Trust Adaptation Leads to Lower Control Effort in Shared Control of Crane Automation

Alexander Broad[1,3], Jarvis Schultz[2], Matthew Derry[1,3], Todd Murphey[2,4] and Brenna Argall[1,2,3,5]

***Abstract*—We present a shared-control framework predicated on a measure of trust in the operator, that is calculated automatically based on the quality of the interactions between a human and autonomous system. This measure of trust is built upon a control-theoretic foundation that rewards stable operation of the system to give more trusted users additional control authority. The level of control authority is used to modify the human input, and as a result, we observe a minimization of the required effort of the controller. We validate this work within a planar crane environment with a receding horizon controller to assist with the regulation of the system dynamics. The human defines the reference trajectory for the controller. In an experimental study users navigate a suspended payload through a set of maze configurations. We find that adaptation of the trust metric over time provides the benefit of substantially ($p < 0.01$) improving the automated system's ability to modulate the user's input, resulting in stable reference trajectories that require less effort to track. In effect, the human and automation spend less time fighting each other during task execution, suggesting that the automated system and user each have a better understanding of the other's ability.**

I. INTRODUCTION

The increasing pervasiveness, capabilities and complexity of autonomous robots in human environments has highlighted the need for more sophisticated control-sharing techniques that allow humans to interact with, control and shape the behaviors of these systems, while also maintaining a high level of safety. Shared control enables a human and autonomous system to simultaneously control a system. As such, control sharing can create a system that leverages the strengths of each source of control while reducing the effects of the weaknesses. For instance, in the automotive domain shared-control systems have contributed to safer driving by autonomously identifying and correcting common control mistakes made by humans [1]. Shared control research has accordingly seen a recent surge in interest and utility in areas ranging from rehabilitation and assistive robotics [2], [3], to search and rescue [4], [5], to transportation [6], [7].

Of particular importance to consider when developing these control-sharing techniques is that systems exhibiting significant dynamics are difficult for humans to control. Human operators often cope with complex dynamics by sufficiently constraining the system to minimize the effect of the dynamics, for example a crane operator moving a payload very slowly. Alternatively, some form of automated assistance can help to mask the dynamics from the human, for example the unique shape of the JAS-39 airplane [8] creates aerodynamics that are uncontrollable by a human alone, thus significant automated assistance from the flight computer is added to stabilize the aircraft.

Our take on control sharing is to combine the relative advantages of the human and robot partners. In this work, we consider that automation and optimal control techniques are good at controlling highly dynamic systems, but require a reference trajectory to try to stabilize to. While these reference trajectories could come from automated path planners, engaging a human partner has the advantage of using the exceptional perceptual capabilities and situational awareness of humans to operate in dynamic environments. The key is for the human to provide reference trajectories that the automated controller can track.

In this work, one of the primary goals is to maximize stability in complex dynamic systems while taking advantage of humans' perceptual and cognitive adaptability. Towards that end, we develop a human-in-the-loop control framework that reasons explicitly about the amount of control authority that should be allocated to the human based on the trust that the autonomous system has *in the operator*. The purpose of this trust metric is to allow the system to learn how able a user is in providing reference trajectories that can be easily tracked by the automated controller. The adaptive trust metric can then be used to develop a more stable shared-control system. Our approach is novel in that we characterize interactions between the human and autonomous system within the framework of control theory in order to build this formalized notion of trust, with which the autonomous system can modulate control authority.

The proposed shared-control framework is validated on a

Manuscript received: March, 1, 2016; Revised May, 31, 2016; Accepted June, 29, 2016.

This paper was recommended for publication by Editor Yasuyoshi Yokokohji upon evaluation of the Associate Editor and Reviewers' comments. *This material is based upon work supported by the National Science Foundation under Grant CNS 1329891. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the aforementioned institutions.

[1]Alexander Broad, Matthew Derry and Brenna Argall are with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208, USA alex.broad@u.northwestern.edu, brenna.argall@northwestern.edu mderry@u.northwestern.edu

[2]Jarvis Schultz, Todd Murphey and Brenna Argall are with the Department of Mechanical Engineering, Northwestern University, Evanston, IL 60208, USA jschultz@northwestern.edu, t-murphey@northwestern.edu

[3]Alexander Broad, Matthew Derry and Brenna Argall are with the Rehabilitation Institute of Chicago, Chicago, IL 60611, USA

[5]Todd Murphey is with the Department of Physical Therapy and Human Movement Sciences, Northwestern University, Chicago, IL 60611, USA

[5]Brenna Argall is with the Department of Physical Medicine and Rehabilitation, Northwestern University, Chicago, IL 60611, USA

simulated planar crane robot platform in which a human operator is tasked with maneuvering a payload through a maze to different target locations. This platform provides a simulation that approximates a real-world, cyber-physical system problem that exhibits significant system dynamics. In analyzing our approach, we place particular importance on how well the system is able to learn and modulate control authority to a human based on their ability to provide suitable reference trajectories, as this has a direct effect on the stability of the system.

The remainder of this paper is structured as follows: Section II reviews related literature in shared control and the foundation of optimal control theory that this proposed work is built upon. Section III presents our proposed trust formulation, and Section IV describes the implementation of our experimental platform and details of the experiment, followed by results and discussion in Section V and conclusions in Section VI.

II. BACKGROUND AND RELATED WORK

This section provides a brief review of shared-control and the role that trust has played in shared control frameworks. Additionally, the foundation for the receding horizon optimal controller used in this work is presented.

A. Shared Control

An important question for any shared-control system is how best to allocate control authority to maximize the effectiveness of the human-robot team [9]. One approach is to predefine control authority allocations that can be switched in a discrete way in response to some action or input [2]. Alternatively, control authority can be assigned on a continuum such that the granting or retracting of control authority is smooth [6], [10].

Shared control has been used within the field of assistive and rehabilitation robotics to introduce automation to the task of aiding individuals with impairments to control assistive machines [3], [11]. In search and rescue, the shared control may happen at the task level where a human instructs a team of robots to carry out a search task, while leaving lower level control and planning to the robots [12].

Within the shared-control literature, a subset of works model trust in an effort to improve human-robot team performance [13], [14], [15]. In these works, the formulation of trust represents the trust that the human has *in the autonomous system*, so that the automated system can use the model to choose actions to maximize trust [16]. Additionally, trust has been studied in the context of robot-robot teams [14]. Our proposed system is differentiated in that we are instead proposing a formulation of trust that represents the trust that the autonomous system has *in the operator*. This trust is then used to allocate control authority in an effort to maximize stability of the full system.

B. Receding-Horizon Optimal Control

In this work, the user's primary mode of interaction with the dynamic system is by defining reference trajectories for the automated controller to attempt to follow. The automated controller uses a discrete-time, receding-horizon, nonlinear optimization procedure to calculate controls to stabilize to the user-defined reference trajectory. The receding-horizon controller (RHC) is a type of Model Predictive Control [17] in which cost is minimized over a short time horizon. Our chosen methodology is based on an online projection-based trajectory optimization technique originally presented in [18]. This technique was later made interactive by adapting the RHC to discrete-time systems [19], and eventually reformulating it as a real-time, receding horizon procedure [20] that was used to experimentally stabilize the physical version of the planar crane system used in this work. Other methods are similarly capable of solving the presented optimization problem (e.g. [21], [22], [23]), however, the described approach was chosen because it provides both an optimal control and a feedback law, increasing the stability of the resulting receding horizon controller. Additionally, the trajectory optimization technique used is general such that it can be applied to different robots with relatively little effort.

In a general receding horizon control algorithm, at every timestep k a discrete-time trajectory optimization problem is solved over a reference trajectory defined over the next N timesteps. The optimization routine only has $\Delta t = t_{k+1} - t_k$ seconds to achieve convergence and produce a control signal for the present timestep, before a new measurement is taken and the procedure must begin again for the next timestep. One of the primary features of the particular algorithm utilized herein is that it produces a dynamically feasible system trajectory after every iteration. So even if Δt seconds does not provide sufficient computation time to achieve full convergence, as long as there is time to take a single step, the algorithm is still capable of producing system controls.

In this work, at timestep k the reference trajectory for the receding optimization is given by

$$\begin{aligned}\bar{\xi}_{ref,\,k} &= (x_{ref,\,k}, u_{ref,\,k}) \\ &= \left(\{x_{ref}(i)\}_{i=k}^{k+N},\, \{u_{ref}(i)\}_{i=k}^{k+N-1}\right)\end{aligned}$$

over the horizon $t_{ref,\,k} = \{t_{ref}(i) = i\Delta t \mid i = k..k+N\}$ where the reference state $x_{ref,\,k}$ and reference input $u_{ref,\,k}$ do not necessarily satisfy the system dynamics. The optimization problem statement at time k is then

$$\begin{aligned}\xi_k^* &= \underset{\xi_k \in \mathscr{T}}{\arg\min}\, J(\xi_k, k) \\ J(\xi_k) &= \sum_{i=k}^{k+N-1} l(x(i), u(i), i) + m(x(k+N))\end{aligned}$$

where ξ is used to indicate N-length sequences of both state x and input u, and $\mathscr{T}$ is the set of dynamically admissible states and input trajectories over the $t_{ref,\,k}$ time horizon. The running cost Lagrangian $l(\cdot)$ and the terminal cost $m(\cdot)$ are given by

$$\begin{aligned}l\left(x(k), u(k), k\right) =& \frac{1}{2}(x(k) - x_{ref}(k))^\intercal Q(x(k) - x_{ref}(k)) + \\ & \frac{1}{2}(u(k) - u_{ref}(k))^\intercal R(u(k) - u_{ref}(k))\end{aligned}$$

and

$$m(x(N)) = \frac{1}{2}(x(N) - x_{ref}(N))^\intercal P_1 (x(N) - x_{ref}(N))$$

where Q, R, and P_1 are positive semidefinite, symmetric weighting matrices [24].

This optimization algorithm is an iterative, indirect optimal control algorithm with simultaneous variations of state and control at each iteration. As with many iterative optimization algorithms, it is guaranteed to converge to a local minimum, but has no guarantees on finding a global optimizer. Both first and second order decent directions are found by minimizing a local quadratic model. Once a descent direction is found, a step size is found to satisfy a sufficient decrease condition, and then the scaled descent direction is added to the current iterate. When the descent direction is added to the current iterate, the state-control trajectory pair no longer satisfies the system dynamics. The projection operator maps this infeasible system trajectory back to the system's trajectory manifold while maintaining convergence guarantees.

Important to note about this framework is that for a given timestep size, Δt, we only allow the optimization to run for up to Δt seconds. Since this computation takes a finite amount of time, the horizon for the optimizations are actually one timestep ahead of real-world time. This way the optimization will have completed by the time its result is needed for sending controls to the system. As a final point, note that the reference trajectory must be defined for at least $N\Delta t$ seconds into the future. The consequence is that the system always operates with $N\Delta t$ seconds of time delay from the user-provided reference.

III. Formulation of Trust

In order to define, adapt and make use of a formal notion of trust, our proposed framework consists of two steps, *evaluation of user input* and *control modulation*. In the *evaluation of user input*, a trust metric is calculated as a function of the deviation from the reference trajectory. The *control modulation* step then uses the trust metric to allocate control authority. This approach allows us to asses and improve the system's understanding of the user's abilities.

A. Evaluation of User Input

After each interaction with the system, a trust metric is calculated using tools provided by optimal control theory. We take a control-theoretic viewpoint in which we use the *deviation* of the executed trajectory from the reference trajectory, as this measure indicates how well the receding horizon controller is able to track the user input. For a given trial i we calculate a deviation metric δ. The deviation from the reference trajectory can be captured by the Fréchet distance [25] between the executed and desired trajectories. To compute this metric in real time, we use a discrete variation of the Fréchet distance [26],

$$\delta_i(f, g) = \inf_{\alpha,\beta} \max_{t\in[0,1]} d(f(\alpha(t)), g(\beta(t))) \tag{1}$$

where $f : [a, b] \to V$ and $a, b \in \Re$, is the reference trajectory and $g : [a', b'] \to V$ and $a', b' \in \Re$, is the control trajectory and (V, d) is a metric space. α and β are continuous nondecreasing functions that map from $[0, 1]$ onto $[a, b]$ and $[a', b']$, respectively. $\inf$ is the infinum, or greatest lower bound. We use the Fréchet distance to compute the deviation between the executed and desired trajectories as this measure accounts for the velocity and ordering of points along each curve, a property not shared by similar metrics such as the Hausdorff distance [27], which computes the distance between two geometries without explicitly considering the paths as time-based trajectories. Additionally, as mentioned, the discrete measure can be computed in real time which is important both for our experiments and for any resulting interactive system.

It should be noted that while we focus on the deviation from the reference trajectory in this work, there are other control-theoretic measures of performance that could be interesting. Measures such as distance from the basin of attraction of the receding horizon controller could be incorporated into the calculation of δ.

We define the trust metric to decrease as the deviation δ increases. To calculate the trust metric τ_i at trial i, we represent the distribution of deviations as a Gaussian distribution, $\delta \sim N(\mu, \sigma^2)$, where μ and σ^2 are the mean and variance of an individual's deviation history. We then update the previous trust metric τ_{i-1} by computing the probability $\mathscr{P}$ of δ_i

$$\tau_i = \tau_{i-1} + \gamma \cdot \mathscr{P}\{\delta = \delta_i\} \tag{2}$$

where γ is a learning rate that determines how quickly the trust decays with performance, and $0.1 \leq \tau \leq 1$.

This Gaussian distribution represents the system's knowledge of the user, and is iteratively updated after each trial. Intuitively, if there is a large deviation from the desired trajectory, then τ_i should be low—because the user is providing reference trajectories that are difficult for the controller to track, which reduces the overall robustness of the system. We use a Gaussian distribution to represent the system's knowledge of the user as it provides a probabilistic method for weighting updates to the trust metric that places a larger emphasis on greater deviations and reduces the effect of smaller changes in deviation history. Additionally, as the distribution is parameterized by an individual user's history, we can recognize significant changes in the performance as either personal learning or a failure to understand the system.

B. Modulation of User Input via Trust

The proposed framework uses trust to allocate control authority. This approach is motivated by the fact that the human might be poor at accounting for the system dynamics in the low-level controls, but by using the learned trust level we can *regulate* the human's input to produce stable reference trajectories that require little effort to track.

Modulation of the user's input is realized through a combination of a low-pass filter and scaling the input speed. By removing the high-frequency content from the input signal, the receding horizon controller is better able to track the reference trajectory. Similarly, by scaling the input speed, users are better able to control for momentary mistakes that can lead

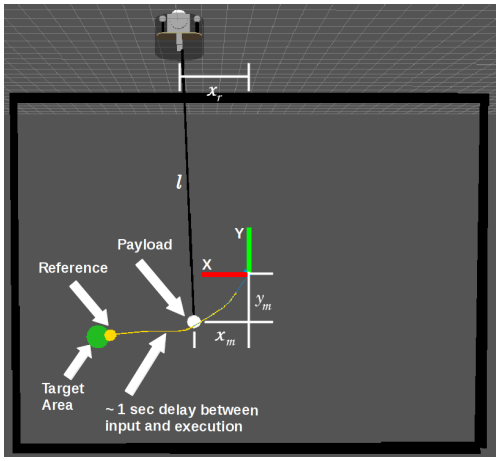

Fig. 1. Definition of the system configuration variables x_m, y_m, x_r and l. The delay between user input (yellow line) and execution (blue line) is approximately 1 second.

to challenging reference trajectories. It is important to note that these transformations can adversely affect more typical task performance measures such as time to completion, but this is a trade-off for system stability.

Trust modulates the cutoff frequency of the low-pass filter ω according to

$$\omega = (\omega_{max} - \omega_{min}) \cdot (\tau_i)^{\lambda} + \omega_{min} \tag{3}$$

where ω_{min} is the minimum cutoff frequency that still allows the user to complete the task, ω_{max} is the maximum frequency that contains control information in the input signal, λ is a parameter that shapes the steepness and direction of the function, and τ_i is the user trust from Equation (2).

Trust also influences the magnitude of the human's input, according to

$$\widetilde{v_i} = \tau_i \cdot v_i \tag{4}$$

where $\widetilde{v_i}$ is the tempered 2D system velocity and v_i is the 2D input velocity.

As trust in the operator input increases, the user is given greater command bandwidth—with the expectation that skilled users generate high frequency inputs only when appropriate and feasible for the controller to track, whereas with novice users the high frequency signals tend to be overshoot or corrective movements.

IV. Experiment Design

Our trust-based shared-autonomy framework is demonstrated on a simulated planar crane system, in which an overhead robot with a winch has a mass suspended by a string. The dynamic simulation and optimal control calculations for the planar crane system are completely done in the open-source Python module `trep`[1] [28], which has been used for a variety of real-time optimal control and estimation problems ranging in complexity from a single degree-of-freedom (DOF) pendulum up to a 40-DOF marionette [29], [30]. One of `trep`'s strengths is its integration with the Robot Operating System (ROS). `Trep`'s ability for real-time optimal control calculations, combined with ROS's ability to interface with hardware and share code, result in a software package that enables shared-control research.

The experiment was run on a Core i7 laptop with 8 GBs of RAM. The winch was initialized at the same location in each experiment (red circle in Fig. 2). The operator used the joystick of a Sony Playstation 3 (PS3) controller to provide the desired trajectory (or reference trajectory), which refers to the ordered set of target positions (yellow line in Fig. 2). As the user moves the target position (green circle in Fig. 2) through the environment, we maintain the desired position and velocity associated with each timestep.

A. Experimental System

The overhead robot is constrained to motion in a single dimension, moving only along the x-axis, and a pulley controls the length of the string. The resulting system has four configuration variables: horizontal position of the mass x_m, vertical position of the mass y_m, horizontal position of the robot x_r, and the length of the string l. The vertical position of the robot is fixed. Fig. 1 illustrates the coordinate system and configuration variables. Our model of the system assumes that the horizontal position of the robot x_r and the length of the string l can be treated as *kinematic inputs* [29], [31]. With this assumption the Lagrangian for the system is only a function of the dynamic configuration variables x_m and y_m, and it is given by

$$L(q, \dot{q}) = \frac{1}{2} m(\dot{x}_m^2 + \dot{y}_m^2) - m\, g\, y_r \tag{5}$$

where m is the mass of the payload and g is the acceleration due to gravity. A holonomic constraint enforces compatibility between the robot's kinematically-controlled horizontal position and string length and the two dynamic configuration variables. In continuous time, this results in a system with an eight-dimensional state vector defined below

$$X(t) = [x_m(t), y_m(t), x_r(t), l(t), \dot{x}_m(t), \dot{y}_m(t), \dot{x}_r(t), \dot{l}(t)]^T$$

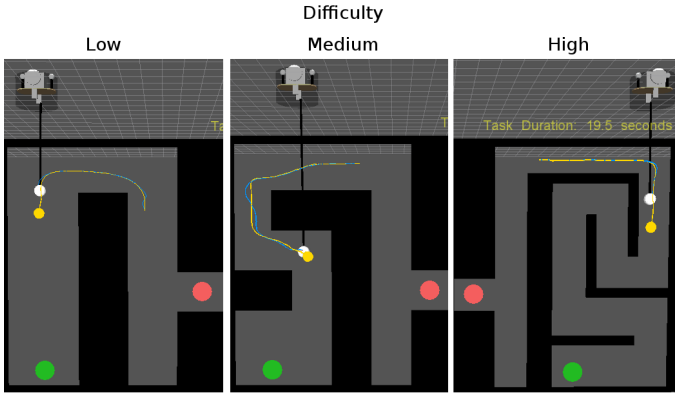

Fig. 2. Example maze environments for the simulated planar crane task, showing initial position (red circle), target position (green circle), current position of the user input (yellow circle), reference trajectory (yellow line), executed trajectory (blue line), suspended mass (white circle), and maze walls (black). A robotic winch (grey) manipulates the location and length of the string supporting the payload.

[1] `trep` is available at http://nxr.northwestern.edu/trep

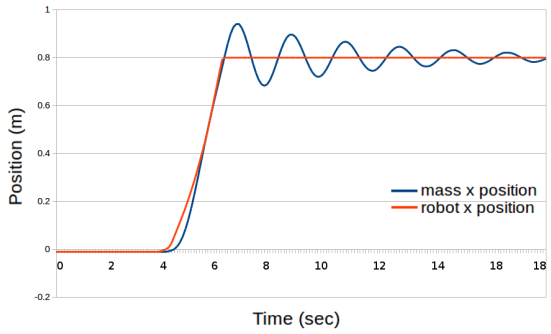

Fig. 3. Robot (red) and suspended mass (blue) position versus mass position along the x-axis during direct control. Note the pendular dynamics of the system when the user does not attempt to issue controls that account for the system dynamics.

and a two-dimensional input vector

$$U(k) = [\ddot{x}(t), \ddot{l}(t)]^T$$

comprised of accelerations of the robot position and string length. To discretize this system and obtain the discrete-time controller, we use variational integrators to represent the discrete-time system [28], [29].

We choose this system because it provides dynamics that are difficult for a human to control, while allowing for the definition of dynamic tasks that are representative of tasks for which control sharing would be beneficial. Fig. 3 illustrates what can happen when a human controls the robot position and winch speeds directly, without aid from the autonomy. In this case, the position of the mass along the x-axis oscillates considerably due to the pendulum-like dynamics of the system. Though the humans control inputs are uncomplicated (red line), the resulting dynamics are unaccounted for, as evidenced by the oscillation in the positional error of the suspended mass position over time (blue line).

B. Experimental Task

Inspired by a task that is currently part of real crane operator certification tests, we implement a maze navigation task within our simulated planar crane environment (Fig. 2). In the certification test task, the operator must negotiate a zigzag corridor with a payload [32]. Our task presents users with a target location within a maze, with walls arranged such that the path to the target is highly constrained. The task is complete once the mass dwells within the target location for 0.5 seconds.

We test three different task configurations of increasing difficulty (Fig. 2). First, a *low* difficulty configuration where the total path length is short, requires few turns ($\sim$3) and the maze hallways are wide. Then, a *medium* difficulty configuration where the total path length is longer, requires more turns ($\sim$5) and the maze hallways are an average of 60% as wide as the low difficulty configuration. Finally, a *high* difficulty configuration where the total path length is long, includes many turns ($\sim$10) and the maze hallways are an average of 50% as wide as the low difficulty configuration. A total of 21 (8 low difficulty, 8 medium difficulty and 5 high difficulty) unique mazes are used in this experiment.

C. Autonomous Control

The automated control uses a receding-horizon controller which controls x_r and l to track a reference trajectory. The human specifies the desired reference trajectory with respect to the position of the mass x_m and y_m. A full reference trajectory actually consists of defining the complete position, velocity and momentum of the system at each time step, so, our simplifying assumptions set the velocity and momentum for both the robot and the suspended mass to zero. The variables x_r and l are calculated using a simple inverse kinematic solution that does not account for mass swing, where $x_r = x_m$ and $l = y_r - y_m$. While this is not a feasible trajectory itself, the receding horizon controller does a good job of tracking the user input variables x_m and y_m.

Unlike trajectory optimization techniques which usually require the entire trajectory prior to generating the optimal trajectory, the receding horizon controller optimizes the trajectory within the window of the receding horizon. This enables online control of the system by a user. This interactivity comes at the cost of not having a global optimal trajectory, due to an inability to look ahead along the trajectory over the entire time period. Consequently, if the receding horizon window is set too small, the controller is unable to account for enough of the system dynamics and does not perform well. Alternatively, if the window is set too large, the optimization can take too long to run in real time. We found that for this system the range of window sizes that work well is between 5 and 20 time steps (approximately 0.5 to 2 seconds). For all of our experiments, we use a window size of 10, meaning there is a delay of about one second between user input and system response (Fig. 1).

We tune the optimization parameters (Q, R and P_1) such that much higher weights are placed on having the mass follow the reference configuration. That is, the diagonal terms corresponding to x_m and y_m had much higher weights than any other entries. N was chosen to be high enough as to achieve good performance, and low enough as to allow the optimization to reliably fully converge in δt seconds.[2] These weights also are the same as the tuned weights for the real-world experimental system [20].

D. Trust Computation

In order to determine the appropriate range of low-pass filter cutoff frequencies to map to trust, we performed a Fourier analysis of user input that contained high and low frequency movements. Fig. 4 shows the analysis which highlights that nearly all of the control information exists in the 0.1 to 2.5 Hz bandwidth. Thus the range of our cutoff frequency for the low-pass, 4th-order Butterworth filter is from 0.1 to 3.0 Hz, where 0.1 Hz maps to zero trust in the user input and 3.0 Hz maps to full trust.

From the Fourier analysis, we see that the control content decreases monotonically. To distribute the capabilities of the user uniformly over the trust range, we use Equation (3) to map

[2]Optimization parameter values : Q = diag($[20, 20, 0.1, 0.1, 0.1, 0.1, 0.25, 0.25]$), R = diag($[0.1, 0.1]$), P1 = Q, δt = 0.1 (10 Hz), N = 10 (with 10 timesteps, each RHC window considers a 1-second horizon).

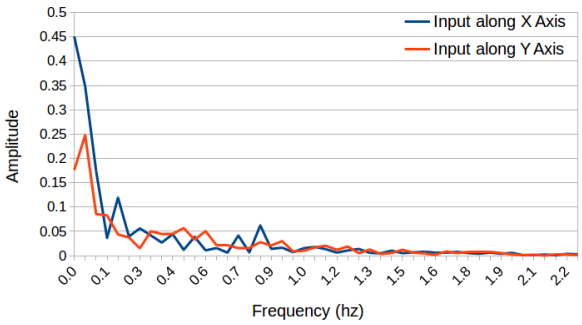

Fig. 4. Fourier analysis of the user input along the x axis (blue) and along the y axis (red). The majority of the signal content of the user input happens below 0.2 Hz. Above 2.5 Hz there is almost no signal content (not visualized).

trust to cutoff frequency ω with $\omega_{max} = 3.0$, $\omega_{min} = 0.1$, and $\lambda = 3$. The intuition behind this is to require large changes in trust for small changes in cutoff frequency when between ω_{min} and ω_{max}.

E. Experimental Protocol

Twenty-two users were recruited from the Northwestern University community.[3] Users were randomly grouped into two cohorts:

- Static: The trust level was held static after the initial training period.
- Adaptive: The trust level evolved throughout the experimental procedure.

Each experiment assumed the following protocol:

- Initialization: Three trials of shared control with fixed trust values on the low-difficulty maze task. Each trial was initialized to one of the (low 0.1, middle 0.5 or high 1.0) fixed trust values, and after each trial the trust was updated according to Equation 2. The presentation order (of which fixed trust value) was random and balanced across subjects.
- Low: Five trials of shared-control on the low-difficulty maze task. Trust was initialized in the first trial to the average of the three updated trust values from the initialization phase, for both cohorts. Trust was then held fixed for cohort *static*, and updated according to Equation 2 after each trial for cohort *adaptive*.
- Medium: Five trials of shared-control on the medium-difficulty maze task. Trust for the *adaptive* cohort was initialized to the final value updated in phase *Low*. Trust for the *static* cohort remained fixed.
- High: Five trials of shared-control on the high-difficulty maze task. Trust for the *adaptive* cohort was initialized to the final value updated in phase *Medium*. Trust for the *static* cohort remained fixed.

[3]Two participants were removed from the study before analysis due to a poor understanding of the task requirements. On average, users completed 11.7 of the 15 test trials. The two removed users completed one trial each.

Each experiment additionally included five trials of direct control on the low-difficulty maze, for a total of 23 trials.

V. Results and Discussion

The validation of the proposed formulation consists of an analysis of the system's *understanding of*, and faculty in *accounting for*, user specific abilities to provide references trajectories that can be easily tracked by the automated controller. One premise underlying this formulation of trust is that the system should be able to learn from an individual user's inputs and leverage this information to modulate the control authority afforded to that user. In our formulation, the goal of the system is to modulate the user input to produce stable references trajectories as defined by a minimization of control effort. We compute the controller effort as the magnitude of the two dimensional control vector, $u(t)$. This vector is comprised of the finite-differenced velocity of the robot's horizontal position and the finite-differenced velocity of the string length. We compute the magnitude at each time step using the Euclidean norm. The average controller effort over the course of an entire trial is defined as

$$U = \frac{\sum_{t=0}^{N} \|u(t)\|}{N} \tag{6}$$

where t is time, and N is the final time-step in a given trajectory. Therefore, larger controller effort indicates that the controller is experiencing some combination of increased error from the reference trajectory and increased input effort. Either scenario indicates that the automated controller has had to work harder to track the reference trajectory.

In this work, we evaluate the performance of the operator on a given task entirely through an analysis of the average required controller magnitude. There are other possible measures including task-specific metrics like number of collisions; however, we chose average controller magnitude as our sole performance metric as it is task-agnostic and will generalize to other applications in which a user provides a reference trajectory.

A. Adaptive vs. Static Trust

Here we analyze the system's ability to modulate a user's trust metric to produce stable reference trajectories. We perform a statistical analysis comparing the average controller magnitude, U, between the adaptive and static trust cohorts in each maze configuration. All statistical analysis is done using a two-tailed Student's t-test. In both the static ($p < 0.01$) and adaptive ($p < 0.01$) trust cohorts, we see a statistically significant decrease in the average controller magnitude required to track the user's reference trajectory in the final maze configuration when compared with the initial maze configuration. This suggests that users in both cohorts are able to learn pertinent aspects of the system dynamics and how to provide stable reference trajectories from the viewpoint of the automated controller.

We also find (Fig. 5) a statistically significant difference between the average controller magnitude, U, required to track reference trajectories provided by users in the static

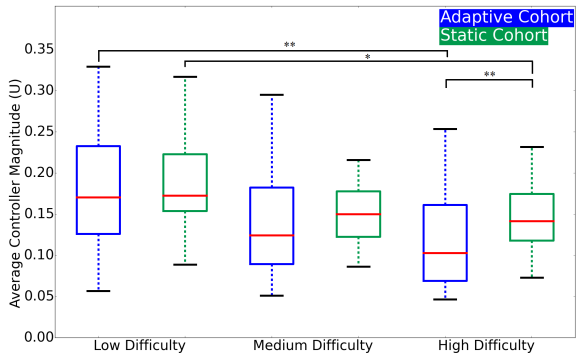

Fig. 5. Average controller magnitude (U) per maze configuration for the static (green) and adaptive (blue) trust cohorts. We see a significant ($p < 0.01$) decrease in average controller magnitude between the initial maze configuration (low difficulty) and the final maze configuration (high difficulty) in both cohorts. We also see that the adaptive cohort requires a significantly ($p < 0.01$) diminished average controller magnitude than the static cohort in the final maze configuration. We do not see this difference in either the low or medium difficulty maze configuration which demonstrates that the rate of learning is significantly accelerated in the adaptive trust cohort. Key : * $p < 0.05$ and ** $p < 0.01$.

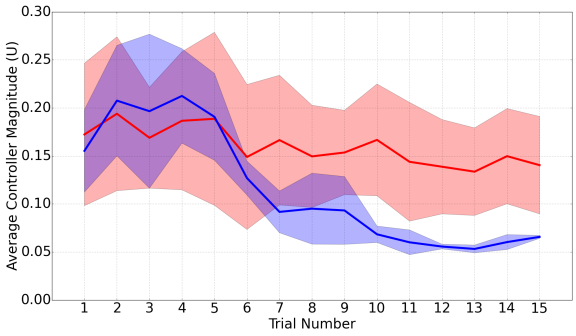

Fig. 6. Evolution of the average controller magnitude (U) per trial. The data are divided into those subjects whose final trust value was higher (red) versus lower (blue) than the initial trust value. Mean (line) and standard deviation (variance envelope) are presented. We see a significant decrease in the required average controller magnitude both in users whose final trust value was lower ($p < 0.01$) and in users whose final trust value was higher ($p < 0.05$) than their initial value. This demonstrates that the results hold regardless of whether the initial control authority allocation is an over- or under-estimate of the user's expertise.

trust cohort versus those in the adaptive trust cohort, in the final maze configuration ($p < 0.01$). As we see no statistical evidence that one cohort outperforms the other in the first two maze configurations, we can infer that the adaptive trust formulation allows the system to adapt to the (possibly changing) abilities of the user, and so modulates the user input to provide reference trajectories that require less effort for the controller to stabilize to—*regardless of the abilities of the individual user*.

B. Average Controller Magnitude in Adaptive Trust Cohort

A more detailed analysis of how trust evolves in the adaptive trust cohort can be seen in Fig. 6. This plot presents preliminary results[4] that further suggest the controller is adapting to the abilities of the user, resulting in reference trajectories that require less effort to track. This plot breaks down the evolution of required controller effort over the course of the experiment based on users who finished the study with higher trust (red) than they began with, and those who finished the study with lower trust (blue).

This plot shows that in cases where the automated system thinks the person can handle more trust, which corresponds with the user being given *greater* control bandwidth, there is a reduction in average control magnitude ($p < 0.05$). Additionally, when the system thinks the person requires less trust, which corresponds to *lower* control bandwidth, and we also see a reduction in average control magnitude ($p < 0.01$). This demonstrates that the results are not due to a specific modulation of the user input (e.g. saturation of the execution speed) as our hypothesis holds true whether the user sees an improvement or decline in their abilities to solve the maze task. That is, when the system's trust is adaptive, the operator is able to produce reference trajectories that require less effort to track than when the system's trust is static.

[4]We say preliminary results because this test included only the members of the adaptive trust cohort, and divides them into two groups (consisting of 7 (red, Fig. 6) and 3 (blue, Fig. 6) subjects), and therefore provided a smaller sample size from which we draw conclusions.

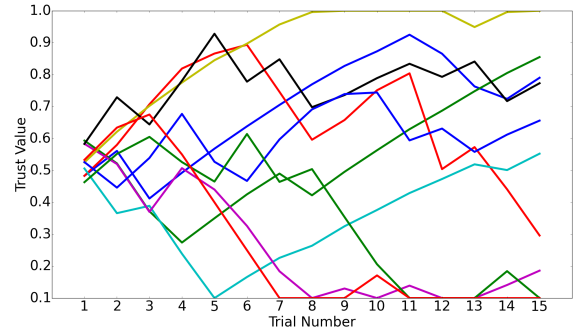

Fig. 7. Evolution of trust over the course of the experiment (15 trials). Each line represents a single user in the adaptive trust cohort. No single pattern emerges, suggesting that the adaptive trust metric based on user performance is the causal variable in reducing the required average controller magnitude of the shared control system.

C. Trust Metric Over Time

Additionally, we find no single trend in the evolution of the trust values that produce the trend of decreasing average controller magnitude (Fig. 7). This helps elucidate the point that it is not simply an increase or decrease in the trust metric that allows a user to produce superior reference trajectories. Rather, from the standpoint of the automated controller, it is a combination of user performance, system learning and the adaptive trust level, which produces this trend.

VI. Conclusion and Future Work

This work has presented a trust-based shared-control framework that utilizes a control-theoretic measure of trust that an automated controller has in the operator. Results show that an adaptive trust metric, based on our control-theoretic formulation, was able to improve the ability of the shared-control system to produce reference trajectories that require

significantly ($p < 0.01$) less effort for the controller to track than those provided by users with a static trust metric. The reduced average controller magnitude, U, reflects the system's ability to learn appropriate methods for modulating the operator's input, resulting in reference trajectories that are easier to track. This work creates a foundation upon which to expand the trust-based shared control framework to include the online, continuous adaptation of trust, more granular user skill level classification, as well as applications to additional tasks and robot platforms.

REFERENCES

[1] K. K. Tsoi, M. Mulder, and D. A. Abbink, "Balancing safety and support: Changing lanes with a haptic lane-keeping support system." in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2010.

[2] A. E. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama, and D. Gossow, "Strategies for human-in-the-loop robotic grasping," in *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2012.

[3] C. Cipriani, F. Zaccone, S. Micera, , and M. C. Carrozza, "On the shared control of an EMG-controlled prosthetic hand: Analysis of user-prosthesis interaction," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 170–184, 2008.

[4] F. Gao, M. Cummings, and E. Solovey, "Modeling teamwork in supervisory control of multiple robots," *IEEE Transactions on Human-Machine Systems*, vol. 44, no. 4, pp. 441–453, 2014.

[5] R. Chipalkatty, H. Daepp, M. Egerstedt, and W. Book, "Human-in-the-loop: Mpc for shared control of a quadruped rescue robot," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[6] J. de Winter and D. Dodou, "Preparing drivers for dangerous situations: A critical reflection on continuous shared control," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2011.

[7] R. Gonalves, S. Ferreira, J. Pinto, J. Sousa, and G. Gonalves, "Authority sharing in mixed initiative control of multiple uninhabited aerial vehicles," in *Engineering Psychology and Cognitive Ergonomics*. Springer Berlin Heidelberg, 2011, vol. 6781, pp. 530–539.

[8] G. Stein, "The practical, physical (and sometimes dangerous) consequences of control must be respected, and the underlying principles must be clearly and well taught." *IEEE Control Systems Magazine*, vol. 272, no. 1708/03, 2003.

[9] M. Oishi, "Assessing information availability for user-interfaces of shared control systems under reference tracking," in *Proceedings of the American Control Conference (ACC)*, 2014.

[10] N. Matni and M. Oishi, "Reachability-based abstraction for an aircraft landing under shared control," in *Proceedings of the American Control Conference (ACC)*, 2008.

[11] S. Au, P. Bonato, and H. Herr, "An EMG-position controlled system for an active ankle-foot prosthesis: an initial experimental study," in *IEEE International Conference on Rehabilitation Robotics (ICORR)*, 2005.

[12] I. Nourbakhsh, K. Sycara, M. Koes, M. Yong, M. Lewis, and S. Burion, "Human-robot teaming for search and rescue," *Pervasive Computing*, vol. 4, no. 1, pp. 72–79, 2005.

[13] A. Xu and G. Dudek, "OPTIMo: Online probabilistic trust inference model for asymmetric human-robot collaborations," in *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2015.

[14] C. Pippin and H. Christensen, "Trust modeling in multi-robot patrolling," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[15] M. Desai, "Modeling trust to improve human-robot interaction," Ph.D. dissertation, University of Massachusetts Lowell, 2012.

[16] A. Xu and G. Dudek, "Trust-driven interactive visual navigation for autonomous robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[17] J. Mattingley, Y. Wang, and S. Boyd, "Receding horizon control," *Control Systems, IEEE*, vol. 31, no. 3, pp. 52–65, 2011.

[18] J. Hauser, "A projection operator approach to the optimization of trajectory functionals," vol. 35, no. 1, 2002, pp. 377–382.

[19] J. Schultz, E. Johnson, and T. D. Murphey, "Trajectory optimization in discrete mechanics," in *Differential-Geometric Methods in Computational Multibody System Dynamics*. Springer International Publishing, 2015.

[20] J. Schultz and T. D. Murphey, "Real-time trajectory generation for a planar crane using discrete mechanics," in *IROS Workshop on Real-Time Motion Generation and Control*, 2014.

[21] Y. Tassa, T. Erez, and W. D. Smart, "Receding horizon differential dynamic programming," in *Proceedings of Neural Information Processing Systems (NIPS)*, 2008.

[22] H. J. Ferreau, H. G. Bock, and M. Diehl, "An online active set strategy to overcome the limitations of explicit MPC," *International Journal of Robust and Nonlinear Control*, vol. 18, no. 8, pp. 816–830, 2008.

[23] M. Diehl, H. G. Bock, and J. P. Schlöder, "A real-time iteration scheme for nonlinear optimization in optimal feedback control," *SIAM Journal on control and optimization*, vol. 43, no. 5, pp. 1714–1736, 2005.

[24] B. D. O. Anderson and J. B. Moore, *Optimal Control: Linear Quadratic Methods*. Dover Publications, 2007.

[25] H. Alt and M. Godau, "Computing the fréchet distance between two polygonal curves," *International Journal of Computational Geometry & Applications*, vol. 5, no. 1, pp. 75–91, 1995.

[26] T. Eiter and H. Mannila, "Computing discrete fréchet distance," Citeseer, Tech. Rep., 1994.

[27] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*. Springer Science & Business Media, 2009, vol. 317.

[28] E. Johnson, J. Schultz, and T. Murphey, "Structured linearization of discrete mechanical systems for analysis and optimal control," *IEEE Trans. on Automation Sci. and Eng.*, vol. 12, no. 1, pp. 140–152, 2014.

[29] E. R. Johnson and T. D. Murphey, "Scalable variational integrators for constrained mechanical systems in generalized coordinates," *IEEE Trans. on Robotics*, vol. 25, pp. 1249–1261, Oct. 2009.

[30] J. Schultz and T. D. Murphey, "Extending filter performance through structured integration," in *American Controls Conf. (ACC)*, 2014.

[31] F. Bullo and A. D. Lewis, *Geometric Control of Mechanical Systems*, ser. Texts in Applied Mathematics. New York-Heidelberg-Berlin: Springer Verlag, 2004, vol. 49.

[32] (2014) Mobile crane operator certification exam. National Commision for the Certification of Crane Operators (NCCCO). [Online]. Available: http://www.nccco.org/nccco/certification-programs/mobile-crane-operator/exam-outline