problem from the input space to the state space. Fig. 1 provides an illustrative diagram of the proposed approach.

Teleoperation. In direct teleoperation of mobile robots, operators issue inputs, such as linear and angular velocities, via an external input device to the robot. The corresponding input is tracked by the controllers of the robot, without regard for the feasibility and safety of the input. As mobile robots are often operating in dynamic environments with disturbances, the operator is required to be vigilant, reactive, and precise in issuing correct inputs in order to keep the robot safe. However, it is well known that humans are imperfect controllers that are subject to time delays of 140-200 milliseconds [38], and also are prone to errors due to physiological stress and fatigue [5, 38] that could introduce instability in the system. Thus teleoperation in a dynamic environment is difficult and stress-inducing for the operator, and may pose danger to the robot and its surroundings.

Humans are, however, recognizably superior as integrated planners and high-level controllers, traits highly desirable in long-duration tasks such as exploration or driving in cluttered, dynamic environments [18]. Humans are more proficient at processing high-resolution visual information than the current state-of-the-art perception systems for navigation, and are able to identify goals and plan trajectories more efficiently than most of the available processing power available on mobile platforms. We recognize human's ability to naturally optimize and conduct motion planning in teleoperation [10] while learning the dynamics of the system over time [38], and leverage this property in our adaptive framework.

To mitigate human error and to allow the operator plan the robot trajectories with certainty, we present a novel abstraction of system-specific inputs, or *actions*, into the state space. Given an action and a kinematic or dynamic model, we map the input using the dynamics into a single motion primitive. Motion primitives are well-known tools in planning for manipulation, gait, and mobile systems [8, 24, 48]. In this work, we leverage this technique for teleoperation. Given all of the degrees-of-freedom (DOF) one can control using a given external input device, we define the action space as the dense discretization of the input for each DOF, which is directly mapped to a set of motion primitives. For systems with a known kinematic or dynamic model, the motion primitives are, in fact, equivalent to input-based teleoperation as one can forward propagate the model for the selected action; however, any kinematic model that does not violate non-holonomic assumptions may be used instead of the full dynamics of the robot. By construction, the direct correspondence between the action space and the state space allows teleoperation to occur directly in the state space of the robot. Motion primitive based teleoperation allows the operator to solely act in the role of the planner, alleviating the operator from having to provide high frequency and reactive inputs in the presence of disturbances. In this paper, we will present motion primitives for ground vehicles and quadrotor air vehicles, as well as for a 3D humanoid model.

Intent Inference. The notion of *intent* vastly differs in the context of assistive technologies, as well as inference methods in shared autonomy problems. In [36], Kulič and Croft define intent as binary variables based on user traits. McLachlan et al. define intent as direct system inputs [39], similarly with [58]. Wang et al. represent intent as a latent variable in a probabilistic model [57]. Task-based intent definitions are observed in [16, 23], and gait modes defined for different walking speeds and standing is defined in [54, 55, 33, 32]. Goal prediction is a common representation of user intent [1, 6, 17], similarly with spatial trajectories [11, 13, 26, 35]. Methods based on Inverse Reinforcement Learning [15, 28, 45] require the explicit definition of a set of discrete goals, but optimize a goal-dependent cost function in order to predict a specific goal. Many of these inference and assistance methods train user intent models offline, and query the model online, and often are limited to classes of known tasks and learn models for these specific classes. We focus on user-independent, online approaches for assisted control that allow for adaptive teleoperation independent of prior knowledge of the operator.

We take inspiration from [9] and represent the user as an optimal controller. Consequently, user intent is modeled as a reward function. We assume that the user optimizes a reward function over time, and issues an action that most closely reflects the optimal reward. This choice of intent representation allows assistance to be formulated for perpetual tasks, such as navigation and exploration for ground and aerial vehicles, and gait movements for humanoid systems.

Assistance via Adaptation. A variety of mathematical frameworks have been proposed for assistance. We restrict the discussion of assistance to mathematical frameworks that aid the user in achieving their intended goal, and disregard discussions on interface-specific assistance (i.e. haptic feedback), as we assume that full observability of the state is provided to the operator by means of sufficient visual feedback. In manipulation, assistance is most notably given in the form of arbitration [1, 28, 27, 45], although methods such as snapping to the closest goal [34] and potential field [2] have also been used. For mobile systems, similar arbitration methods have been used [3, 16, 19], although Bayesian methods [11, 12] and formulating *Partially Observable Markov Decision Processes* (POMDPs) [11, 31] have been applied. Furthermore, trajectory manipulation assistance is proposed in [6, 23]. For gait systems, assistance has been provided in hardware-level controller designs, and few assistance approaches have been offered for adaptation to high-level user intent [30].

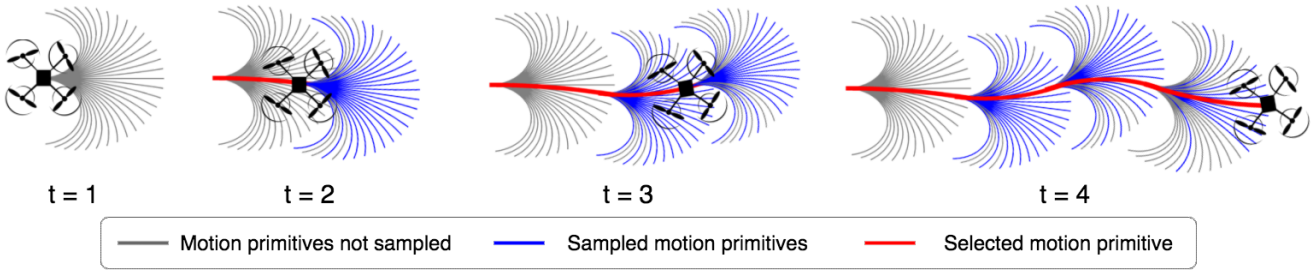


Fig. 1: An illustrative graphic of the proposed approach. The user begins each trial with access to an uniformly dense motion primitive library (MPL) that varies only in angular velocity. Over time, our algorithm updates the belief distribution over the set of motion primitives with respect to the operator's intent. At each input time t, a subset of the motion primitives is sampled with respect to the belief distribution. The operator input maps to a motion primitive from the subsampled set via a selector function.

In this work, we introduce a novel assistance methodology by moderating the available range of motion according to the predicted directional intent via sampling. By inferring directional intent, the proposed method achieves task-agnostic in that it does not require prior knowledge in the form of a defined goal or descriptive behavior[1] that the operator intended for it to perform. Contrary to other methods, limitation of the available motion primitive set still allows the user to retain control over the robot. *The implicit assumption here is that a human's criterion of optimality will be satisfied with some bound around the quantifiably optimal value; thus restriction of the allowable set of motion primitives will actively aid the user.* By subsampling the available range of motions, we ensure that the user's choice of motion follows the probabilistic distribution of the directional intent of the operator.

The main contribution of this paper is to extend the adaptive teleoperation framework, which was first introduced in [60], to show application of the proposed methodology to mobile systems amenable to motion primitives. In this work, we extend the assistance strategy to include hybrid systems, such as gaited systems, in addition to continuous-dynamic quadrotor air vehicles. We elaborate on several choices of motion primitive based on kinematic and dynamic models for continuous differential drive systems, quadrotors, and humanoid systems, and present results for a quadrotor and bipedal humanoid model. We derive generalizability of the proposed adaptive teleoperation framework to any system that is amenable to continuous-input motion primitive based teleoperation, as shown in this paper via two representative systems. Furthermore, we compare our approach to two baseline tests: a naïve filtering method on previous inputs, and direct velocity-based teleoperation without adaptation. We evaluate the proposed and baseline approaches using behavioral entropy techniques and show provably better performance of the resulting trajectories and adherence to the user's directional intent than baseline methods.

This paper proceeds as follows: the framework is introduced in detail in Section 3 with the application to the quadrotor and humanoid systems. Specifically, we introduce motion primitives for differential drive robots, quadrotors in Section 3.1.1 and humanoids in Section 3.1.2, and discuss intent models and adaptation in Section 3.2. We discuss validation methods and experimental results in Section 4 for teleoperating a quadrotor and a humanoid in Sections 4.3 and 4.4 respectively. The paper concludes with future work in Section 5.

2 Related Work

Assistive technologies have primarily been applied to specific system applications, notably in the field of manipulation and vehicle control. Most assistive technologies first predict the intent of the operator, then employ a shared control strategy to aid the operator in reaching their goal. In this section, we detail past works in intent prediction and shared autonomy, and the assumptions that are made with respect to specific application domains.

Intent Prediction. Prediction methods vary from probabilistic representations to machine learning methods given some definition of intent. In [58], a direct robot *input* is inferred using a physics-based model. For intent modeled as a *latent variable*, [57] constructs a probabilistic representation for predicting measurement and transition models using Gaussian Processes (GP), whereas [53] employs a single latent variable to represent user behavior trained using an artificial neural network (ANN). *Task*-based intent can be inferred using pre-defined task features [16] or probabilistically using Gaussian Mixture Autoregression with statistical features [23]. For *trajectory* inference, a mixture distribution over a set of composite trajectories for multiple agents is used in

[1] An example of goal-based task definition is a command with a fixed goal location in the configuration space, such as *go to the door*. A description based task is a command that may describe an objective that is nebulous in the execution required to achieve it; for example, *inspect the building for damages*.

[35], whereas a Bayesian trajectory recognition framework that provides a probabilistic distribution over a set of possible trajectories to possible goals is used in [11, 13, 26]. If intent is described as a *cost function*, [29] defines a joint user-robot cost function which is refined iteratively using Kullback-Liebler Divergence based on the principle of minimum cross-entropy. Intent defined as *goals* can be inferred using a Voronoi diagram combined with local Gaussian Mixture Models [14], using inverse models of system states [6], using artificial potential fields [17], or using Maximum Entropy inverse optimal control (MaxEnt IOC) to construct a distribution of all possible goal states, assuming the user is approximately optimizing some cost function for their intended goal [15, 28, 45]. In controlled settings, intent representation trends toward goal-based methods, whereas dynamic environments typically utilize task or trajectory based methods.

Shared Autonomy. Various methods can be used to provide control guidance. In the hierarchy of control [51], if the operator takes on a *supervisory* role, then the operator can provide high-level commands by selecting a task or setting a goal [50], or provide a trajectory shape for a vehicle to follow [18]. For *shared control* methods, an intended policy is usually predicted, and the final system input is a function of both the predicted policy and the user's actual policy. If the policy is in the form of an *input*, linear arbitration [15] is a widely used method to allocate control between the system and the user given a particular arbitration function [1, 3, 16, 27, 28, 19, 45]. Some examples of arbitration functions are based on uncertainty [40], a max function over probability [16], or manually tuned [27]. For policies modeled as a POMDP, the choice of action can either be influenced by the user input [11, 31], or it could be used to indicate whether to follow the user's policy or the predicted policy [42]. If the predicted policy is a *trajectory*, [6] generates safe mini-trajectories for each incremental waypoint while [23] uses a cooperative motion planner to optimize the robot's trajectory to the forecasted one. [12] describes three Bayesian approaches for providing navigational assistance: a Maximum Likelihood (ML) approach that chooses the trajectory that maximizes the user model, a Maximum A Posteriori (MAP) approach that maximizes the posterior probability, and a greedy POMDP approach for multi-model estimation. Assistance approaches vary in terms of the level of control abstraction from the actual user input, from distinct arbitration to the user input acting as a prior in Bayesian methods.

3 Adaptive Teleoperation Framework

The adaptive teleoperation framework is presented in this section for general mobile robots. The framework proceeds as follows: Operators control the vehicle with an external control device. Inputs are mapped to motion primitives, and the user-issued input selects a motion primitive which the robot follows. We define a user intent model over the space of motion primitives with the operator acting as an optimizing controller. Using the inference model, we provide assistance in the form of adaptation by subsampling the set of available motion primitives. However, a key assumption of the following framework is that the operator is *not adversarial*. This is to say, the operator will always act in favor of their intended motion and will not attempt to circumvent assistance.

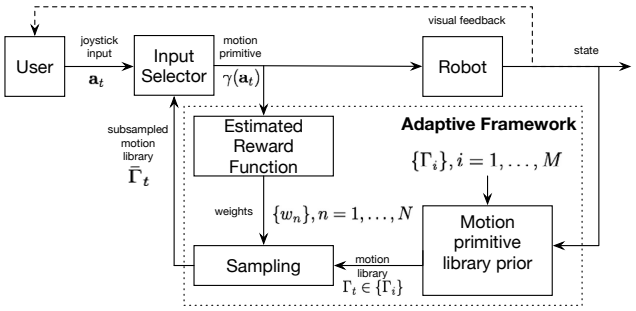


Fig. 2: System diagram of the proposed adaptive framework. The operator issues an input from a joystick and selects a motion primitive as described by the input selector. While the robot follows the motion primitive, the choice of motion primitive is used in the adaptation framework to generate a subsampled set of the full underlying motion primitive library that tailors better towards the user's intent. At the next time step, the joystick action is mapped to a motion primitive in the subsampled set. Relevant notations can be found in Table 1.

Table 1: Notations and Definitions

$\mathbf{a}_\tau$	select action at timestep τ
$\gamma_\tau^{\mathbf{a}_n}(t)$ or $\gamma_\tau^n(t)$	motion primitive correlated with action $\mathbf{a}_n$ at timestep τ, evaluated at time t, $0 < t \leq T$
Γ_τ	motion primitive library at timestep τ
$\bar{\Gamma}$	subsampled motion primitive library
w_n	probability of motion primitive n being selected
T	fixed time duration for length of motion primitives

The system diagram of the framework is shown in Fig. 2. This section begins with a discussion of teleoperation using motion primitives in Sect. 3.1 and discuss motion primitive generation for continuous and hybrid systems. Then, we define the user intent model in Sect. 3.2. Adaptive teleoperation using the inferred user intent is presented in Sect. 3.3.

3.1 Motion Primitive Based Teleoperation

We define an *action* to be a set of inputs provided via an external input device. For q input dimensions, an action is denoted as $\mathbf{a} = \{a_1, \ldots, a_q\}$. It is assumed that the external input device used to teleoperate the robot is continuous, i.e. a joystick interface rather than directional buttons that increase the input value at a fixed discretization. Furthermore,

we assume that each of the input domain a_i is convex. For the continuous input, we discretize the input values in each dimension to obtain the action space, which consists of q sets of N finite actions.

For each action, a motion primitive maps the action space to the state space $\gamma^{\mathbf{a}} : \mathbb{R}^q \rightarrow \mathbb{R}^s$ where s is the dimension of the state. The span of the action set generates a *motion primitive library* (MPL), which is a group of motion primitives parameterized by an action set, and denoted by $\Gamma = \{\gamma^{\mathbf{a}_i}\},\ i = 1, \ldots, N$, with N actions, $\{\mathbf{a}_i\}, i = 1, \ldots, N$. Hence, an MPL will contain $q \times N$ motion primitives, and is treated as an indexed set of motion primitives parameterized by the action space. We further define the set of MPL to be a *motion primitive library collection*, which is denoted by $\{\Gamma_j\}, j = 1, \ldots, M$. The simplest class of motion primitives are obtained by forward propagating the kinematics (or dynamics) of the robot. If the kinematic model is known, then a motion primitive can be defined as the position and higher order derivatives obtained by propagating the selected action by some constant amount of time, T. In this case, teleoperation using this model can be shown to be equivalent to input-based teleoperation. An example of this is the unicycle model for differential drive robots, discussed in Sect. 3.1.1. However, if the kinematics or dynamics of the robot are not known or cannot easily be forward propagated, any choice of motion primitives can be used given the robot's dynamical constraints and non-holonomic properties (if any) are not violated.

Between sequential inputs, continuity and smoothness are enforced in the transition from one motion primitive to another. If the user provides an input at time t_f, where $0 < t_f \leq T$, then $\gamma_t(t_f) = \gamma_{t+1}(0)$ for position and higher derivatives of the sequential motion primitives. As unsmooth transitions between motion primitives may saturate motor limitations, the design of the transitions is crucial for providing teleoperation safety. Furthermore, motion primitives enable teleoperation in the presence of disturbances, as various disturbance rejection strategies can be incorporated into the vehicle controller to keep the vehicle stable and ensure the vehicle adheres to the issued motion primitive which is a local trajectory. In addition, teleoperation safety can further be guaranteed by performing obstacle avoidance by pruning motion primitives given a local map, which is not discussed in this paper. A method of constructing dynamically feasible ground and aerial motion primitives is presented in Section 3.1.1, and gait libraries are presented in Section 3.1.2.

3.1.1 Motion Primitives for Vehicles

Forward-arc motion primitives are a type of parameterized motion primitive for ground vehicles. These are created by propagating the dynamics of a unicycle model with a constant linear and angular velocity for a specified amount of time, T [47]. The motion primitives are given by the solution to the unicycle model (1) and its higher order derivatives:

$$\mathbf{x}_{t+T} = \mathbf{x}_t + \begin{bmatrix} \frac{v_{xt}}{\omega_t}(\sin(\omega_t T + \theta_t) - \sin(\theta_t)) \\ \frac{v_{xt}}{\omega_t}(\cos(\theta_t) - \cos(\omega_t T + \theta_t)) \\ \omega_t T \end{bmatrix}, \tag{1}$$

where $\mathbf{x}_t = [x_t, y_t, \theta_t]^{\mathrm{T}}$ represents the pose of a ground vehicle at time t in the body frame, and v_{xt}, ω_t are the linear and angular velocities of the vehicle at time t in the body frame, respectively. The action space is given by uniformly dense sets of actions, denoted as the following: $\mathcal{V}_x = \{v_{xi}\}, i = 1, \ldots, N_{v_x}$, $\Omega = \{\omega_j\}, j = 1, \ldots, N_\omega$. A single motion primitive at each time t is denoted by $\gamma_t = \{\mathbf{a}_t, T\}$ where $\mathbf{a}_t = \{v_{xt}, \omega_t\}$ and T is some fixed duration indicating the length of the motion primitive. Note that the operator has the ability to generate a new input at any time t_f for $0 < t_f \leq T$. Thus if no new input is received, then $\mathbf{a}_{t+1} = \mathbf{a}_t$ and $\Gamma_{t+1} = \Gamma_t$.

For differential drive ground vehicles, these motion primitives are equivalent to input-based teleoperation. However, we choose to extend these motion primitives to aerial robots by incorporating linear dynamics for the vertical velocity. For a quadrotor air vehicle, the forward arc motion primitives become (2) and its higher order derivatives:

$$\mathbf{x}_{t+T} = \mathbf{x}_t + \begin{bmatrix} \frac{v_{xt}}{\omega_t}(\sin(\omega_t T + \theta_t) - \sin(\theta_t)) \\ \frac{v_{xt}}{\omega_t}(\cos(\theta_t) - \cos(\omega_t T + \theta_t)) \\ v_{zt} T \\ \omega_t T \end{bmatrix}, \tag{2}$$

where $\mathbf{x}_t = [x_t, y_t, z_t, \theta_t]^{\mathrm{T}}$ is the pose of the aerial robot with θ_t being the yaw of the vehicle. The additional input dimension has action space $\mathcal{V}_z = \{v_{zk}\}, k = 1, \ldots, N_{v_z}$ for vertical velocity. The corresponding action then becomes $\mathbf{a}_t = \{v_{xt}, v_{zt}, \omega_t\}$. For ground vehicles, the heading of the vehicle is fixed to the yaw of the vehicle by nature. Although aerial platforms such as quadrotors can independently control heading from yaw, we maintain the use of a unicycle model by ensuring that the heading is equivalent to the yaw of the vehicle, as humans naturally optimize for curved trajectories in robot control [10] with heading aligned with the zero yaw angle.

From Eq. (1), it is evident that forward-arc motion primitives preserve continuity up to velocity in concatenating consecutive motion primitives, as it is the solution to the kinematic model of ground vehicles. To apply these motion primitives for quadrotor vehicles, we need to preserve the smoothness between consecutive motion primitives. For quadrotors, smoothness usually requires continuity up to jerk [41]. In our scenario, we require smoothness up to acceleration and assume it is sufficient for teleoperation. This is to say, trajectories must be continuous in jerk but may be nonsmooth. To address continuity in concatenating motion primitives, we pre-compute a *motion primitive library collection*, which con-

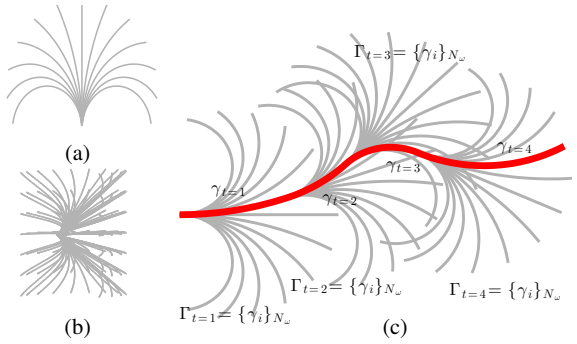


Fig. 3: (a) A motion primitive library generated for a 2D unicycle model. For easier visualization, only discretizations in angular velocity ω are shown. (b) 3D motion primitive library with variations in angular velocity ω and vertical velocity v_z. (c) A trajectory formulated over four time steps. The selected motion primitives at each time step (in red) form a single trajectory.

tains a set of motion primitive libraries with various initial conditions $\{\mathbf{x}_0, \dot{\mathbf{x}}_0, \ddot{\mathbf{x}}_0\}$ in the body frame. These initial conditions are obtained by discretizing all possible higher order derivatives; for our case, we consider continuity in velocity and acceleration with other higher orders held at zero. Each MPL in the collection computes a unique transition from the initial condition to the final condition of the MPL, which specifies a linear and angular velocity. This is to say, at transition time t_f of timestep t, the final condition of timestep t uniquely identifies a MPL from the MPL collection such that at timestep $t+1$, $[\mathbf{x}_t\ \dot{\mathbf{x}}_t\ \ddot{\mathbf{x}}_t](t_f) = [\mathbf{x}_{t+1}\ \dot{\mathbf{x}}_{t+1}\ \ddot{\mathbf{x}}_{t+1}](0)$.

An example of the forward-arc MPL is shown in Fig. 3a-3b for ground and air vehicles, respectively. Figure 3c depicts the MPL selected at each time t with the initial condition matching that of the current robot state. The set of each selected motion primitive at each time t forms a smooth trajectory in a fixed frame.

3.1.2 Motion Primitives for Gait Systems

We utilize a hybrid dynamical model of *flat-footed* walking for a 3D humanoid robot, and its associated Poincaré map, to generate a set of motion primitives for walking along different heading angles. The hybrid model consists of alternating phases of a continuous single-support (swing) phase and an instantaneous, double-support (impact) phase. The hybrid dynamical model of this system is given by:

$$\Sigma : \begin{cases} \dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})u, & \mathbf{x} \notin S, \\ \mathbf{x}^+ = \Delta(\mathbf{x}^-), & \mathbf{x} \in S, \end{cases} \tag{3}$$

where $\mathbf{x} = (q, \dot{q})^T$ is the state of the robot, and $q \in \mathcal{Q} = \mathbb{R}^3 \times SO(3) \times \mathbb{R}^{15}$ is the configuration of the robot, $u \in \mathbb{R}^{15}$ are the control inputs, which are the joint torques, and $\mathbf{x}^+$ and $\mathbf{x}^-$ are the pre- and post-impact states respectively. The switching surface S is defined as the set of states $(q, \dot{q})^T$ where the transition from single-support to double-support takes place and is defined as,

$$S = \{(q, \dot{q}) \in \mathcal{TQ} \mid p^v_{sw}(q) = 0, \dot{p}^v_{sw}(q) < 0\}, \tag{4}$$

where p^v_{sw} is the vertical position of the swing foot and $\mathcal{TQ}$ denotes the tangent space of $\mathcal{Q}$.

To begin defining motion primitives for the humanoid system, we use Hybrid Zero Dynamics (HZD) [21, 59, 25]. A set of outputs $y(\mathbf{x}, \alpha, \beta) \in \mathbb{R}^{15}$ is defined for the hybrid system in (3) and an input-output linearizing controller drives these outputs to zero exponentially. Here, $\mathbf{x}$ is the state of the robot as defined in (3); and parameters α, β, are sets of Bézier Polynomial coefficients that parametrize the desired outputs, which encode the desired walking behavior, including step length, step width, forward walking speed, and heading angle. The parameters α are obtained through a constrained nonlinear program that searches for periodic solutions $\mathbf{x}^*$ of the closed-loop system in (3) (See [25] for details). The parameters $\boldsymbol{\beta}$ correspond to heading angles of the humanoid. In particular, $\beta = 0$ corresponds to walking along a straight line.

Similar to motion primitive generation for vehicles, we define the action space to be of the heading angle, $\mathbf{a}_k = \{\psi[k]\}$. The corresponding action space is given by $\Psi = \{\psi_j\}, j = 1, \ldots, N_\psi$, with the constraint $\psi_j \in [\psi_{min}\ \psi_{max}]$.

To generate a library of dynamically feasible gaits, Γ, corresponding to ψ, the input-output linearizing controller is integrated with a Poincaré map based stride-to-stride controller [52, 7, 44]. The Poincaré map, P, is a composition of the swing-phase and impact dynamics (3) and transfers the state of the robot one walking step ahead, as follows:

$$\mathbf{x}[k+1] = P(\mathbf{x}[k], \beta[k]). \tag{5}$$

The state of the robot at step k, $\mathbf{x}[k]$, consists of the robot's position, orientation, and joint angles. Computing the Poincaré map analytically is difficult, but the linearized Poincaré map can be numerically obtained [7]. The linearized Poincaré map is defined as,

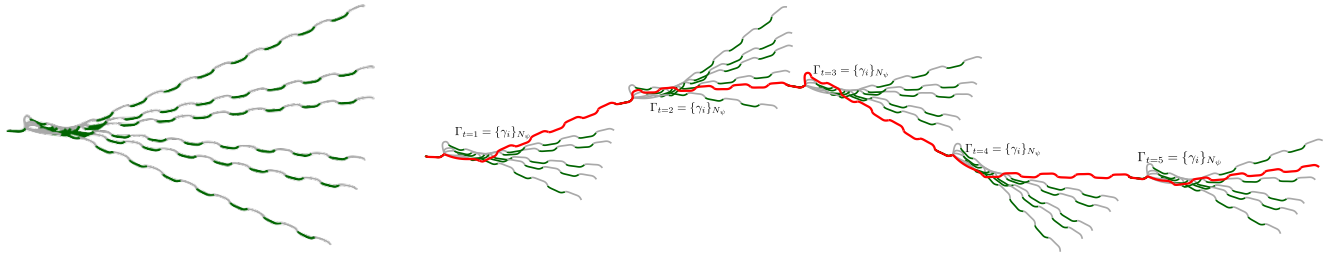
$$\delta\mathbf{x}[k+1] = \mathbf{A}\delta\mathbf{x}[k] + \mathbf{B}\beta[k], \tag{6}$$

where $\delta\mathbf{x} := \mathbf{x} - \mathbf{x}^*$ and $\beta[k]$ is a linear feedback controller:

$$\beta[k] = -K_\delta \cdot (\mathbf{x}[k] - \mathbf{x}(\psi[k])). \tag{7}$$

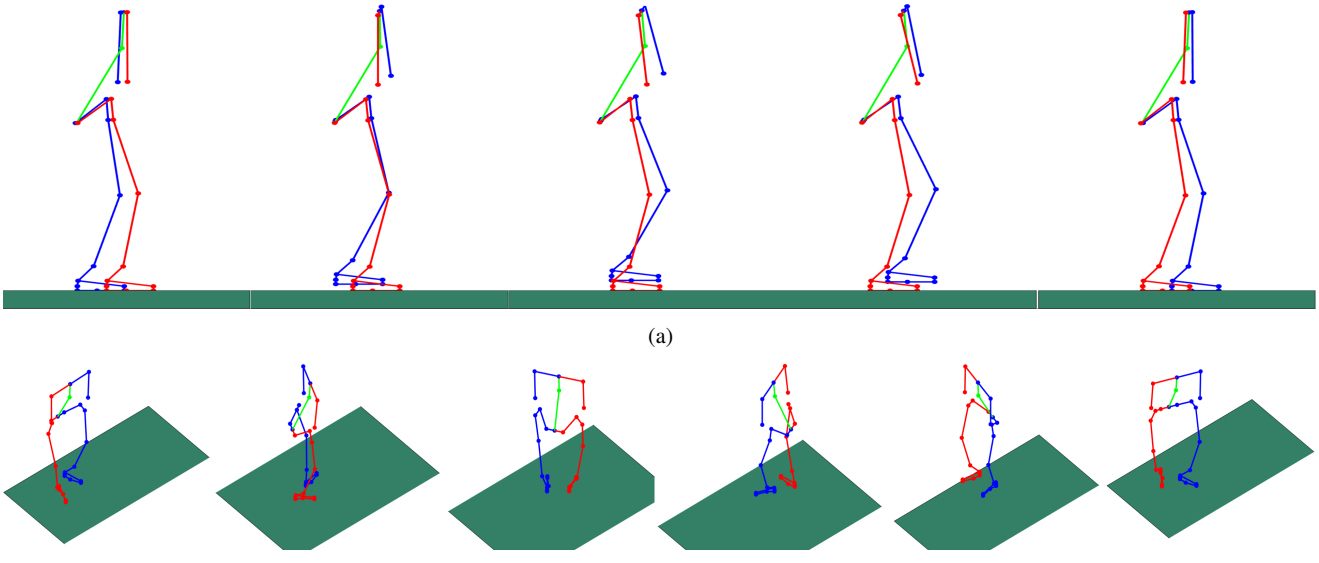
The gain K_δ can be obtained using the Discrete Time Linear Quadratic Regulator (DLQR) method for the linearized Poincaré map, and $\mathbf{x}(\psi[k])$ is the state of the robot corresponding to the desired heading angle at step k. The pseudo-algorithm for motion primitive generation for humanoid systems is provided in Table 2.

The Poincaré map based controller allows the abstraction of continuous-time joint control and the hybrid dynamics of the system in the motion-primitive generation process. This enables a parameterization of the motion primitives



(a) (b)

Fig. 4: (a) Some illustrative motion primitives for different desired heading angles (starting from zero heading). Green lines indicate the center of mass trajectories when the right foot is the stance foot, and grey lines indicate the center of mass trajectories when the left foot is the stance foot.(b) A trajectory formulated over 50 walking steps. The selected motion primitive at each walking step (in red) form a trajectory.



(a)

(b)

Fig. 5: Simulation of the 3D bipedal system with 21 joints. (a) Snapshots of a single walking step achieved using the Hybrid Zero Dynamics control framework. (b) Snapshots of the simulated robot walking on a circular path.

by the heading angle, forgoing independent joint control. Fig. 4a shows the path traversed by the robot for different desired heading angles while Fig. 5b shows snapshots of the robot walking on a circular path, both using the event-based controller described above.

3.2 User Intent Model and Inference

We assume that the operator inherently optimizes a reward function, but the action selected at each time step does not optimally reflect this function. This is the notion of "good-enough" – that humans operate within some region of optimality but do not always select a single optimal action [37]. In this particular problem, the operator issues action $\mathbf{a}$ at each input time t, which is in some neighborhood of $\mathbf{a}^\star$, the optimal action that satisfies:

$$\mathbf{a}^\star = \underset{\mathbf{a}}{\operatorname{argmax}}\, R_t(\gamma^{\mathbf{a}}) \approx \underset{\mathbf{a}}{\operatorname{argmax}} \sum_{i}^{Q} \alpha_t^i \phi_t^i(\gamma^{\mathbf{a}}), \tag{8}$$

Table 2: Pseudo-algorithm of the motion primitive generation for humanoid sytems

1. Compute (using trajectory optimization) the gait parameters α such that the solution of the closed-loop system in (3) is periodic. (See [25, 22, 21]).
2. Numerically compute the linearized Poincaré map around the periodic solution $\mathbf{x}^*$ such that $\delta\mathbf{x} = \mathbf{A}\delta\mathbf{x} + \mathbf{B}\beta$ (See [7]).
3. Compute the gain-matrix K_δ by solving the discrete-time Riccati equation (this is the DLQR method and the matrix K_δ can be obtained).
4. Compute the parameter $\beta[k]$ for the desired heading angle ψ_k. (See (7)).
5. The continuous-time HZD controller then drives the outputs $y\left(x, \alpha, \beta[k]\right)$ to zeros exponentially.

where ϕ^i's are basis functions defined with respect to quantifiable natural human behavior, for a total of Q basis functions. We assume the reward function is composed of linear basis terms. Thus, the inference problem is the prediction of the underlying reward function, $\hat{R}_t = \sum_i^Q \hat{\alpha}^i \phi_t^i$, from the series of noisy user inputs $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{t-1}\}$.

Using this model, we infer the user's behavior as the solution to the following optimization problem:

$$\begin{aligned}\hat{\gamma}_{t+1} &= \underset{\gamma_{t+1}^{\mathbf{a}}}{\operatorname{argmax}}\, R_t(\gamma_{t-m:t}, \gamma_{t+1}^{\mathbf{a}}) \\ &= \underset{\gamma_{t+1}^{\mathbf{a}}}{\operatorname{argmax}} \sum_i^Q \alpha^i \phi_t^i(\gamma_{t-m:t}, \gamma_{t+1}^{\mathbf{a}}),\end{aligned} \tag{9}$$

where $\gamma_{t-m:t}$ represents a trajectory formed by the past m motion primitives at time t, and $\gamma_{t+1} \in \Gamma_{t+1}$. Equation (9) is the key assumption that reflects the notion of "good-enough," which allows us to define assistance that inherently reflects this property.

Given an estimate of the user's reward function, we can iteratively update the probability of a motion primitive being selected at the next time iteration. The prediction update is provided in Eq. (10). Given an initial uniform distribution over the set of motion primitives, one can update the probability of the n-th motion primitive being chosen at the next time step given the probability of the reward based on a segment of the previous trajectory:

$$\begin{aligned}p(\gamma_{t+1}^n | \gamma_{t-m:t}, \hat{R}_t) &= \frac{p(\hat{R}_t | \gamma_{t-m:t}, \gamma_{t+1}^n) p(\gamma_{t+1}^n | \gamma_{t-m:t})}{p(\hat{R}_t | \gamma_{t-m:t})} \\ &= \eta\, p(\hat{R}_t | \gamma_{t-m:t}, \gamma_{t+1}^n) p(\gamma_{t+1}^n | \gamma_{t-m:t})\end{aligned} \tag{10}$$

where the past trajectory is denoted by $\gamma_{t-m:t}$, which is the past m selected motion primitives as observed at time t, and the distribution $p(\hat{R}|\gamma_{t-m:t}, \gamma_{t+1})$ estimates the reward function of the user, as introduced in Sect. 3.2, with η being the normalization weight.

3.2.1 Reward bases

For mobile robots, the operator acts as a high level trajectory planner. To this end, we define several bases that are intrinsic to human motion planning, inspired by [35, 43]. As the operator issues inputs over time, the inputs are made to follow a trajectory that is more conducive to human motions; thus a natural choice of bases is to optimize for what humans deem to be natural trajectories.

It is entirely possible for bases to only be functions of past trajectories (called *hindsight bases*) for the purposes of intent prediction. However, it is possible to incorporate prior knowledge, either in the form of a known trajectory or other environment models such as disturbance or obstacles fields. Here, we present three hindsight bases for the quadrotor air vehicle (smoothness, orthogonality, and time), and an additional distance metric given a desired trajectory to follow (distance error). These are defined as follows:

Smoothness We define smoothness as the magnitude of change in the input:

$$\phi_{\text{smoothness}}(\gamma_{t+1}, \gamma_{t-m:t}) = \sum_{j=t-m+1}^{t+1} \|\mathbf{a}_j - \mathbf{a}_{j-1}\|_1 \tag{11}$$

Orthogonality We penalize any drastic deviation in heading from the previous trajectory. This is defined using a simple ratio for three points as follows:

$$\phi_{\text{orthogonality}}(\gamma_{t+1}, \gamma_{t-m:t}) = \frac{\|p_t - p_{t-m}\| - \|p_{t+1} - p_t\|}{\|p_{t+1} - p_{t-m}\|} - 1 \tag{12}$$

where each point $p_\tau \in \mathbb{R}^3$ is the position at time step τ at time T obtained from the motion primitive: $p_\tau = \gamma_\tau^{\mathbf{a}}(t = T)$.

Time We define the cost of time as the inverse of the linear body velocity, which allows for inference over the desired speed of motion:

$$\phi_{\text{time}} = \frac{1}{v_x} \tag{13}$$

Distance Error If a desired trajectory is provided, the distance error between the motion primitive and the trajectory can be calculated by approximating the area or volume for 2D and 3D trajectories respectively [4]. Given two paths $\{p_1, ..., p_n\}$ and $\{q_1, ..., q_n\}$, the distance error between the two can be defined as:

$$\phi_{\text{distance}} = \sum_{i=2}^{n} \frac{1}{2} \left(\|p_i - p_{i-1}\|_2 + \|q_i - q_{i-1}\|_2 \right) \|p_i - q_i\|_2 . \tag{14}$$

3.2.2 Reward function estimation

The reward function is inferred from a past window of m motion primitives. The belief distribution of the reward $p(\hat{R}|\gamma_{t-m:t}, \gamma_{t+1}^n)$ is computed over the entire motion primitive library. For this study, we assume that each input dimension is conditionally independent. The belief distribution is computed using an online function approximation to estimate the reward function, $\hat{R}_t = \sum_i^Q \hat{\alpha}^i \phi_t^i$. We employ Locally Weighted Projection Regression (LWPR), a computationally efficient online method for local approximations of high dimensional nonlinear functions [56] to estimate the reward function. The incremental algorithm performs global function approximation by taking a weighted sum of the local regressors that influence the region.

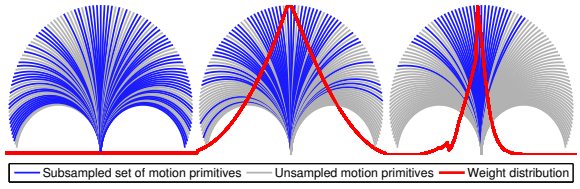


Fig. 6: Motion primitives and distribution over the motion primitives at selected times along a racetrack at timesteps $t = 0$, $t = 5$, and $t = 150$ respectively. The prediction becomes more peaked near the mean of the predicted motion primitive.

The regression over the reward bases is defined with respect to a linear global reward function, which is estimated using LWPR. A global LWPR model is sufficient for tasks with a single intent. However, tasks that may require dynamic changes in intent – for example, high-speed driving or flight with inference over the angular velocity – such tasks require a more temporally local prediction in order to adjust to these changes. To continue leveraging the speed of LWPR, we keep a rolling queue of l LWPR models. At each input time, a model M_t is popped off the queue and a new model M_{t+l} is added to the queue. Each model in the queue is updated with the data received at that time. The prediction at $p(\hat{R}|\gamma_{t-m:t}, \gamma^n_{t+1})$ is then generated with model M_t. This is a local batch estimation method for time-varying intent functions that has shown to be computationally tractable over the time span of interest. More succinct online regression methods that are amenable to time varying models and conducive to a small number of data points are an area of interest and will be addressed in future work.

3.3 Adaptation using the operator intent model

Given a dense set of motion primitives and a probabilistic distribution over these motion primitives, assistance is provided by limiting the set of available motion primitives to a subset adhering to the operator's intent. By defining a "good-enough" region of interest around the optima, the set of motion primitives within the region remains safe and feasible, and most closely reflect the operator intent. We assume that within some short duration, the operator's intent does not fluctuate widely and single instances of input that are outside of the interest region are unintended by the operator.

We adaptively modify the subset of available motion primitives from an underlying set of motion primitives such that the density of the subsampling reflects the reward function distribution $p(\hat{R}|\gamma_{t-m:t}, \gamma_{t+1})$. By use of motion primitives, a particular choice of action $\mathbf{a}_t$ at time t is represented by its parameterized motion primitive γ_t for some fixed duration T. The key insight here is that we have removed the dependency of trajectories on the continuous input space, thus allowing inference to be made over a set of motion primitives, which is the set of local trajectories that is safe and feasible.

To construct the set of available motion primitives, we subsample motion primitives using importance sampling. As such, within the region of interest, fine-grained control of the action is preserved by the density of motion primitives. The subsampled set closely adheres to the user's underlying intent and circumvents misaligned motions to the user's interest.

Let the weight of the n^{th} motion primitive be $w_n = p(\hat{R}|\gamma_{t-m:t}, \gamma^n_{t+1})$. Given a motion primitive library Γ of size N, we sample K motion primitives using the weights $\{w_n\}, n = 1, \ldots, N$ with replacement such that we obtain a subsampled set:

$$\bar{\Gamma} = \{\gamma^k\} \subseteq \Gamma, k = 1, \ldots, K, \tag{15}$$

which is the available set of motion primitives provided to the operator.

To map the user input to a specific motion primitive in the subsampled set, a selector function (16) is used to select the motion primitive with the closest parameterization of the actual user input $\mathbf{a}_{\text{input}}$ in the continuous input case:

$$\gamma_{\text{selected}} = \gamma\{\underset{\mathbf{a}}{\text{argmin}}\, \mathbf{a} - \mathbf{a}_{\text{input}}\} \in \bar{\Gamma}. \tag{16}$$

The adaptation methodology is summarized in Algorithm 1. A visualization of this algorithm is provided in Fig. 6.

Algorithm 1 Algorithm for updating construction of the subsampled set

$\Gamma_{t+1} \leftarrow \{\gamma^n\};\ n = 1, \ldots, N;\ \Gamma_{t+1} \sim U(0, 1)$
$\bar{\Gamma}_{t+1} \leftarrow \emptyset$
for $n = 1 : N$ motion primitives **do**
    Calculate weights for each motion primitive $w_n = p(\hat{R}|\gamma_{t-m:t}, \gamma^n_{t+1})$
end for
for $k = 1 : K$ **do**
    Sample $\gamma^k \in \Gamma_t$ with probability w_k with replacement
    $\bar{\Gamma}_{t+1} \leftarrow \bar{\Gamma}_{t+1} + \gamma^k$
end for
return $\bar{\Gamma}_{t+1}$

4 Experiments

We present simulation and experiment results for a quadrotor micro-air vehicle and simulation results for a bipedal system. We introduce a method validation criterion in Section 4.1. Vehicle experiment results are presented in Section 4.3, and bipedal results are presented in Section 4.4.

Fig. 7: The quadrotor and joystick used in the experiment (top) and the quadrotor in flight (bottom).

Table 3: LWPR parameters for estimating the reward function for both scenarios

D_{init}	7	w_{gen}	0.3
α_{init}	250	w_{cutoff}	0.5
meta	false	penalty	1.0

Table 4: Motion primitive library parameters used in the experiments.

Quadrotor parameters

$v_{x\,\max}$	0.5 m/s	$v_{x\,\min}$	-0.5 m/s	N_{v_x}	101
$v_{z\,\max}$	0.5 m/s	$v_{z\,\min}$	-0.5 m/s	N_{v_z}	101
$\omega_{\max}$	3 rad/s	$\omega_{\min}$	-3 rad/s	N_{ω}	101

Gait parameters

$\psi_{\max}$	0.4 rad	$\psi_{\min}$	-0.4 rad	N_{ψ}	101

4.1 Method Validation

We validate the efficacy of our method versus baseline approaches using *Behavioral Entropy* [20], an online, nonintrusive measure of workload that characterizes operator efficiency. Behavioral entropy characterizes the efficiency of an operator's interaction with a robot and measures the consistency of an operator with respect to a predictive model based on the observed behavior. The implication that skilled or desired behavior is consistent becomes quantifiable in the form of entropy.

For joystick-based continuous-input systems, Joystick Steering Entropy (JSE) [46] is a behavioral entropy technique that uses a Taylor series approximation model. We evaluate our assistive teleoperation approach as compared to no adaptation using this metric, and briefly describe JSE below.

At time t, the error between a second order Taylor approximation and the actual input is evaluated:

$$e_t = u_t - \hat{u}_t$$

$$\hat{u}_t = u_{t-1} + (u_{t-1} - u_{t-2}) + \frac{1}{2}((u_{t-1} - u_{t-2}) - (u_{t-2} - u_{t-3}))$$

where $u \in \mathbb{R}$ is the continuous input. A frequency distribution of the error e_t is constructed and divided into 9 bins. The total steering entropy, Hp, for each trial is given by:

$$Hp = \sum_i -P_i \log_9 P_i. \tag{17}$$

A slight modification to the algorithm is made by padding the proportion P_i of each bin by $\epsilon \approx 1e\text{-}6$ in order to avoid asymptotes:

$$P_i = \frac{n_i}{\sum_i n_i} + \epsilon,\ i = 1, \ldots, 9. \tag{18}$$

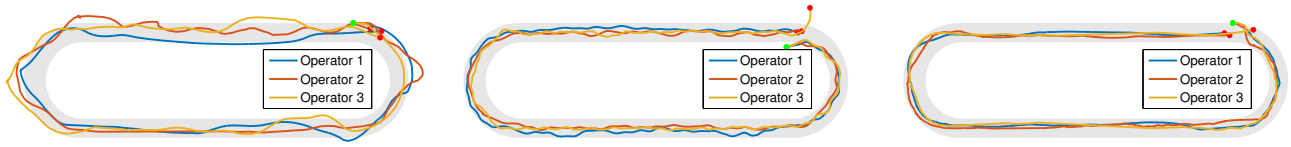
As efficiency increases, the steering entropy decreases accordingly.

We further evaluate the correctness of prediction by computing the *frequency of inputs* over time. We assert that once the performance is sufficient as deemed by the operator, the operator will provide inputs less frequently as the generated motion primitive accurately follows their intent. For mobile vehicles, we compute the frequency of inputs as number of inputs received per second, and for the humanoid robot, we compute the frequency of inputs as the number of inputs received per walking step.

4.2 Parameters and definitions

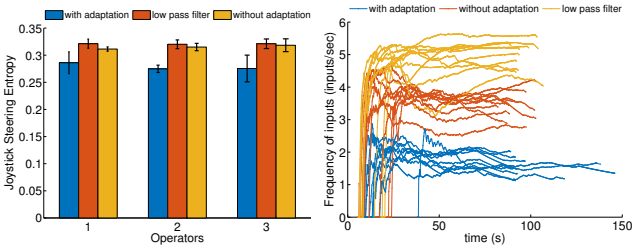
The LWPR parameters used in this study are provided in Table 3. All parameters that are not listed here take on default values provided by [56]. For both scenarios, we bound the rate at which inputs are received to 10 Hz. However, real life bipedal systems typically operate at 1 step per second, which is slower than typical human walking speeds (1.2 m/s) [49]. Thus, input rates of 10 Hz are reasonable given that the operator generates a maximum of 2 inputs per step.

The MPL was generated using the discretization and range values provided in Table 4. We select velocity bounds that are conducive for non-aggressive maneuvers for this study and leave adaptation for aggressive maneuvers as future work. We choose a discretization of 0.01 m/s for the linear and vertical velocities, and 0.06 rad/s (approximately 3.44 deg/s) as the discretization in angular velocity. We find this discretization to be sufficient for the purposes of this study and does not interfere with the operator's intentions. One could choose an arbitrarily large number of motion primitives with finer discretizations, however, we find that this adds to the computational complexity and our choice of granularity does not pose a hindrance to the teleoperation performance.



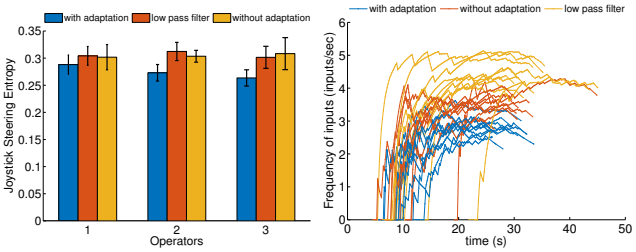
(a) (b) (c)

Fig. 8: Illustrative odometry results for three operators teleoperating a quadrotor one lap around a simulated racetrack (a) without adaptation, (b) with a low pass filter with $\alpha = 0.5$, and (c) with adaptation. Adaptation results in the smoothest trajectory while low pass filter causes the operator to overcorrect due to smoothing effects.



(a) (b)

Fig. 9: Results of a quadrotor completing one lap around a racetrack, where adaptive teleoperation is compared with a low pass filter with $\alpha = 0.5$ and with no adaptation. (a) Joystick steering entropy averaged over 5 trials for each test case. (b) Frequency of inputs over time.



(a) (b)

Fig. 10: Results of a quadrotor completing a lemniscate motion, where adaptive teleoperation is compared with a low pass filter with $\alpha = 0.5$ and with no adaptation. (a) Joystick steering entropy averaged over 5 trials for each test case. (b) Frequency of inputs over time.

4.3 Quadrotor Vehicle Experiments and Results

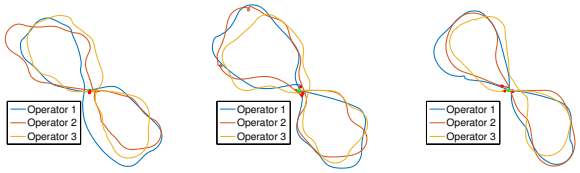
Two scenarios are used to test the proposed framework using a quadrotor vehicle: a racetrack of size $30\,\text{m} \times 10\,\text{m}$ at a height of $1\,\text{m}$ and a lemniscate motion with length of $5\,\text{m}$. The racetrack scenario is used to validate single-intent long-duration adaptation. Due to the size of the racetrack, we perform this task in simulation only. To demonstrate our framework for dynamic intent and the fidelity of our simulation framework, we evaluate the lemniscate motion in both simulation and in the flight arena (Fig. 7). For simplicity, we only perform inference over the heading of the robot for all of the subsequent experiments. This is to say, the motion primitive library only contains variations in ω.

For all of the subsequent experiments, we compare teleoperation results using our adaptation framework to that without adaptation, and we perform additional trials using a exponential moving average filter with weight $\alpha = 0.5$.

Racetrack. Operators are asked to teleoperate a simulated quadrotor vehicle using a joystick and follow the racetrack to the best of their ability. As the trajectory that the user is trying to follow is known, this information can be incorporated into the distance error basis as discussed in Sect. 3.2.1. The racetrack is tested with 15 trials (five with adaptation, five without, and five with a low pass filter). All of the trials are randomized in arbitrary order, and are anonymized to the operator. For consistency, we perform the same experiment with three colleagues at Carnegie Mellon University, not including the authors. All operators do not have any prior experience in teleoperating a quadrotor aerial vehicle using this joystick setup.

The resulting trajectory with adaptation, without adaptation, and with the low pass filter is shown in Fig. 8. We observe that teleoperation with adaptation produces very smooth trajectories, much smoother than the trajectories without adaptation. Filtered inputs produce trajectories that are much more controlled than without adaptation, but we observe that the trajectories demonstrate periodic behaviors. This is likely due to the lagging effect of the filter: as the inputs are smoothed over time, this creates a lagging effect which causes the operators to over-correct their inputs. We assert that filtering the operator's inputs over time misrepresents their true intent and causes them to provide control-level actions instead of providing only navigational inputs.

We further evaluate our method using Joystick Steering Entropy and frequency of inputs over time, as shown in Fig. 9. While filtered inputs sometimes exhibit higher entropy due to the operator's overcorrection, adaptation consistently produced lower entropy than without adaptation, and with a low pass filter, with an average reduction in entropy of 13%. While a reduction in entropy indicates that the user's inputs are smoother, and consequently result in smoother trajectories, we notice that frequency of inputs with adaptative



(a) (b) (c)

Fig. 11: Illustrative odometry results for three operators teleoperating a quadrotor with a free-form (i.e without pre-defined trajectory) lemniscate motion (a) without adaptation, (b) with a low pass filter with $\alpha = 0.5$, and (c) with adaptation. Adaptation results in the smoothest trajectory while low-pass filtering causes the operator to overcorrect due to smoothing effects.

teleoperation is consistently lower than with filtered input or without adaptation as observed in Fig. 9b. This is a key improvement in adaptation as users provide *less* frequent inputs when the behavior of the robot is aligned with the intended behavior.

Lemniscate. For this scenario, a specific trajectory was not provided to the operator. Instead, the operator is asked to teleoperate a quadrotor vehicle via joystick to perform a free-hand lemniscate motion in simulation and in the flight arena. As no prior trajectory is provided, predictions are based purely on the hindsight bases as defined in Sect. 3.2.1. This experiment uses the incremental adaptive approach outlined in Sect. 3.2.2.

We first observe the resulting trajectory as shown in Fig. 11. We observe slightly smoother performance with adaptation (Fig. 11c) than without (Fig. 11a), and similar overcorrecting behavior for the filtered input is again evident. However, we observe the performance with adaptation is not as smooth as what was observed with the racetrack scenario. We attribute this to the fast changes in directional intent that may cause high variance in the incremental prediction.

Joystick steering entropy and frequency of inputs are also evaluated, as shown in Fig. 10a. As with the racetrack scenario, we notice that while filtered inputs and teleoperation without adaptation show fluctuating entropies, teleoperation with adaptation shows slightly lower entropy but the reduction is less than expected. We posit that this is a result of the high variance with the incremental prediction, which we will address as future work in Sect. 5. Frequency of inputs also is reduced over time and is lower than that with filtered input and without adaptation, however the margin between the average frequency of inputs of adaptive teleoperation and non-adaptive teleoperation is much smaller than that of the racetrack scenario.

The adaptation process is shown for three different runs of the lemniscate with adaptation in Fig. 13. The raw input is compared to the predicted mean of the input, where we

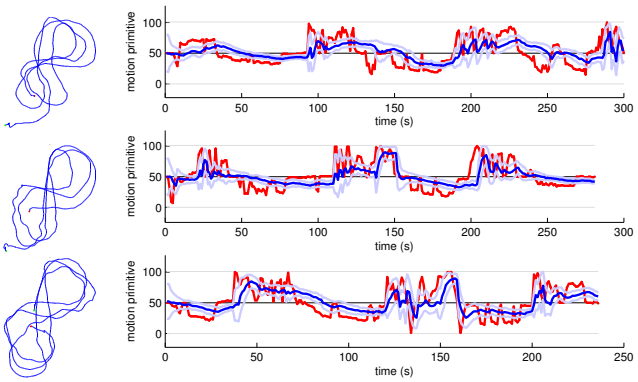


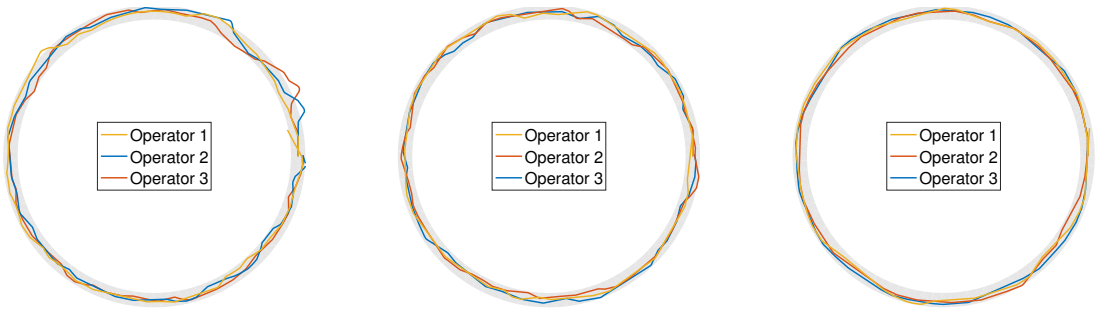
Fig. 13: Odometry (left) and mean of prediction (right) for three of the experimental trials for the lemniscate experiment. The actual input (red) is compared to the mean of predicted distribution (blue) with light blue highlighting upper and lower bounds based on the covariance. Near regions of rapid user input changes, variance increases and the mean adjusts to the new prediction.

observe a smoothing of the user input. In addition, we notice that near regions of rapid user input changes, variance of the prediction increases and decreases accordingly as the mean adjusts over time.

We now validate the number of motion primitives. From the trajectories in the above scenarios, the choice of discretizing 101 motion primitives seemed reasonable and conducive to good performance. Furthermore, we visualize the sampled motion primitives over time for several trials, as shown in Fig. 17a and Fig. 17b. For the racetrack scenario, we observe that the number of subsampled motion primitives quickly decreases over time. The lemniscate result is more interesting. We see that as directional intent changes, the number of subsampled motion primitives increases suddenly and converges again.

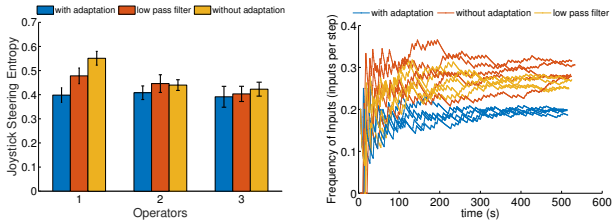
4.4 Bipedal Experiments and Results

For the humanoid robot, we consider two scenarios: walking on a circular track of radius 20m and a sinusoidal path of frequency 1m^{-1} and amplitude 1.5m. Operators were asked to follow the desired trajectory as closely as possible in a simulation environment, using a joystick to control the heading angle and with visual feedback of the robot's base frame pose. We compare teleoperation using our adaptation framework to teleoperation without adaptation, and we perform an additional trial using a moving average filter with weight $\alpha = 0.5$. In both scenarios, the desired trajectory was provided to the operators. A total of fifteen trials (five with adaptation, five without adaptation, and five with a moving average filter) were conducted for each scenario across three different operators from Carnegie Mellon University. All of the trials



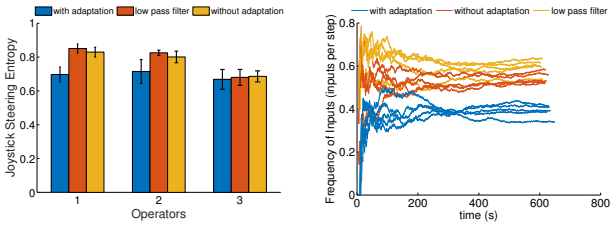
(a) (b) (c)

Fig. 12: Illustrative odometry results for three operators teleoperating humanoid robot on a circular trajectory (a) without adaptation, (b) with a low pass filter with $\alpha = 0.5$, and (c) with adaptation.



(a) (b)

Fig. 14: Results of a humanoid robot completing one lap around a circular track, where adaptive teleoperation is compared with a low pass filter with $\alpha = 0.5$ and with no adaptation. (a) joystick steering entropy averaged over 5 trials for each test case. (b) and frequency of inputs over time.



(a) (b)

Fig. 15: Results of a humanoid robot traversing a sinusoidal track, where adaptive teleoperation is compared with a low pass filter with $\alpha = 0.5$ and with no adaptation. (a) joystick steering entropy averaged over 5 trials for each test case. (b) and frequency of inputs over time.

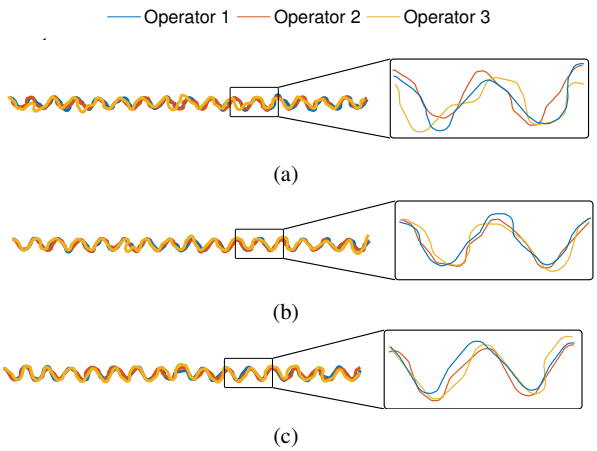
are randomized in arbitrary order, and are anonymized to the operator.

Circular Track. The resulting trajectories with adaptation, without adaptation, and with the low-pass filter are shown in Fig. 12. Similar to the racetrack trajectory for the aerial vehicle, we observe much smoother trajectories for teleoperation with adaptation. While the resulting trajectory for teleoperation with the low-pass filter is slightly smoother than without adaptation, the overcorrecting behavior is not as prominent as was the case with the aerial vehicle, potentially due to the slower speed at which the humanoid operates at.

A reduction in Joystick Steering Entropy is observed for teleoperation with adaption as compared to without adaptation as well as with the low-pass filter (Fig. 14a). We also observe a lower input frequency across multiple trials (Fig. 14b).

Sinusoidal Track. Similar to previous experiments, qualitative observations indicate smoother trajectories for teleoperation with adaptation as seen in Fig. 16. The over-correcting behavior with the low-pass filter is more prominent, as rapid changes demand users change their joystick inputs often.

As with previous experiments, joystick steering entropies



(a)

(b)

(c)

Fig. 16: Illustrative odometry results for three operators teleoperating a humanoid robot on a sinusoidal trajectory (a) without adaptation, (b) with a low pass filter with $\alpha = 0.5$, and (c) with adaptation.

are lower for teleoperation with adaptation, while teleoper-

ation without adaptation and with the low-pass filter show fluctuating behaviors (Fig. 15a). Frequency of inputs is also consistently lower across multiple trials for teleoperation with adaptation (Fig. 15b).

For both scenarios, the number of sampled motion primitives over time for several trials consistently does not exceed the initial number of motion primitives chosen, as shown in Fig. 17c and Fig. 17d.

5 Conclusion and Future Work

In this work, we showed generalizability of the adaptive teleoperation framework first presented in [60] to mobile systems, and demonstrated provably better performance than naïve baseline methods. We experimentally test our framework via teleoperation of a quadrotor air vehicle and a 3D bipedal model. The proposed approach demonstrated lower behavioral entropy, indicating increased performance. Furthermore, we demonstrated the correctness of our prediction algorithm to the underlying operator intent by showing a reduction in the frequency of inputs over time.

Applications involving teleoperation present a rich opportunity in testing the proposed framework. This method of intent inference and adaptation can be applied to exoskeleton control using torso angles in assistive robotics for constrained operators. Assisted driving is another area that may leverage our framework to alleviate errors resulting from fatigue from drivers and aid inexperienced drivers to mitigate fatal errors.

Our adaptive teleoperation provides a robust platform for future work in investigating shared control and intent inference. As we have presented a methodology that can be used when environmental information is not available, we plan to investigate the addition of local environment models via online environment model generation techniques to influence local planning using motion primitives given the system. Furthermore, we plan to investigate our methodology of predictive assistance in various scenarios and evaluate its efficacy. We introduced adaptation with respect to dynamic intent via directional changes; however, we aim to characterize dynamic intent detection and prediction more precisely for unknown and dynamic environments.

References

1. Admoni H, Srinivasa S (2016) Predicting User Intent Through Eye Gaze for Shared Autonomy. In: The 2016 AIII Fall Symposium Series: Shared Autonomy in Research and Practice. Technical Report FS-16-05
2. Aigner P, McCarragher B (1997) Human integration into robot control utilising potential fields. In: Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on, IEEE, vol 1, pp 291–296
3. Anderson SJ, Walker JM, Iagnemma K (2014) Experimental Performance Analysis of a Homotopy-Based Shared Autonomy Framework. In: IEEE Trans. Systems, Man and Cybernetics, vol 44, pp 190–199
4. Bachrach AG (2013) Trajectory Bundle Estimation For Perception-Driven Planning. PhD thesis, Dept. Elect. Eng. and Comp. Sci., Massachusetts Institute of Technology, Cambridge, MA, USA
5. Borghini G, Astolfi L, Vecchiato G, Mattia D, Babiloni F (2014) Measuring Neurophysiological Signals in Aircraft Pilots and Car Drivers for the Assessment of Mental Workload, Fatigue and Drowsiness. Neuroscience and Biobehavioral Reviews 44:58–75
6. Carlson T, Demiris Y (2008) Human-Wheelchair Collaboration Through Prediction of Intention and Adaptive Assistance. In: Proc. of the IEEE Int. Conf. on Robot. and Autom., Pasadena, CA, pp 3926–3931
7. Chevallereau C, Grizzle JW, Shih CL (2010) Steering of a 3d bipedal robot with an underactuated ankle. In: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, IEEE, pp 1242–1247
8. Cohen BJ, Subramania G, Chitta S, Likhachev M (2011) Planning for manipulation with adaptive motion primitives. In: Proc. of the IEEE Int. Conf. on Robot. and Autom., IEEE, pp 5478–5485
9. Crandall JW, Goodrich MA (2002) Characterizing Efficiency of Human Robot Interaction: A Case Study of Shared-Control Teleoperation. In: Proc. of the IEEE/RSJ Int. Conf. on Intell. Robots and Syst., Lausanne, Switzerland, pp 1–6
10. Delson N, West H (1994) Robot Programming by Human Demonstration: The Use of Human Inconsistency in Improving 3D Robot Trajectories. In: Proc. of the IEEE/RSJ Int. Conf. on Intell. Robots and Syst., pp 1248–1255
11. Demeester E, Hüntemann A, Vanhooydonck D, Vanacker G, Brussel HV, Nuttin M (2008) User-Adapted Plan Recognition and User-Adapted Shared Control: A Bayesian Approach to Semi-Autonomous Wheelchair Driving. In: Autonomous Robots, pp 193–211
12. Demeester E, Hüntemann A, Poorten EV, Schutter JD (2012) ML, MAP and Greedy POMDP Shared Control: Comparison of Wheelchair Navigation Assistance for Switch Interfaces. In: Proc. of the Int. Sym. on Robotics, Taipei, Taiwan, pp 1106–1111
13. Demeester E, Poorten EV, Alexander H, Lau B, Kuderer M, Fossati A, Roig G, Boix X, Ristin M, Hofmann M (2012) Robotic ADaptation to Humans Adapting to Robots: Overview of the FP7 project RADHAR
14. Derry M, Argall B (2014) A Probabilistic Representation of User Intent for Assistive Robots. In: Int. Conf. on Intell. Robots and Syst. Workshop on Rehabilitation and Assistive Robotics, Chicago, IL, USA
15. Dragan AD, Srinivasa SS (2013) A policy-blending formalism for shared control. Int J Robot Res 32(7):790–805
16. Gao M, Oberl J, Schamm T, Marius JZ (2014) Contextual Task-Aware Shared Autonomy for Assistive Mobile Robot Teleoperation. In: Proc. of the IEEE/RSJ Int. Conf. on Intell. Robots and Syst., Chicago, IL, USA, pp 3311–3318
17. Glinton R, Owens S, Giampapa J, Sycara K, Lewis M, Grindle C (2005) Intent Inference Using a Potential Field Model of Environmental Influences Intent Inference Data Flow. In: 2005 7th International Conference on Information Fusion, IEEE
18. Gnatzig S, Schuller F, Lienkamp M (2012) Human-machine interaction as key technology for driverless driving - A trajectory-based shared autonomy control approach. In: Proceedings - IEEE International Symposium on Robot and Human Interactive Communication, pp 913–918
19. Goil A, Derry M, Argall BD (2013) Using Machine Learning to Blend Human and Robot Controls for Assisted Wheelchair Navigation. In: IEEE Int. Conf. Rehabil. Robot, Seattle, WA, USA
20. Goodrich MA, Boer ER, Crandall JW, Ricks RW, Quigley ML (2004) Behavioral entropy in human-robot interaction. Tech. rep., Brigham Young University

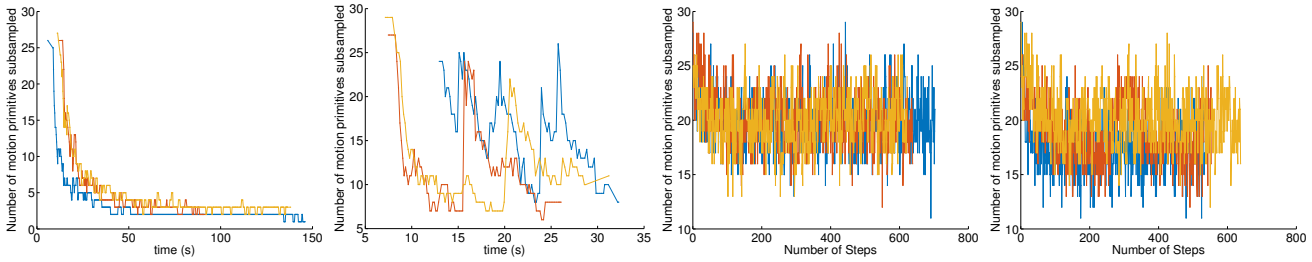


Fig. 17: Subsampling of motion primitives over time for quadrotor teleoperation in the (a) racetrack and (b) lemniscate scenario, and bipedal teleoperation in the (c) sinusoidal and (d) circle scenario. Three random trials are shown for adaptation over angular velocity. This illustrates that a reasonable and non-restricting discretization of the motion primitive is sufficient for constructing the motion primitive library.

21. Grizzle JW, Abba G, Plestan F (2001) Asymptotically stable walking for biped robots: Analysis via systems with impulse effects. IEEE Transactions on automatic control 46(1):51–64
22. Grizzle JW, Chevallereau C, Ames AD, Sinnet RW (2010) 3D bipedal robotic walking: models, feedback control, and open problems. IFAC Proceedings Volumes 43(14):505–532
23. Hauser K (2013) Recognition, prediction, and planning for assisted teleoperation of freeform tasks. Autonomous Robots 35(4):241–254
24. Hauser K, Bretl T, Harada K, Latombe JC (2008) Using motion primitives in probabilistic sample-based planning for humanoid robots. Algorithmic foundation of robotics VII pp 507–522
25. Hereid A, Cousineau EA, Hubicki CM, Ames AD (2016) 3D dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics. In: IEEE Int. Conf. on Robot. and Autom.
26. Huntemann A, Demeester E, Poorten EV, Brussel HV (2013) Probabilistic approach to recognize local navigation plans by fusing past driving information with a personalized user model. In: Proc. of the IEEE Int. Conf. on Robot. and Autom., Karlsruhe, Germany, pp 4376–4383
27. Jain S, Argall B (2016) An Approach for Online User Customization of Shared Autonomy for Intelligent Assistive Devices. In: Proc. of the IEEE Int. Conf. on Robot. and Autom., Stockholm, Sweden
28. Javdani S, Srinivasa SS, Bagnell JA (2015) Shared Autonomy via Hindsight Optimization. In: Proc. of Robot.: Sci. and Syst., Rome, Italy
29. Javdani S, Bagnell JA, Srinivasa SS (2016) Minimizing User Cost for Shared Autonomy. In: 2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp 621–622
30. Jimenez-Fabian R, Verlinden O (2012) Review of control algorithms for robotic ankle systems in lower-limb orthoses, prostheses, and exoskeletons. Medical engineering & physics 34(4):397–408
31. Kaelbling LP, Littman ML, Cassandra AR (1998) Planning and Acting in Partially Observable Stochastic Domains. Artificial Intelligence 101(1–2):99–134
32. Kawamoto H, Kanbe S, Sankai Y (2003) Power assist method for hal-3 estimating operator's intention based on motion information. In: Robot and human interactive communication, 2003. proceedings. ROMAN 2003. The 12th IEEE international workshop on, IEEE, pp 67–72
33. Kawamoto H, Lee S, Kanbe S, Sankai Y (2003) Power assist method for hal-3 using emg-based feedback controller. In: Systems, Man and Cybernetics, 2003. IEEE International Conference on, IEEE, vol 2, pp 1648–1653
34. Kofman J, Wu X, Luu TJ, Verma S (2005) Teleoperation of a Robot Manipulator Using a Vision-Based Human-Robot Interface. IEEE Transactions on Industrial Electronics 52(5):1206–1219
35. Kretzschmar H, Kuderer M, Burgard W (2014) Learning to Predict Trajectories of Cooperatively Navigating Agents. In: Proc. of the IEEE Int. Conf. on Robot. and Autom.
36. Kulič D, Croft EA (2003) Estimating Intent for Human-Robot Interaction. IEEE Int Conference on Advanced Robotics
37. Loeb GE (2012) Optimal isn't good enough. Biological Cybernetics 106(11):757–765
38. MacAdam CC (2003) Understanding and Modeling the Human Driver. Vehicle System Dynamics 40:1-3(January 2014):101–134
39. McLachlan S, Arblaster J, Liu D, Miro JV, Chenoweth L (2005) A Multi-Stage Shared Control Method for an Intelligent Mobility Assistant. In: IEEE Int. Conf. Rehabil. Robot, Chicago, IL, USA
40. Medina JR, Lorenz T, Hirche S (2015) Synthesizing Anticipatory Haptic Assistance Considering Human Behavior Uncertainty. IEEE Transactions on Robotics 31(1):180–190
41. Mellinger D, Kumar V (2011) Minimum snap trajectory generation and control for quadrotors. In: Proc. of the IEEE Int. Conf. on Robot. and Autom., IEEE, pp 2520–2525
42. Milliken L, Hollinger GA (2016) Modeling User Expertise for Choosing Levels of Shared Autonomy. In: Proc. of Robot.: Sci. and Syst. Workshop on Planning for Human-Robot Interaction, Ann Arbor, MI, USA
43. Mombaur K, Laumond Jp, Yoshida E (2010) An Optimal Control-Based Formulation to Determine Natural Locomotor Paths for Humanoid Robots. Advanced Robotics 24:515–535
44. Motahar MS, Veer S, Poulakakis I (2016) Composing limit cycles for motion planning of 3d bipedal walkers. In: IEEE Conference on Decision and Control. To appear
45. Muelling K, Venkatraman A, Valois JS, Downey JE, Weiss J, Javdani S, Hebert M, Schwartz AB, Collinger JL, Bagnell JA (2015) Autonomy Infused Teleoperation with Application to BCI Manipulation. In: Proc. of Robot.: Sci. and Syst., Ann Arbor, MI, USA
46. Nakayama O, Futami T, Nakamura T (1999) SAE Technical Development of a Steering Entropy Method for Evaluating Driver Workload. Tech. Rep. 724
47. Nelson EA, Michael N (2015) Environment Model Adaptation for Autonomous Exploration. Master's thesis, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA
48. Pivtoraiko M, Kelly A (2011) Kinodynamic motion planning with state lattice motion primitives. In: Proc. of the IEEE/RSJ Int. Conf. on Intell. Robots and Syst., IEEE, pp 2172–2179
49. Ralston HJ (1958) Energy-speed relation and optimal speed during level walking. European Journal of Applied Physiology and Occupational Physiology 17(4):277–283
50. Sa I, Corke P (2014) Vertical infrastructure inspection using a quadcopter and shared autonomy control 92:219–232
51. Sheridan TB, Parasuraman R (2005) Human-Automation Interaction. Reviews of human factors and ergonomics 1(1):89–129
52. Shih CL, Grizzle J, Chevallereau C (2012) From stable walking to steering of a 3d bipedal robot with passive point feet. Robotica 30(07):1119–1130

53. Vanhooydonck D, Demeester E, Hüntemann A, Philips J, Vanacker G, Brussel HV, Nuttin M (2010) Adaptable navigational assistance for intelligent wheelchairs by means of an implicit personalized user model. Robotics and Autonomous Systems 58(8):963–977
54. Varol HA, Goldfarb M (2007) Real-time intent recognition for a powered knee and ankle transfemoral prosthesis. In: Rehabilitation Robotics, 2007. ICORR 2007. IEEE 10th International Conference on, IEEE, pp 16–23
55. Varol HA, Sup F, Goldfarb M (2010) Multiclass real-time intent recognition of a powered lower limb prosthesis. IEEE Transactions on Biomedical Engineering 57(3):542–551
56. Vijayakumar S, D'Souza A, Schaal S (2005) Incremental Online Learning in High Dimensions. Neural Computation 17(12):2602–2634
57. Wang Z, Mülling K, Deisenroth MP, Amor HB, Vogt D, Schölkopf B, Peters J (2013) Probabilistic Movement Modeling for Intention Inference in Human-Robot Interaction. Int J Robot Res 32(7):841–858
58. Wasson G, Sheth P, Huang C, Ledoux A (2004) A Physics-Based Model for Predicting User Intent in Shared-Control Pedestrian Mobility Aids. In: Proc. of the IEEE/RSJ Int. Conf. on Intell. Robots and Syst., Sendai, Japan, pp 1914–1919
59. Westervelt ER, Grizzle JW, Chevallereau C, Choi JH, Morris B (2007) Feedback control of dynamic bipedal robot locomotion, vol 28. CRC press
60. Yang X, Sreenath K, Michael N (2017) A framework for efficient teleoperation via online adaptation. In: Int. Conf. Robot. Autom. (ICRA), Singapore