Blended Shared Control of Zermelo's Navigation Problem

Aaron Enes, *Student Member, IEEE,* and Wayne Book, *Fellow, IEEE*

***Abstract*— Many machines–from hydraulic excavators to mobile wheelchairs–are manually controlled by a human operator. In practice, the operator assumes responsibility for completing a given task at maximum utility, even though the optimal inputs may be unknown to the operator. Here we introduce a simple technique termed Blended Shared Control, whereby the human operator commands are continually merged with the commands of a robotic agent. This approach is shown to result in a lower task completion time than manual control alone when applied to a problem motivated by Zermelo's navigation problem. Experimental results are presented to compare blended shared control to other types of controllers including manual control, heads up display, and haptic feedback. Trials indicate that the shared control does in fact decrease task completion time when compared to fully manual operation.**

***Index Terms*— shared control, haptics, minimum time control.**

I. INTRODUCTION

Despite the increased capabilities of autonomous control and with the exception of a few research prototypes, many machines–from hydraulic excavators to mobile wheelchairs–are manually controlled by a human operator. The operator assumes responsibility for giving the inputs to cause the machine to complete a given task at maximum utility, for example in minimum time or with least energy consumption. However, even the optimal control solutions of simple nonlinear systems, such as when controlling the heading of a boat modeled as a particle moving at constant velocity relative to a field of linearly varying currents, may be non-intuitive or otherwise too complex for the human operator responsible for providing the control input. Consequently, the machine is operated suboptimally. To bolster the ability of the operator to achieve some degree of *near-optimality*, an electronic agent may be given authority to share control with the operator.

Shared control, teleoperation, supervisory control, manual control, and human-machine interaction are well studied areas and excellent books thoroughly address these topics [1], [2]. A distinguishing feature of each of these domains is that humans maintain some degree of authority within the control loop, as apposed to fully autonomous architectures for which an operator cedes practically all control to the robot.

In this paper, we loosely define *shared control* as a control scheme that causes the output to be influenced (e.g. either indirectly or through direct action) by a set of two or more entities, here considered to be a human agent or *operator* and an autonomous electronic agent or *robot*. The need for a human operator to share control with a semi-autonomous machine often arises from a combination of physical limitations (i.e. the operator *knows* how to perform the desired motion but is physically incapable) and cognitive limitations (i.e. the operator has a lack of understanding, has finite processing capabilities, or is occupied with tasks of higher importance). The numerous embodiments of shared control presented in literature can be categorized into one of several flavors as described in Fig. 1 and in the text below with illustrative examples of relevant academic research.

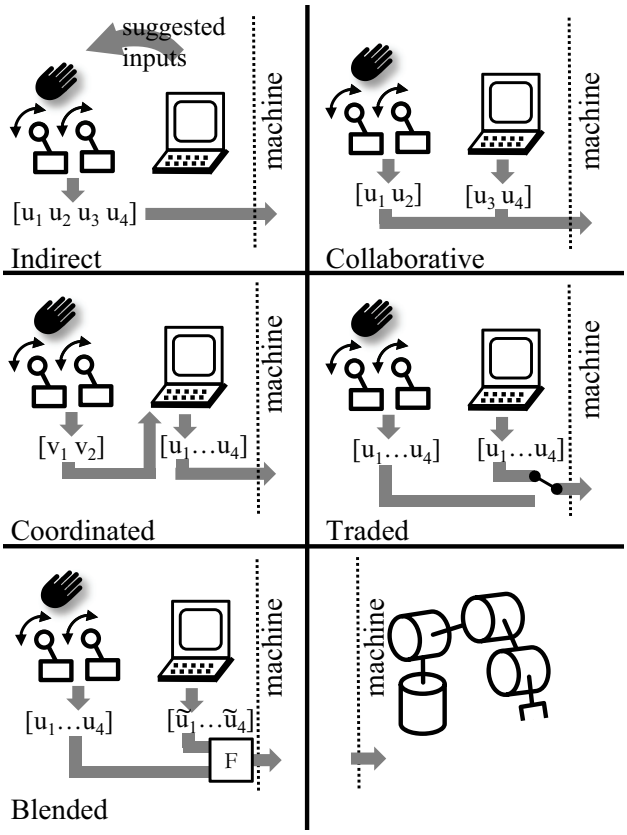


Fig. 1: Examples of human/robot interactions during various embodiments of shared control.

Indirect shared control through cues: Sensory cues are derived by the robot based on programmed criteria and displayed to the operator who nominally responds in a manner amenable to the stimulus, but the robot does not directly influence the input to the machine. Examples including visual indications of suggested process inputs in the control of power plant systems [2], haptic feedback e.g. for vehicle lane tracking [3], and improved digging performance in hydraulic excavation systems [4].

Collaborative control: A certain subset inputs are controlled by the operator while others are controlled by the robot. Examples include automobile cruise control (where

the operator controls steering while the robot controls the throttle) and automatic parking [5] (for which the operator controls the throttle while the robot controls steering).

Active constraint: The robot disallows or ignores a subspace of operator commands as a function of certain criteria such as speed, proximity to obstacles, or type of payload. For instance, a robot controller may prevent inputs which cause a wheelchair to collide with a wall while allowing all other inputs [6].

Coordinated control: Reduce the dimensionality of controlled degrees of freedom (DOF), for example by allowing an operator to control a robot's end effector position without calculating inverse kinematics or worrying about the control of individual actuators [4], [7]–[9]. This is often implemented by establishing a virtual or practical constraint such as a manifold of lower dimension than then total degrees of freedom upon which the operator inputs act. The constraint may be a mathematical formulation or a specific mapping from the input space of the operator interface device to the output space of the manipulator.

Traded control: Switching, either on demand or automatically, between fully manual and fully automatic control. Applications include aircraft autopilot systems for which the operator cedes low-level control authority during cruising yet maintains authority during takeoff and landing, and systems that allow recording a playing back of robot trajectories.

Blended shared control: operator inputs and commands calculated by the robot will simultaneously influence the response, for example in semi-autonomous wheelchair navigation [6], [10]–[12] and expert/apprentice scenarios for training in telesurgery applications [13].

The next section of this paper discusses a prosed structure for blended shared control, and presents a particular example problem developed to demonstrate the new control approach. Then, we describe an experiment used to evaluate the blended shared control approach in comparison to three alternative control methods. Finally, the results of the experiment are presented.

II. Blended shared control

Here, we discuss a proposed blended shared control architecture for a single input system, followed by experimental results for this and other types of shared control.

The approach proposed here is the blended shared control of a single input as outlined in Fig. 2. This architecture consists of a (human) operator, a three-module robotic controller, and a controlled machine. The operator issues input command U_0 via a human interface device such as a joystick, and perceives the machine response Y through sensory feedback. A high-level robotic controller modifies the original operator command through some general functional relationship with δ. Here the functional relationship is a simple summation $U = U_0 + \delta$. The command perturbation is calculated by the *blended shared control* module and may be a function of several terms, including the optimal input $\tilde{U}$ as calculated by the *optimization* module, the original input command U_0, and machine response Y. The optimized command $\tilde{U}$ is

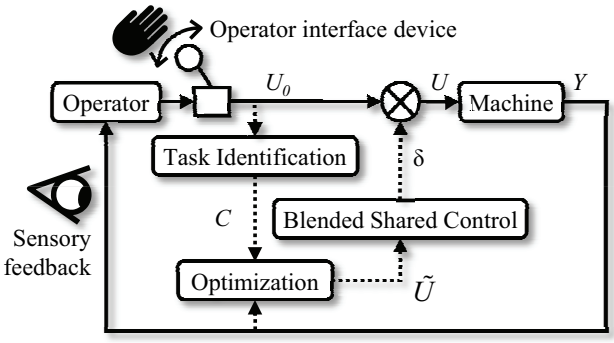


Fig. 2: The proposed architecture for *blended shared control* consists of an operator, a three-module robotic controller, and the controlled machine.

determined by dynamic models of the machine, the feedback Y, and of subtle importance a set of data C consisting of constraints and objective functions which are specific to the particular task being completed by the operator. The constraints and objective function are determined by the *task identification* module of the robotic controller.

There are several areas in this process that enable a positive synergy between the robot and human operator, as there are certain capabilities of a human operator: e.g. reasoning, safety awareness, robustness, "ideal" cost function; and certain complimentary attributes of a robotic controller: e.g. incorporation of complex system models, numerical capacity to solve those models, storage of much expert knowledge. These synergies of the blended shared control will nominally be leveraged to increase utility of the overall process. However, there are several stages in this process which may result in dis-utility and hence must be studied. Such unresolved issues include the effects of conflicting objectives between the controller and the robot (e.g. one agent values minimum time while the other wishes minimum energy), and under which conditions can it be shown that the modified machine command is less costly then the original command (this deals with the *convexity* of the problem). As a first approach in illustrating this process, in the next section we present a formulation of a single-input example.

A. Shared Control Scheme

In this section we formulate the blended shared control law for a system with a single control input. The difference of the operator's command and the optimal command as calculated by the robot agent is

$$\Delta = \theta_0 - \tilde{\theta} \tag{1}$$

where the operator input is θ_0 and the optimal command is $\tilde{\theta}$. The optimization as calculated by the shared controller depends on the machine models and a cost function internal to the robot. A *command perturbation* δ calculated by the shared controller is added to the operator command to give

$$\theta = \theta_0 + \delta$$

where θ is the control input to the machine. Designing the command perturbation is a major subject of the forthcoming research into blended shared control. In the case of a pursuit or interception problem, for example, the perturbation may be a function of any number of terms including an operator setpoint, distance to target, time on target, or Δ. For example, choosing $\delta = -e\Delta$ gives

$$\theta = \theta_0 - e\Delta \tag{2}$$

with the *blended shared control parameter* $e \in [0, 1]$. Note, when $e = 0$ the system is under manual control (i.e. $\theta = \theta_0$) and when $e = 1$ the system is fully autonomous (i.e. $\theta = \tilde{\theta}$). Varying e on the interval $[0, 1]$ thus gives a contiuum between full automation and full manual control.

III. Zermelo's Problem: Time-optimal navigation

A classic optimal control problem known as Zermelo's problem is useful for studying the proposed shared control law because of its known closed-form solution [14]. Further, the task can be easily defined and explained to a human operator: minimize the transit time to the origin.

In Zermelo's problem a ship (modeled as a particle) travels with constant speed V relative to the water while navigating a region of strong currents. The captain must control the ship's heading θ to minimize travel time to the origin. The equations describing the optimal path for the case of linearly varying current velocity are [14]

$$\begin{aligned} \dot{x} &= V\cos\theta + u(y) \\ \dot{y} &= V\sin\theta \end{aligned} \tag{3}$$

and

$$\dot{\theta} = -\cos^2\theta \frac{du}{dy} \tag{4}$$

where θ is the ship's heading measured from the x-axis, (x, y) are its coordinates, and $u = Vy/h$ is the velocity of the current. The initial value of θ is chosen so that the path passes through the origin. For the linearly varying current strength considered here, the optimal steering angle can be related to the ship position through a system of implicit feedback equations [14]

$$\begin{aligned} \frac{y}{h} &= \sec\theta_f - \sec\theta \\ \frac{x}{h} &= \frac{1}{2}\left[\sec\theta_f\left(\tan\theta_f - \tan\theta\right) - \tan\theta\left(\sec\theta_f - \sec\theta\right)\right] \\ &\quad + \frac{1}{2}\log\frac{\tan\theta_f + \sec\theta_f}{\tan\theta + \sec\theta}. \end{aligned} \tag{5}$$

Solutions to the above equations are plotted in Fig. 3. The blended shared control of Zermelo's problem is achieved by using the single input control law in (2). The control designer has freedom in selecting the particular form of e; suppose the shared control parameter e selected to be

$$e = \max(0, 1 - \frac{d}{d_0}) \cdot \max(0, 1 - (\Delta/\Delta_0)^2). \tag{6}$$

Fig. 4 shows plots of e for the parabolic form (6). This particular form in (6) allows manual operation if the ship

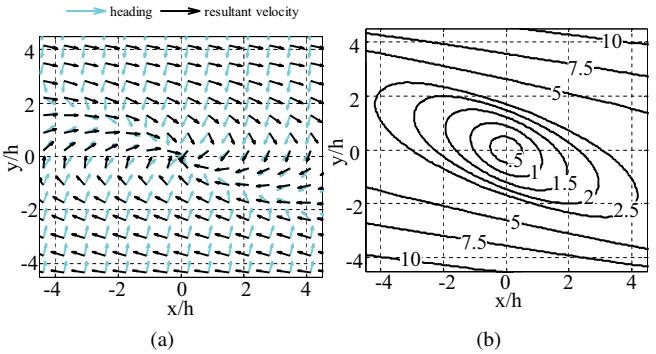


Fig. 3: (a) Directions of optimal heading (gray arrows) and resultant ship velocity (dark arrows) and (b) the optimal time to reach the origin for each location in units of h/V. Graphs shown for $h/V = 4$.

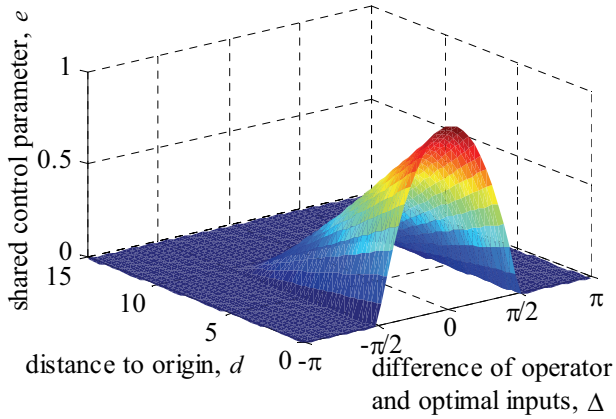
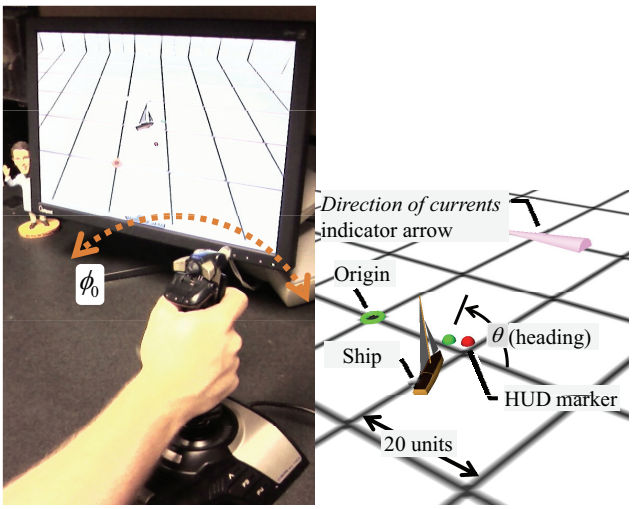


Fig. 4: The shared control parameter e in parabolic form of (6) depends on the distance to origin d and the difference of the operator and optimal inputs Δ. Shown with $\Delta_0 = \frac{\pi}{2}$ and $d_0 = 10$

is greater than distance d_0 from the goal or if the input command deviates from the optimal by greater than Δ_0. The blended shared control relinquishes control authority to the operator in the presence of large "errors" between the operator input and the robot's optimal input. The operator (rather than a complicated automatic controller) provides for the robustness and corrective action of the system in these cases. This is a first attempt at increasing robustness by resolving the conflict that may arise between the shared controller and the operator; such conflict may stem from inaccurate models plant or environment models, dissimilar cost functions used, or different goals altogether between the operator and robot.

IV. The Shared Control Evaluation

Here we describe the experimental setup for evaluating the blended shared control. An operator views a monitor (Fig. 5) depicting a ship moving in a simple virtual reality (VR) environment with dynamics governed by (3). A green



(a) (b)

Fig. 5: (a) The operator video display and joystick. (b) Close-up of operator display showing ship, origin (green ring), heading indicator (green dot), optimal steering heading shown in HUD mode (red dot), and nominal direction of currents (red arrow).

ring represents the origin to which the operator is instructed to navigate as quickly as possible. A green sphere is drawn in front of the ship to represent the present heading θ. The operator is not aware of the specific nature of the currents, but an arrow drawn on the VR display illustrates the *direction* of the flow.

The operator displaces the joystick an angle ϕ_0 to command a ship heading θ_0 through the relation

$$\theta_0 = \alpha \int \phi_0 dt$$

where α is a constant for tuning the *snappiness* of the ship response to changes in operator input. A small deadzone on the joystick input angle ϕ_0 is applied in software to prevent unintentional drift of the ship's heading.

A. Description of control types evaluated

The five types of control methods studied in this experiment are summarized next.

Manual control (MC): Implemented by setting $e = 0$ in (2), thus the operator is in full control of the ship heading giving $\theta = \theta_0$. No cues are displayed to the operator, besides the standard VR interface. This control is used as a baseline for determining operator performance in the abscence of supplementary information or aiding controls.

Heads up display (HUD): In this case, the operator has manual control of the ship ($e = 0$ so $\theta = \theta_0$). A red dot (as in Fig. 5b) is displayed to represent the optimal ship heading. The operator is instructed before the experiment to align the green heading indicator dot with the red HUD marker dot. This control is used as a baseline for determining the maximum operator capabilities, that is, the capability the operator would have *if* the optimal solution was known to the operator. The HUD is a form of indirect shared control, in the sense defined in the Introduction.

Haptic feedback (Haptic): This is a second type of indirect shared control. The operator has manual control of the ship and a Saitek Cyborg EVO Force joystick displays a restoring force $F = -\min(|\Delta/\Delta_{\max}|, 1) F_{\max} \mathrm{sgn}(\Delta)$. This resulting force will *push* the operator's hand in a direction that causes θ to approach $\tilde{\theta}$. For example, if $\Delta \leq 0$ then the joystick applies a force to the *right*, thus cueing the operator to decrease angle ϕ_0. The particular values used were $\Delta_{\max} = \pi/2$, $F_{\max} = 2.1$ newtons (measured at the joystick palm grip). The haptic feedback is motivated by the master/apprentice shared control techniques proposed for surgery training [13], and was also chosen to compare to the experiments in [3], [15] where the operator chooses how to respond to haptic cues on a steering wheel.

Shared control, heading (SC2): The operator and robot share control of the ship heading through the relation $\theta = \theta_0 - e\Delta$, with $e = \max(0, 1 - \frac{d}{d_0}) \cdot \max(0, 1 - (\Delta/\Delta_c)^2)$ as in Fig. 4. No additional cues are displayed to the operator. The particular values during the experiment were $d_0 = 25$, $\Delta_0 = 3\pi/4$.

Shared control, rate (SCJS): Here, the original operator joystick input angle ϕ_0 is modified by the shared controller to give an effective joystick input angle of $\phi = \phi_0 + \delta_\phi \cdot u(\delta_\phi, \phi_0)$, where

$$\delta_\phi = \begin{cases} -\phi_0/2 & \text{for } |\Delta| \leq \theta_{th}\,, \\ -k \cdot \mathrm{sgn}(\Delta) & \text{otherwise}\,. \end{cases}$$

and

$$u(x,y) = \begin{cases} 1 & \text{if } \mathrm{sgn}(x) = \mathrm{sgn}(y)\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

Hence, the operator and robot share control of the *rate* at which the ship's commanded heading changes. No additional cues are displayed to the operator. ($\theta_{th} = \pi/12$, $k = 0.5$).

The difference between SC2 and SCJS is subtle: in SC2 the operator's intended ship heading θ_0 is perturbed by the shared controller, whereas in SCJS the intended joystick *angle* ϕ is perturbed.

B. Effect of shared control on minimum time-to-go

Let $T(\mathbf{x})$ be the *minimum time-to-go* at the location $\mathbf{x} = [x, y]^T$, that is, the time remaining before reaching the origin assuming the ship starts at $\mathbf{x}$ and follows the time-optimal path. It can be shown that

$$T(\mathbf{x}) = h/V(tan\theta(\mathbf{x}) - tan\theta_f(\mathbf{x}))$$

where θ and θ_f are implicit functions of $\mathbf{x}$ from (5). Consider the case $\theta = \theta_0 = \tilde{\theta} + \Delta$ so the ship heading is controlled by the operator at location $\mathbf{x}$ and for a length of time dt. Then after time dt the minimum time-to-go will be

$$T_{MC} = T(\mathbf{x} + f(\mathbf{x}, \tilde{\theta} + \Delta)dt)$$

where $f(\mathbf{x}, \theta)$ is the vector form of equations of motion (3) and Δ is as defined in (1). If, on the other hand, the heading

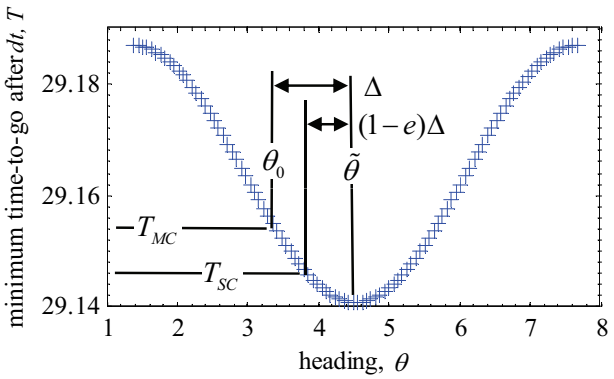


Fig. 6: The minimum time-to-go, $T(x + f(x, \theta)dt)$, plotted for $\mathbf{x} = [10, 12]^T$ and $dt = 0.01$. The minimum time-to-go using shared control never exceeds the time with manual control.

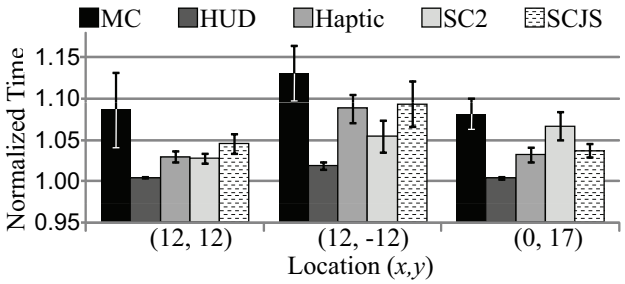


Fig. 7: Summary of completion times from each location to the origin, for each type of controller. Values are normalized with respect to the optimal time to origin from each location, then averaged among all operators for each controller. Error bars denote 95% confidence interval. For each controller, at each location, *N=24*. The optimal times to origin are 14.97 s, 7.43 s, and 18.47 s respectively for $(12, 12)$, $(12, -12)$, and $(0, 17)$. For all cases, $h = 4$ and $V = 2$.

is under shared control as in (2) then the minimum time-to-go after time dt can be written as

$$T_{SC} = T(\mathbf{x} + f(\mathbf{x}, \tilde{\theta} + (1 - e)\,\Delta)dt).$$

While not shown here for succinctness, the function $T(\mathbf{x} + f(\mathbf{x}, \theta)dt)$ is convex in the variable θ (holding all other variables constant) for all $dt > 0$. Thus, for $e \in [0, 1]$ and any $\mathbf{x}$

$$T_{SC} \leq T_{MC}$$

as illustrated in Fig. 6. Thus, for a *certain* operator input θ at $\mathbf{x}$, the minimum time-to-go with blended shared control will never be strictly worse then that with manual control. Obviously, for other systems where the cost function is *not* convex, the blended shared control may result in greater cost than manual control alone. Also, it is not clear whether or not the operator input a particular operator input at $\mathbf{x}$ is independent of the type of control which is active, so we assumed that the operator is agnostic to the . Finding efficient ways to settle these issues is the subject of ongoing research.

C. Experimental procedure

Before the experiment begins the operator is allowed five practice runs starting from various locations in the field. During the practice runs, the HUD type control is active, giving the operator a sense of how an expert would navigate the currents. Each of the experimental trials begin with the ship at one of three locations: $(12, 12)$, $(12, -12)$, and $(0, 17)$. The constants are $h = 4$ and $V = 2$. The operator triggers a *start* button on the joystick and the simulation proceeds in real time with one of the five control laws active. The trial concludes after the operator navigates within $d = 1.5$ of the origin. A trial can be prematurely stopped and re-initialized under two circumstances: the operator pushes a *stop* button on the joystick or the ship exceeds a distance of $d = 60$ from the origin. Only two resets are allowed per subject. At no time during the experiment is the operator explicitly informed which of the possible control laws is active. The starting locations and controller orders are randomized to help null learning effects, and each operator visits each location exactly three times for *each* controller, totaling 45 trials per participant.

V. Results

Eight computer literate participants volunteered for the experiment. Results summarizing the performance of all operators are summarized in Fig. 7. The mean HUD controller times were very consistent and only marginally exceeded the optimal time, as expected, presumably because the tracking skill of the operators was more than sufficient to follow the displayed optimal command. Much more variation is present in the times of the other controllers; however, both of the blended shared control approaches generally surpassed the performance under manual control.

A fair criticism of blended shared control is that the operator (who, incidentally, may be used to a particular machine *feel*) cedes *too much* authority to the robot, as both the robot and the operator will simultaneously affect the machine response. This may at best lead to a benign sense that the machine is not responding in a manner consistent with operator expectations, and at worst lead to the machine failing to respond to an operator's evasive maneuvers in the name of safety.

To test for loss of control in this single-input example, four additional trials (two with MC, two with SC2) with each operator are performed starting from $(15, 0)$ with a barrier intentionally placed to occlude the optimal path as in Fig. 8; hence the shared controller tries to cause the operator to hit the barrier while the operator is instructed to miss it . The operator performance with barriers present was 19.1 s and 19.8 s, respectively for MC and SC2; however, the data lacked sufficient statistical significant to clearly deem one approach superior to the other. Traversing the optimal path from $(15, 0)$ to the origin in absence of the barrier takes 16.0 s; but the optimal path which avoids the barrier was not calculated. More significant was the fact that under both

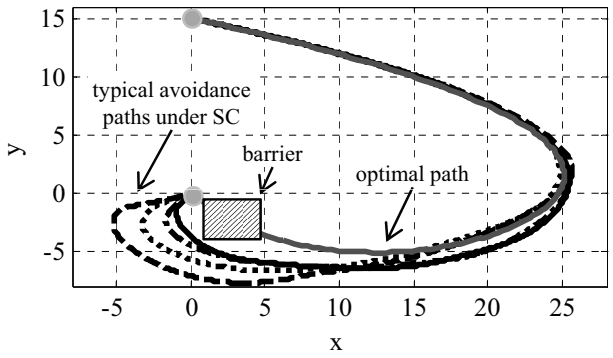


Fig. 8: A *barrier* is placed to intersect the optimal path to test if blended shared control causes operators to *lose control*. The operator is instructed to navigate to the origin in minimum time without colliding with the barrier.

controllers, only two trials among *all* experimental subjects resulted in collision with the barrier. No operator affirmed a feeling of *loss of control* when queried about navigating around a barrier, indicating that the shared controller is transparent, even when the operator commands motions that are not optimal.

VI. Conclusion

A proposed structure for blended shared control of a system with a single input was presented. We investigated the blended shared control for a class of problems with a well-defined task and a closed-form optimal solution which was globally convex in the shared control variable. Further, the operator and robot agent had equivalent cost functions. For this class of problems, initial experimental evidence indicates that the blended shared control approach is superior to purely manual control in the minimum-time problem considered here.

There are several issues to be studied before applying this approach to more complicated problems, with the most forthcoming issue being the extension to multi-input cases with higher dimensional cost functions. In this case, the convexity of the problem becomes critical, as we must ensure that the perturbed command is not *more* costly than the original. Hence, efficient methods to verify the (at least local) convexity of practical multi-dimensional optimization problem are needed. Another *softer* unresolved issue involves the effects of conflicting objectives arising, for example, from task mis-identified or inaccurate estimations of the optimal solution.

VII. Acknowledgments

This work was partially supported in part by the National Science Foundation, contract EEC-0540834.

References

[1] E. C. Poulton, *Tracking skill and manual control*. New York: Academic Press, Inc., 1974.

[2] T. B. Sheridan, *Telerobotics, automation, and human supervisory control*. Cambridge, MA: The MIT Press, 1992.

[3] P. G. Griffiths and R. B. Gillespie, "Sharing control between humans and automation using haptic interface: primary and secondary task performance benefits," *Human Factors*, vol. 47, no. 3, pp. 574–90, 2005.

[4] M. E. Kontz and W. J. Book, "Flow control for coordinated motion and haptic feedback," *International Journal of Fluid Power*, vol. 8, no. 3, pp. 13–23, 2007.

[5] W. T. Good and J. E. Vilhauer Jr, "Automatic parallel parking system," U.S. Patent United States Patent 4 735 274, 1988.

[6] G. Pires and U. Nunes, "A wheelchair steered through voice commands and assisted by a reactive fuzzy-logic controller," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 34, no. 3, pp. 301–14, 2002.

[7] H. Araya and M. Kagoshima, "Semi-automatic control system for hydraulic shovel," *Automation in Construction*, vol. 10, no. 4, pp. 477–486, 2001.

[8] P. Lawrence, S. Salcudean, N. Sepehri, D. Chan, S. Bachmann, N. Parker, M. Zhu, and R. Frenette, "Coordinated and force-feedback control of hydraulic excavators," in *International Symposium on Experimental Robotics (ISER 95)*, Stanford, CA, 1995.

[9] M. Haga, W. Hiroshi, and K. Fujishima, "Digging control system for hydraulic excavator," *Mechatronics*, vol. 11, no. 6, pp. 665–676, 2001.

[10] R. A. Cooper, "Intelligent control of power wheelchairs," *Engineering in Medicine and Biology Magazine, IEEE*, vol. 14, no. 4, pp. 423–431, 1995.

[11] B. Fernandez-Espejo, A. Poncela, C. Urdiales, and F. Sandoval, "Collaborative emergent navigation based on biometric weighted shared control," in *Proceedings 9th International Work-Conference on Artificial Neural Networks, Computational and Ambient Intelligence (IWANN 2007)*, San Sebastian, Spain, 2007, pp. 814–21.

[12] C. Urdiales, A. Poncela, I. Sanchez-Tato, F. Galluppi, M. Olivetti, and F. Sandoval, "Efficiency based reactive shared control for collaborative human/robot navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007)*, San Diego, CA, 2007, pp. 3586–3591.

[13] S. S. Nudehi, R. Mukherjee, and M. Ghodoussi, "A shared-control approach to haptic interface design for minimally invasive telesurgical training," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 588–592, 2005.

[14] A. E. Bryson and Y. Ho, *Applied Optimal Control*. Watham, MA: Blaisdell Pub. Co., 1969.

[15] P. Griffiths and R. B. Gillespie, "Shared control between human and machine: haptic display of automation during manual control of vehicle heading," in *12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS 04)*, Chicago, IL, 2004, pp. 358–366.