Highly Parallelized Data-driven MPC for Minimal Intervention Shared Control

Alexander Broad*‡, Todd Murphey†, Brenna Argall*†‡
*Department of Computer Science
†Department of Mechanical Engineering
Northwestern University, Evanston, IL 60208
Email: alex.broad@u.northwestern.edu
‡Shirley Ryan AbilityLab, Chicago, IL 60611

***Abstract*—We present a shared control paradigm that improves a user's ability to operate complex, dynamic systems in potentially dangerous environments without *a priori* knowledge of the user's objective. In this paradigm, the role of the autonomous partner is to improve the general safety of the system without constraining the user's ability to achieve unspecified behaviors. Our approach relies on a data-driven, model-based representation of the joint human-machine system to evaluate, in parallel, a significant number of potential inputs that the user may wish to provide. These samples are used to (1) predict the safety of the system over a receding horizon, and (2) minimize the influence of the autonomous partner. The resulting shared control algorithm maximizes the authority allocated to the human partner to improve their sense of agency, while improving safety. We evaluate the efficacy of our shared control algorithm with a human subjects study (n=20) conducted in two simulated environments: a balance bot and a race car. During the experiment, users are free to operate each system however they would like (i.e., there is no specified task) and are only asked to try to avoid unsafe regions of the state space. Using modern computational resources (i.e., GPUs) our approach is able to consider more than 10,000 potential trajectories at each time step in a control loop running at 100Hz for the balance bot and 60Hz for the race car. The results of the study show that our shared control paradigm improves system safety without knowledge of the user's goal, while maintaining high-levels of user satisfaction and low-levels of frustration. Our code is available online at https://github.com/asbroad/mpmi_shared_control.**

I. Introduction

Shared control is a paradigm that incorporates an autonomous partner into the control loop of a robotic system to help a human partner achieve tasks they would otherwise be unable to on their own [1]. This approach offers an alternative to fully autonomous robotic systems, and can be used to extend the efficacy of modern robots in human-oriented domains (e.g., surgery, assistance and rehabilitation, search and rescue, etc...) through collaboration. In these domains the two partners often need to communicate frequently and may even be co-located. The most important consideration of the autonomous partner is therefore the safety of the joint system, as unsafe behavior can lead to injury to the human operator. However, there are other features that may be equally important when we consider both the range of behaviors the human partner may wish to perform, and the user's acceptance of the assistance provided the autonomous partner.

In the majority of related work, the decision of which partner should be in control at a given time is made based on (1) the safety of the system and (2) an evaluation of who would provide better input to achieve a perceived, or known, task goal [26]. These task-specific systems can frustrate the human partner when the user's intended goal is difficult to predict, and they fail out-right when there is no desired trajectory, task or goal the user is trying to achieve. Consider, for example, a person operating a lower-limb exoskeleton for rehabilitation purposes. In this scenario, task-level and performance-based metrics (e.g., the amount of area covered during a search mission), are not important in determining which partner should be in control at a given time, as there is often no explicit notion of a goal (i.e., the human operator may simply want to wander around aimlessly). Instead, the person's instantaneous desires are often the most relevant feature, conditioned on the general safety of the system.

The question we consider in this work then is, how does one define safety constraints that can enhance a user's ability to operate complex, dynamic machines without artificially constraining their capacity to achieve unspecified behaviors? We address this gap in the literature with a task-agnostic control allocation strategy that balances the human's desires to achieve a wide range of possible behaviors, while simultaneously improving the safety of the joint system. Our shared control paradigm therefore adheres to the following three ideals:

1) safety is paramount,
2) the user has no explicit task goal, and
3) the autonomy should exert as little influence as possible.

In other words, the goal of our shared control paradigm is to allow the user to do whatever they would like, so long as the safety of the joint system is satisfied. Additionally, when the autonomous partner does intervene, it should only minimally modify the user's input (a.k.a., the minimal intervention principal [3]). By adhering to these ideals, we hope to increase the influence of the human partner and consequently improve their acceptance of the assistance provided by the autonomy.

In this work, we develop a shared control algorithm that uses a highly parallelizable sampling-based model predictive control (MPC) algorithm to generate the autonomous partner's policy. By sampling densely at uniform over the input space,

we can evaluate a large set of potential actions that the user may wish to take, without *a priori* knowledge of a specific goal. We then use ideas from model-based reinforcement learning and model predictive control to generate predicted trajectories (or imagined rollouts) that represent the configuration (and safety) of the robot over a receding horizon. Conditioned on the predicted safety of the robotic system, we iteratively select the sampled action that most closely matches the human partner's input, allowing the user to more safely move around the environment without adhering to a single objective. Our approach relies on a representation of the human-machine system that is valid both with, and without, a known analytical model. We focus on the latter case and therefore learn a model of the joint system offline from data. We evaluate the efficacy of our approach with a human subjects study in two simulated environments. Additionally, we provide an open-source, scalable implementation of our algorithm in both environments that uses a GPU for real-time interaction.

The main contributions of this work are therefore:

- A highly parallelizable sampling-based model predictive control algorithm for autonomous policy generation.
- A predictive notion of safety that can evaluate the impact of a current action over a receding horizon.
- A human-motivated cost function that only considers the instantaneous desires of the human-in-the-loop and requires no knowledge of an explicit goal.
- Results of a human subjects study to evaluate the efficacy of, and user experience with, our shared control paradigm.
- A GPU-implementation for real-time control of two simulated systems.

In Section II we provide background information and related work. In Section III we detail the theoretically exact solution to our problem. In Section IV we describe our approximation, why it scales well to the majority of devices in our target domain (human-centered robotics), and provide analytical bounds on the sub-optimality of the applied control with respect to the human's desired motion. In Section V we describe the experiment we use to validate our shared control paradigm through a human subjects study consisting of 20 participants in two simulated environments. We also detail pertinent metrics to analyze the human partner's control skill and style, which are easily computable due to our sampling-based approach. In Section VI we present the study results which we discuss in Section VII. We conclude in Section VIII.

II. Background and Related Work

In this section we discuss background and related literature in both shared and fully autonomous control, with a particular focus on human-oriented domains.

A. Shared Control

The majority of the shared control literature focuses on assisting a human operator when a desired task is known *a priori* [20, 11] or predicted based on a model of the operator's intent [13]. For this reason, task success is often the primary metric of concern in analyzing shared control systems, while the user's desires relating to *how* a motion is achieved are often disregarded. Related literature that addresses this same limitation in shared control includes techniques that rely on model-free control policies [27], POMDPs [18], and control barrier functions [8]. In some application domains there is a welcome trade-off between achieving the desired high-level goal and intervention from the autonomous partner (e.g., with ground vehicles on a roadway where a large amount of structure is enforced on the motion of the dynamic system). In more human-centered domains, such as assistive and rehabilitation robotics, there is often significantly less structure imposed on the motion of the machine and there may be no explicit goal. For this reason, the same trade-off in task success and autonomous intervention is not consistently accepted by users [14]. In these domains, it is instead of utmost importance that the user retains a sense of personal agency and a feeling of control over the mechanical device.

Despite these differences, the most closely related work to our own (from a methodological standpoint) can be found in the semi-autonomous vehicle literature. The semi-autonomous vehicle paradigm is distinct from the concept of a fully autonomous self-driving cars as the autonomy's goal is not to take full control of the vehicle, but instead to act as a guardian or intelligent co-pilot [4], intervening on the person's control when deemed necessary to ensure safety. For this reason, there is a growing line of work that follows the minimum intervention principle in the semi-autonomous vehicle domain [25]. For example, Schwarting et al. [28, 29] describe a parallel autonomy framework that develops control trajectories for semi-autonomous vehicles that minimize deviation from user-input and achieve task-specific metrics like road following and contour tracking. Anderson et al. [3, 5] describe a geometric, homotopy-based algorithm for computing *free space* in the environment. The human operator is then allocated full control of the dynamic system so long as their input will not violate the constraints defined by the geometric constructs. In this work we provide an alternative method (prediction) of computing safe space in the environment that is simpler (e.g., there is no need to integrate constraints defined by potentially complex geometries into the optimization problem), and acts over the entire control space (i.e., instead of only the steering angle [3, 11, 15]). Additionally, all prior work in this area assumes *a priori* knowledge of the system dynamics whereas our technique extends to models learned from data.

B. Sampling-based and Stochastic Optimal Control

From a control-theoretic standpoint, related work includes shared control algorithms that build on ideas from sampling-based optimal control. For example, Carlson et al. have explored the idea of controlling a wheelchair using *safe mini-trajectories* [11, 10], however, this work again relies on *a priori* knowledge of the human operator's goal (or predicted goals based on sensor information). Relatedly, Shia et al. [30] use a probabilistic model of the user's potential future inputs to achieve a pre-specified task, instead of allowing the operator the freedom to move however they would like at each instant.

Our approach is also related to fully autonomous control solutions that rely on sampling-based and stochastic optimal control methods. For example, Lavalle et al. propose Rapidly-expanding Random Trees (RRTs) [24], which develop random trajectories through the state space that are achievable as they are constrained by the system dynamics. More recently, Kousik et al. extend this idea through a model predictive control algorithm that is based on the notion of a forward reachability set [22, 23] to ensure safety. Finally, Williams et al. have proposed Model Predictive Path Integral (MPPI) [32, 36] control as a method of solving the optimization problem through path integrals. These ideas build on similar theory to our own (approximating an optimal solution from a nominally infinite set of trajectories), however, they are again all standardly defined in relation to a specified start and goal configuration. We instead consider an approximation to the infinite set of trajectories that stem from a single point and extend in *all* directions for a given time-horizon.

III. Safe Minimal Intervention Shared Control

In this section, we describe a theoretically correct (though computationally infeasible) solution to the problem of safe minimal intervention shared control. We begin by defining the shared control problem mathematically. This problem can be posed in standard optimal control terms with an additional constraint to incorporate information from both partners. That is, our goal is to find an optimal action sequence $\bar{\mathbf{u}}$, and corresponding state sequence $\bar{\mathbf{x}}$, that minimizes the cost

$$\begin{aligned} \underset{J}{\text{minimize}} \quad & J(x(t),u(t)) = \int_{t=0}^{T} l(x(t),u(t)) + l_T(x(t)) \\ \text{subject to} \quad & \dot{x}(t) = f(x(t),u(t)), \\ & u(t) = g(u_h(t),u_a(t)) \end{aligned} \tag{1}$$

where $x(t)$ and $u(t)$ are the state and control trajectories, and l and l_T are the running and terminal costs. The optimization is subject to constraints f representing the nonlinear system dynamics, and g representing the control allocation between the human (u_h) and autonomous partners (u_a).

A. Safe Control and Inevitable Collision States

The primary constraint in g relates to the safety of the human-machine system. That is, the autonomous partner should only produce trajectories that remain safe over a receding horizon. This requires ensuring that the system does not enter an Inevitable Collision State (ICS) [6, 17]. ICSs refer to configurations from which it is impossible to safely recover, regardless of the control trajectory taken. So long as the system does not enter an ICS, it is possible to develop a control strategy that results in continued safe interaction.

B. Minimum Intervention Principle

The second feature we embed in g is known as the minimal intervention principle (MIP) which states that "an autonomous partner should only augment the human partner's control by the minimum amount necessary to achieve the desired result" [3, 25, 28]. By developing a shared control algorithm that adheres to the MIP we maximize the influence of the human partner's control at each given moment.

Algorithm 1 Safe Minimal Intervention Shared Control

1: **procedure** SMI-SC(x_t, u_h)
2: $\quad\Gamma \leftarrow$ all possible trajectories (γ) from $x(t)$
3: $\quad\Gamma_{safe} \leftarrow \gamma \in \Gamma$ where $\gamma \notin ICS$
4: $\quad u_r \leftarrow \text{argmin}(u_h, cost(\Gamma_{safe}))$
5: $\quad$**return** u_r
6: **end procedure**

C. Minimal Intervention Shared Control

The described solution to minimal intervention shared control is outlined in Algorithm 1. Here Γ is the set of all possible trajectories γ that stem from the current state $x(t)$, and Γ_{safe} is the subset of safe trajectories. The inputs u_r and u_h are the signal sent to the robot, and the signal provide by the human partner, respectively. The $cost$ function describes how desirable a trajectory is based on it's distance from the human operator's input. If the human's command does not lead to an ICS, their exact input will be passed to the system; otherwise, a perturbation (computed as the minimum deviation required to ensure safety) will be applied to the control signal.

This exact (full information) algorithm is computationally infeasible to compute online for any reasonably complex human-in-the-loop system. In particular, the solution requires exploring an infinite set of potential actions over a receding horizon to (1) ensure collision-free trajectories [34], and (2) select the input that most closely matches the human's desired action. In this work, we instead propose an approximation to this solution that can be computed in real-time.

IV. Model Predictive Minimal Intervention Shared Control

To compute an approximation to the optimal solution described in the prior section, we make a few key methodological choices. First, instead of considering *all* possible trajectories from the current sate, we only consider a representative set that we generate by sampling densely (from a uniform prior) over the input space and only at the present time. We then predict the motion of the system over a receding horizon and reject any inputs that generate a trajectory that violate the defined safety constraints. From the subset of inputs that do not produce unsafe trajectories, we select the control signal that is closest to the human partner's input. Notably, this approach does not require a model of the user's actions, or knowledge of a desired goal, as the user is free to dynamically adjust their objective at each timestep. We describe each step of this algorithm in detail in the following subsections.

A. Model Representation and Data-driven Approximations

The majority of related work requires hand-written (and potentially complex) models of the system and control dynamics [3, 25, 28]. In this work, we instead use a representation

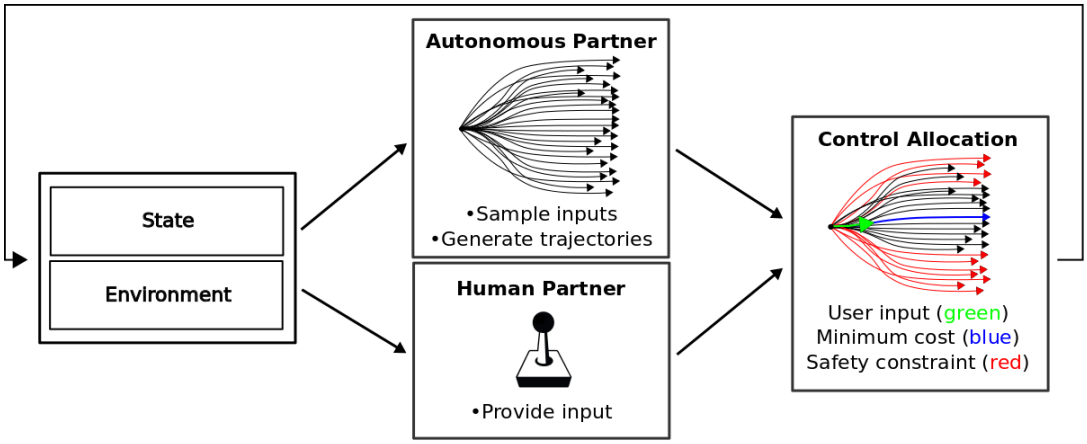


Fig. 1: Pictorial representation of our model predictive minimal intervention shared control (MPMI-SC) paradigm. The autonomous partner samples densely from the input space and generates a cloud of potential trajectories using a massively parallel processor. The human partner provides their desired input. The optimal control solution is then computed according to the Minimal Intervention Principle (MIP) and constrained based on the safety of the system over a receding horizon.

that is simple (i.e., linear) regardless of the underlying dynamics and can be learned from data when the model is not known *a priori*. This representation is known as the Koopman operator [21] and it can be used to model nonlinear dynamical systems as a linear operator because it *maps functions of state to functions of state* instead operating in the original state space. This representation is valid both when we know an analytical model of the system in the standard state space [21], and when we must learn the model from data [37].

In this work, we opt to learn the system model from data to demonstrate the efficacy of our approach when we have no prior knowledge of the robotic system. In particular, we use sparsity-promoting Dynamic Mode Decomposition [19] to select an appropriate basis and approximate the Koopman operator [21]. Data-driven Koopman operators have recently been explored as a method of generating model-based control in robotics [2] and in shared control [7, 8]. Unlike in prior work, however, the choice of the Koopman operator representation is explicitly motivated by our sampling-based policy generation method (see Section IV-B) and modern computational resources (e.g., multi-core CPUs and GPUs). That is, the Koopman operator representation is particularly well suited for sampling-based control as forward predicting the state of the system requires only a single matrix-vector multiplication, and generating a large set of potential trajectories requires only a single matrix-matrix multiplication. Both of these operations can be easily parallelized on a GPU [33]. Related work in model-based control for dynamic systems has utilized linear representations (e.g., Bayesian linear regression [35]), however, to the best of our knowledge, ours is the first work to develop a model-based controller the integrates a Koopman operator representation with sampling-based optimal control.

B. Sampling-based Optimal Control and Predictive Safety

To generate a set of *unconstrained* potential trajectories the user may wish to execute, we sample N inputs from an equally-spaced discretization of the control space. By relying on a uniform prior, we make no assumptions about the user's desired action at the next step (i.e., *there is no model of the user*). These samples can also be generated stochastically. However, stochasticity has known downsides in human-in-the-loop systems. For example, inputs that generate motion directly along a single dimension—a common desire of human operators—are unlikely to be sampled as they exist in segments of the input space that have near zero probability mass when sampling from a continuous distribution.

To ensure that our shared control algorithm only considers trajectories that satisfy the safety constraint described in Section III, we evaluate the configuration of the system at each time step over the receding horizon in each predicted trajectory. If the system violates geometric safety checks that are defined with respect to the environment, we reject the input that generated that trajectory. If, however, the system is safe over the entire course of the predicted trajectory, we consider that input as a viable solution. This can be seen as a *predictive notion of safety* as we evaluate the likelihood of a particular signal leading to a catastrophic failure by observing how we expect the controlled device to evolve over time.

C. Model Predictive Minimal Intervention Shared Control

The full Model Predictive Minimal Intervention Shared Control (MPMI-SC) approach can be seen in Fig. 1 and is outlined in Algorithm 2. The inputs are the current time t, the current state x_t and the human partner's input u_h. ξ is the sampled control, $\mathbb{U}$ is the distribution the control is sampled from, M is the dimensionality of the input space, N is the number of samples, and T is the prediction horizon. During forward prediction (Line 6), z_i is sampled i.i.d. from a white noise Gaussian process to account for inaccuracies in the dynamics model. Notably, this computation is done *in parallel on a GPU*. The learned Koopman operator system model $f(x, u)$ predicts the state x_p at the next timestep t_p,

Algorithm 2 MPMI-SC

procedure MPMI-SC(t, x_t, u_h)
  $\xi \sim \mathbb{U}^{MxN}$ ▷ unbiased control samples
  for i in N **in parallel do** ▷ forward predict system
    $t_p, x_p, safe \leftarrow t, x_t, \text{True}$
    while $t_p < t + T$ and $safe$ **do** ▷ prediction
      $x_p \leftarrow f(x_p, \xi(i)) + z_i$ ▷ z_i is Gaussian noise
      $safe = \text{isSafe}(x_p)$ ▷ system safe at x_p
      $t_p = t_p + \Delta t$ ▷ Δt is the timestep
    end while
    if $safe = \text{True}$ **then** ▷ safe over full trajectory
      store $\leftarrow \xi(i)$
    end if
  end for
  $u_r \leftarrow \text{argmin}(cost(\xi(i), u_h))\ \forall\, \xi(i) \in \text{store}$
  return u_r ▷ signal is safe and adheres to MIP
end procedure

which is evaluated for safety. The *cost* function minimizes the influence of the autonomous partner, and u_r is the control signal sent to the robot.

D. Minimal Intervention Principle and Expected Deviation

An additional benefit of our sampling-based approach is that we can provide an explicit bound on the sub-optimality of the applied control *with respect to the user's instantaneous desires*. In particular, the deviation between the user's input and the applied control signal (*when the user input is safe*) is upper bounded by half the distance between the sampled inputs. This can be computed based on the number of samples generated at each timestep and the Lebesgue measure [31] of the control space. This relationship is described in Equation (2).

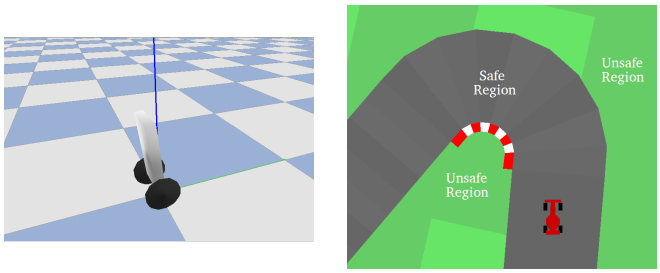
$$\mathbb{E}[\|u_h, u_r\|] = \frac{\lambda^*(U)}{2N} \tag{2}$$

where u_h is the human partners input, u_r is the signal sent to the robot, and $\|u, v\|$ is the Euclidean distance. $\lambda^*(U)$ is the n-dimensional volume (i.e., Lebesgue measure) of the bounded input space and N is the number of samples. As the number of samples grows ($N \to \infty$), the maximum deviation between the user's input and the applied signal shrinks to 0. However, we also note that while the number of samples (denominator) grows linearly, the Lebesgue measure (numerator) grows exponentially with each additional control dimension (i.e., the measure is defined as the Cartesian product of the intervals of each dimension). This describes a potential issue in the scalability of our sampling based solution; however, this issue is mitigated in the majority (if not all) of our application domains as the dimensionality of the input generally remains low as the human operator must capable of providing input to the system (e.g., using a joystick). The interval of each control dimension also generally remains small (e.g., [-1, 1] or [0, 1]) so that it is understandable by the human partner. A more in-depth discussion of the scalability of our algorithm is presented in the supplementary material.

V. Experimental Evaluation

We validate MPMI-SC with a human subjects study in two simulated environments which we describe below.

A. Simulated Environments



(a) Simulated balance bot. (b) Simulated race car.

Fig. 2: Pictorial representation of simulated environments.

The first dynamic system that users operate is a simulated balance bot (Fig. 2a). When controlling this system, users are told (1) that they are free to move around the environment however they would like, and (2) that they should try to make sure that body of the robot does not collide with the ground (i.e., the safety constraint). This can be challenging for a novice operator due to the stabilization requirements. The system is based on an open-source package [12] developed using the PyBullet physics engine. The observation space is a three-dimensional continuous vector that includes the angular position and velocity of the robot's body, and the linear velocity of the system. The input is a one dimensional continuous signal that sets a target velocity for both wheels.

The second dynamic system that users operate is a simulated race car (Fig. 2b). When controlling this system, users are told (1) that they are free to move around the environment however they would like, and (2) that they should take caution not to drive off of the road (i.e the safety constraint). This can be challenging for a novice operator due to the narrowness of the road and the fact that the car can go unstable (e.g., skid out) if the force applied to the system exceeds the friction limit of the ground surface. This system is based on an open-source package released by OpenAI [9] and developed using the Box2D physics engine. The observation space is a six dimensional continuous vector that includes the (x, y, θ) position of the car and the associated velocities $(\dot{x}, \dot{y}, \dot{\theta})$. The input to the system is a three dimensional continuous vector that defines the desired heading for the robot, the positive acceleration (gas) and negative acceleration (break).

The complexity of these two systems makes them useful as testbeds to validate the impact of our shared control algorithm. In particular, stabilization and environmental constraints are important challenges for robotic systems in human-centered domains where the human partner is often co-located with the mechanical system.

B. Experimental Design

To evaluate the impact of our shared control algorithm we ran a human-subjects study (n=20) in which participants

operated the system under two distinct interaction paradigms:

- User-only control (No assistance)
- Model Predictive Minimal Intervention Shared Control

The order in which the participants saw the two paradigms was randomized and counter-balanced to account for ordering effects. In each trial, participants were told to perform whatever action they would like so long as the system remained safe. They were also told that in some conditions an autonomous partner would help maintain safety, but they were not told *how* it would help. A trial would end when the system violated one of the defined safety constraints or after a maximum alloted time (balance bot: 20 seconds, race car: 30 seconds). Each participant interacted with the system 10 times in each environment under each control condition. In the race car environment, the morphology of the road was generated randomly at the start of each trial, however, the same random seeds were maintained across participants to ensure that each subject saw the same road configurations.

C. Implementation Details

1) Safety Computation: In both experimental environments, the system is considered unsafe when it violates physical barriers. For the balance bot this is defined as the robot body colliding with the ground. For the race car this is defined as driving off the track. To compute the safety of a predicted trajectory we evaluate the configuration of the system at each discrete timestep. Theoretically, these geometric checks provide strong safety guarantees as we always pick the applied control signal from the subset of sampled inputs that do not violate the defined bounds. In practice, the accuracy of this method relies both on precise system models and our ability to account for noise in the dynamics and/or sensors. In this work, we account for these errors by adding an inflated barrier beyond the natural collision points to reduce the impact of inaccurate predictions. Importantly, even with errors in the dynamics model, and noise in the sensor measurements, one can define an inflation radius that provides strong guarantees on the safety of the system [22, 23]. However, in this initial work, we simply rely on a hand-tuned inflation radius.

2) Algorithm Parameters: The implementation of our algorithm requires defining a small set of parameters, the two most important of which are the number of control samples N, and the length of the receding horizon T. As these parameters increase the approximation to the true solution improves, however, this comes at a cost of increased computational complexity. To address this issue, we provide a highly parallelized implementation of our algorithm that evaluates each trajectory independently on an NVIDIA GeForce 860M GPU (details in supplementary material). To provide a good user experience and demonstrate the scalability of our algorithm, we select a large N: 10,000 for the balance bot and 10,120[1] for the race car. The receding horizon T ($T = 30$ for the balance bot, $T = 25$ for the race car) was chosen based on observations of each system under MPMI-SC with random inputs.

[1]Chosen to produce equally spaced samples in all 3 input dimensions.

3) Basis Selection: Koopman operator system identification requires defining a basis to approximate the nominally infinite dimensional Hilbert space that the Koopman operator acts on. We select this basis through data-driven methods (as described in Section IV-A). In particular, we generate a set of 50 and 150 random basis functions for the balance bot and race car environments, respectively. The chosen learning algorithm [19] selects the most relevant basis functions for modeling each system, resulting in 6 and 26 basis functions for the balance bot and race car environments, respectively. All parameter choices are well documented in the open-source code.

VI. Results

We first explore the impact of our algorithm on the safety of the human-machine system, and the users' response to the intervention of the autonomy. We then detail the computational performance of our system using the defined parameter settings. Finally, we provide a secondary analysis of metrics that relate to the human operator's control skill and style.

A. Impact of Shared Control on Safety

To evaluate the impact of MPMI-SC on the safety of the joint system we compare (1) the average fraction of safe interactions to unsafe interactions and (2) the average time it took for the system to enter an unsafe state under each control paradigm (Fig. 3). To compare these values, we use a non-parametric Wilcoxon signed-rank test. In both experimental environments we find that MPMI-SC significantly improves the rate at which users are able to safely control the system when compared to a user-only control paradigm ($p < 0.005$). Similarly, we find that users are able to safely control the system for a significantly longer amount of time under shared control then when under user-only control ($p < 0.005$).

B. User Acceptance of Shared Control Paradigm

We next evaluate the users' acceptance of the assistance provided by the autonomous system. As mentioned in Section II, the majority of shared control systems assist a user in

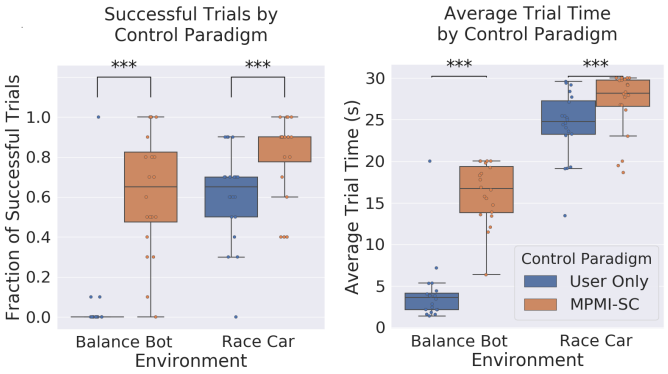


Fig. 3: Average success rate (left) and average time to failure (right), broken down by environment and control paradigm. The maximum interaction time was 20s in the balance bot and 30s in the race car. Both metrics improve significantly ($***: p < 0.005$) under shared control.

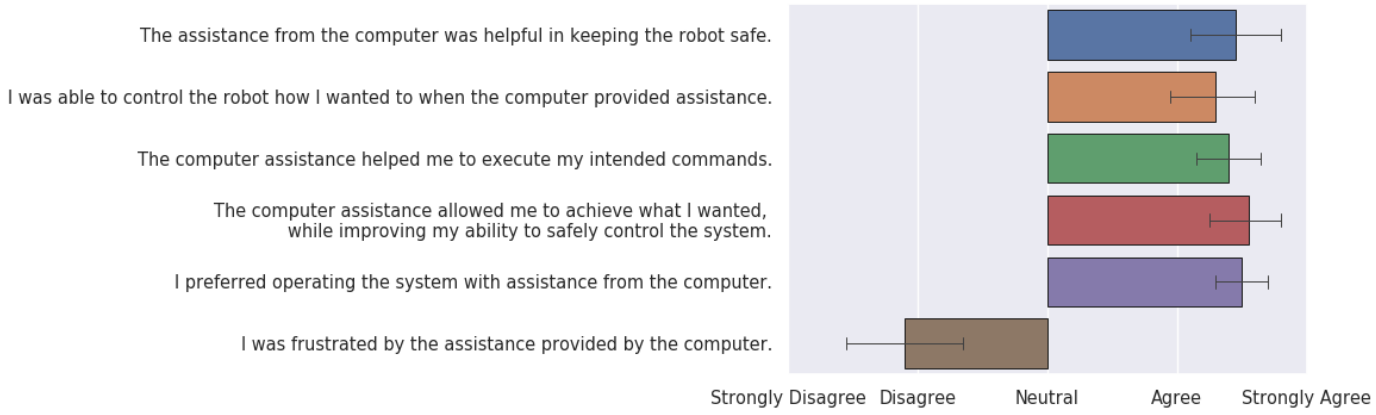


Fig. 4: Average user response (agreement) to post-experiment questionnaire. Black bars are standard deviations.

achieving a specific task. The assistance can therefore come at the expense of user satisfaction as the human partner often feels that they are fighting the autonomy [14]. In contrast, we develop a task-agnostic shared control paradigm that adheres to the human partner's desires at each instant.

To evaluate the user experience under MPMI-SC we asked participants to fill out a post-experiment questionnaire (Fig. 4). Statements were rated on a 5-point Likert-type scale where 1 represents strong disagreement and 5 represents strong agreement. Overall, user's felt the assistance provided by MPMI-SC helped them keep the robot safe and execute their intended commands. Perhaps most telling is that the participants strongly preferred operating the system with assistance from the computer, and did not feel frustrated by the assistance.

C. System Performance

We now describe the computational efficiency of MPMI-SC using the parameters defined in Section V-C2. Our algorithm is capable of generating trajectories at $\sim$7000 Hz and $\sim$3500 Hz in the balance bot and race car environments. Incorporating the safety checks, the system runs at $\sim$100 Hz and $\sim$60 Hz, respectively (i.e., between 600,000 and 1,000,000 trajectories every second). MPMI-SC is faster in the balance bot as it relies on a smaller basis and more efficient safety checks. A detailed description of our GPU implementation and the scalability of MPMI-SC is provided in the supplementary material.

We also compute the maximum possible deviation between a user's input and the closest safe signal at each timestep. This value is determined by evaluating Equation (2) with a known n-dimensional input volume ($\lambda*$) and number of samples (N). If the user's input is considered safe over the receding horizon (i.e., through our prediction method described in Section IV-B), this value represents *the maximum influence of the autonomous partner on the applied signal*. If the user's input is not safe over the predicted trajectory, this bound represents *the maximum possible difference between the true closest safe signal and the applied signal*. For the balance bot, the maximum deviation is $\mathbb{E} = \mathbf{0.0001}$ because $\lambda^*(U) = 2$ (i.e., 1D input that spans $[-1, 1]$) and $N = 10,000$. For the race car, the maximum deviation is $\mathbb{E} = \mathbf{0.0002}$ because $\lambda^*(U) = 4$ (i.e., 3D input where the heading spans $[-1, 1]$, the gas spans $[0, 1]$, and the break spans $[-1, 0]$) and $N = 10,120$.

How large an impact an intervention of the described magnitudes has on the dynamics of a dynamic system is dependent on each particular machine. However, we note that both values described in this work are *very* small. We therefore hypothesize that there is likely no discernible difference between the effect of the intended signal and the applied signal when the user's input is considered safe. If there were a large impact on the system dynamics due to an intervention of this magnitude, the system may be chaotic and therefore challenging to operate no matter what controller was used [16].

D. User Control Skill and Style

Finally, we provide a secondary analysis of the data collected during our experiment to evaluate the users' control skill and style. The proposed metrics are easy to compute based on the users' input and the trajectories generated at each timestep.

1) Operator Control Skill: We first calculate the average observed deviation between the user input and the *closest safe signal* as a proxy for the users' understanding of the system dynamics and their control skill. To illustrate this relationship consider a car making a tight turn. If the human partner understands that they need to change their heading and speed to stay on the road during the turn, this metric will remain low (Sec. IV-D). If, however, the human partner does not understand this relationship and relies on the autonomous partner to maintain the safety of the system during the turn, this metric will increase. The average value of this metric will be within the described bound ($\mathbb{E}$, above) if, and only if, *all* of the user's controls are considered safe during the course of their interaction. Otherwise this value will increase according to the minimal amount required to maintain safety.

2) Operator Control Style: We then calculate the average percentage of sampled rollouts that are safe at each timestep as a proxy for the user's control style. A lower percentage of safe trajectories suggests that the user is operating the system in a more *dangerous* manner (i.e., the system may be closer to entering an ICS). However, we note that this metric alone does not directly relate to the user's control skill as there are many cases in which one may trade off notions of

Metric	Control	Balance Bot	Race Car
Avg. Deviation	User-only	0.46 ± 0.37	0.05 ± 0.03
	MPMI-SC	0.21 ± 0.21	0.07 ± 0.06
Avg. % Safe Rollouts	User-only	0.28 ± 0.15	0.38 ± 0.09
	MPMI-SC	0.50 ± 0.10	0.35 ± 0.08

TABLE I: Average deviation and average percentage safe rollouts, broken down by environment and by control condition.

safety for performance. To illustrate this relationship consider, again, a person in a car taking a tight turn to decrease the time of their drive. If this value correlates positively with safe control of the system, we can say that the operator likely has a high degree of skill and is explicitly trading off conventional notions of safety (e.g., distance from an obstacle) for improved performance. However, if this metric correlates positively with unsafe behavior, it is likely that the operator is unskilled and making poor control decisions.

3) Analysis: In the balance bot environment, we find that the user's input more closely aligns with the closest safe signal under MPMI-SC (Table I). We also observe that the system remains in a state where more potential actions remain safe over the horizon. We interpret these results as evidence that the users provided more competent control with assistance than without. In the race car environment, we find that, in both control conditions, the user's input is nearly equally aligned with the closest safe signal, and that there are similar percentages of potentially safe actions. We interpret these results as evidence that MPMI-SC had less impact in this environment then in the balance bot. Evidence of this interpretation can also be seen in Figure 3 where we find larger raw differences between the safety metrics in the balance bot environment than in the race car environment. However, the differences between each of these primary safety metrics are statistically significant suggesting that, even when MPMI-SC is less impactful, it still meaningfully improves the safety of the joint system. Both proposed metrics are indirect measures of the user's control skill and style, but the preliminary results (Table I) suggest they warrant further investigation. In future work we plan to evaluate how they evolve over time for an individual operator.

VII. Discussion

A. Study Observations

One piece of information that is not reflected in our analysis is how people alter their *control strategy* when they are under different paradigms. For example, participants were observed *testing* the limits of the assistance. That is, users would intentionally operate the system at the boundary of safety, and even act in an adversarial manner to test the reliability of MPMI-SC. In contrast, under user-only control, participants were much more cautious. This is potentially a consequence of not explaining *how* the autonomous partner would provide assistance; however, we also believe this behavior aligns with human nature as people often *explore* while they learn.

Another observation relates to why MPPI-SC had a larger impact on the balance bot than on the race car. In particular, participants were more likely to have prior experience operating car-like systems then unstable machines like the balance bot, which could mean they were better able to provide safe control based solely on experience and intuition.

B. Limitation in Prediction of Safety

As described in Section IV, the control signal sent to the robot is selected based on the user's desires at each instant and therefore does not require a model of the user. However, to compute the safety of a given control action, we implicitly embed a naïve model of the user that assumes the human partner will continue to apply nearly the same input over the time-horizon. The negative implication of this assumption is that we will occasionally reject user inputs that *seem* dangerous, but are not in reality if the user quickly adjusts their strategy. Therefore, there will be trajectories that the person would like to execute and are safe (by recovering at a later timestep) that MPMI-SC incorrectly rejects. Despite this limitation, it is possible that our iterative, receding horizon approach alleviates the impact on the user's acceptance by recomputing the set of dangerous actions at each timestep. We leave a deeper evaluation of this idea to future work.

VIII. Conclusion

In conclusion, we described a shared control paradigm that enhances a human partner's ability to operate complex, dynamic machines by incorporating safety constraints without explicit knowledge of the user's long-term objective. Our approach relies on a simple representation of the joint system (i.e., the Koopman operator) which, in turn, means we can very quickly generate and evaluate the safety of a large number of potential trajectories through the parallelization capabilities of a GPU. Importantly, this representation can be learned from data and therefore generalizes to any pair of partners. Finally, our approach adheres to the minimal intervention principle to ensure that the human partner is allocated the majority of the decision making authority throughout the interaction.

We evaluated the efficacy of our Model Predictive Minimal Intervention Shared Control (MPMI-SC) paradigm with a human-subjects study consisting of 20 participants. The results demonstrated that our approach is able to improve the general safety of the joint system without *a priori* knowledge of the user's desires. Additionally, we found that the participants enjoyed the assistance provided by the autonomy (and reported low levels of frustration), a feature lacking in many shared control paradigms [14]. We have released our code with an open-source license on GitHub at https://github.com/asbroad/mpmi_shared_control.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grants CNS 1329891 & 1837515. 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. The authors would also like to thank Emily Bernstein for her help in the study design.

References

[1] David A Abbink, Tom Carlson, Mark Mulder, Joost CF de Winter, Farzad Aminravan, Tricia L Gibo, and Erwin R Boer. A Topology of Shared Control Systems—Finding Common Ground in Diversity. *IEEE Transactions on Human-Machine Systems*, (99):1–17, 2018.

[2] Ian Abraham, Gerardo De La Torre, and Todd D Murphey. Model-Based Control Using Koopman Operators. In *Robotics: Science and Systems*, 2017.

[3] Sterling J Anderson, Sisir B Karumanchi, and Karl Iagnemma. Constraint-based Planning and Control for Safe, Shared Control of Ground Vehicles. In *IEEE International Vehicles Symposium*, 2012.

[4] Sterling J Anderson, Sisir B Karumanchi, Karl Iagnemma, and James M Walker. The Intelligent CoPilot: A Constraint-based Approach to Shared-Adaptive Control of Ground Vehicles. *Intelligent Transportation Systems Magazine*, 5(2):45–54, 2013.

[5] Sterling J Anderson, James M Walker, and Karl Iagnemma. Experimental Performance Analysis of a Homotopy-based Shared Autonomy Framework. *Transactions on Human-Machine Systems*, 44(2):190–199, 2014.

[6] Antoine Bautin, Luis Martinez-Gomez, and Thierry Fraichard. Inevitable Collision States: A Probabilistic Perspective. In *International Conference on Robotics and Automation*, pages 4022–4027. IEEE, 2010.

[7] Alexander Broad, Todd Murphey, and Brenna Argall. Learning Models for Shared Control of Human-Machine Systems with Unknown Dynamics. In *Robotics: Science and Systems*, 2017.

[8] Alexander Broad, Todd Murphey, and Brenna Argall. Operation and Imitation under Safety-Aware Shared Control. In *International Workshop on the Algorithmic Foundations of Robotics*, 2018.

[9] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *arXiv*, abs/1606.01540, 2016.

[10] Tom Carlson and Yiannis Demiris. Collaborative Control for a Robotic Wheelchair: Evaluation of Performance, Attention, and Workload. *Transactions on Systems, Man, and Cybernetics*, 42(3):876–888.

[11] Tom Carlson and Yiannis Demiris. Human-Wheelchair Collaboration through Prediction of Intention and Adaptive Assistance. In *IEEE International Conference on Robotics and Automation*, pages 3926–3931, 2008.

[12] Yannis Chatzikonstantinou. Balance bot. https://github.com/yconst/balance-bot, 2018.

[13] Anca D Dragan and Siddhartha S Srinivasa. A Policy-Blending Formalism for Shared Control. *The International Journal of Robotics Research*, 32(7):790–805, 2013.

[14] Ahmetcan Erdogan and Brenna D Argall. Prediction of User Preference over Shared Control Paradigms for a Robotic Wheelchair. In *IEEE International Conference on Rehabilitation Robotics*, July 2017.

[15] Stephen M Erlien, Susumu Fujita, and Joseph Christian Gerdes. Shared Steering Control using Safe Envelopes for Obstacle Avoidance and Vehicle Stability. *Transactions on Intelligent Transportation Systems*, 17(2):441–451, 2016.

[16] Alexander L Fradkov and Robin J Evans. Control of Chaos: Methods and Applications in Engineering. *Annual Reviews in Control*, 29(1):33–56, 2005.

[17] Thierry Fraichard and Hajime Asama. Inevitable Collision StatesA Step Towards Safer Robots? *Advanced Robotics*, 18(10):1001–1024, 2004.

[18] Shervin Javdani, Henny Admoni, Stefania Pellegrinelli, Siddhartha S Srinivasa, and J Andrew Bagnell. Shared Autonomy via Hindsight Optimization for Teleoperation and Teaming. *The International Journal of Robotics Research*, 37(7):717–742, 2018.

[19] Mihailo R Jovanović, Peter J Schmid, and Joseph W Nichols. Sparsity-promoting Dynamic Mode Decomposition. *Physics of Fluids*, 26(2):024103, 2014.

[20] Hyun K Kim, J Biggs, W Schloerb, M Carmena, Mikhail A Lebedev, Miguel AL Nicolelis, and Mandayam A Srinivasan. Continuous Shared Control for Stabilizing Reaching and Grasping with Brain-Machine Interfaces. *IEEE Transactions on Biomedical Engineering*, 53(6):1164–1173, 2006.

[21] Bernard O Koopman. Hamiltonian Systems and Transformation in Hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.

[22] Shreyas Kousik, Sean Vaskov, Matthew Johnson-Roberson, and Ram Vasudevan. Safe Trajectory Synthesis for Autonomous Driving in Unforeseen Environments. In *ASME Dynamic Systems and Control Conference*, 2017.

[23] Shreyas Kousik, Sean Vaskov, Fan Bu, Matthew Johnson-Roberson, and Ram Vasudevan. Bridging the Gap Between Safety and Real-Time Performance in Receding-Horizon Trajectory Design for Mobile Robots. *arXiv:1809.06746*, 2018.

[24] Steven M LaValle and James J Kuffner Jr. Randomized Kinodynamic Planning. *The International Journal of Robotics Research*, 20(5):378–400, 2001.

[25] K. Leung, E. Schmerling, M. Chen, J. Talbot, J. C. Gerdes, and M. Pavone. On Infusing Reachability-Based Safety Assurance within Probabilistic Planning Frameworks for Human-Robot Vehicle Interactions. In *International Symposium on Experimental Robotics*, 2018.

[26] Selma Musić and Sandra Hirche. Control Sharing in Human-Robot Team Interaction. *Annual Reviews in Control*, 2017.

[27] Siddharth Reddy, Anca D Dragan, and Sergey Levine. Shared Autonomy via Deep Reinforcement Learning. In *Robotics: Science and Systems*, 2018.

[28] Wilko Schwarting, Javier Alonso-Mora, Liam Paull, Sertac Karaman, and Daniela Rus. Parallel autonomy in automated vehicles: Safe motion generation with mini-

mal intervention. In *IEEE International Conference on Robotics and Automation*, 2017.

[29] Wilko Schwarting, Javier Alonso-Mora, Liam Paull, Sertac Karaman, and Daniela Rus. Safe Nonlinear Trajectory Generation for Parallel Autonomy with a Dynamic Vehicle Model. *IEEE Transactions on Intelligent Transportation Systems*, (99):1–15, 2017.

[30] Victor A Shia, Yiqi Gao, Ramanarayan Vasudevan, Katherine Driggs Campbell, Theresa Lin, Francesco Borrelli, and Ruzena Bajcsy. Semiautonomous Vehicular Control using Driver Modeling. *IEEE Transactions on Intelligent Transportation Systems*, 15(6):2696–2709, 2014.

[31] Terence Tao. *An Introduction to Measure Theory*. American Mathematical Society, 2011.

[32] Evangelos Theodorou, Jonas Buchli, and Stefan Schaal. A Generalized Path Integral Control Approach to Reinforcement Learning. *Journal of Machine Learning Research*, 11(Nov):3137–3181, 2010.

[33] Vasily Volkov and James W Demmel. Benchmarking GPUs to Tune Dense Linear Algebra. In *International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–11. IEEE, 2008.

[34] Li Wang, Aaron D Ames, and Magnus Egerstedt. Safety Barrier Certificates for Collisions-Free Multirobot Systems. *IEEE Transactions on Robotics*, 33(3):661–674, 2017.

[35] Grady Williams, Paul Drews, Brian Goldfain, James M Rehg, and Evangelos A Theodorou. Aggressive Driving with Model Predictive Path Integral Control. In *International Conference on Robotics and Automation (ICRA)*, pages 1433–1440. IEEE, 2016.

[36] Grady Williams, Andrew Aldrich, and Evangelos A Theodorou. Model Predictive Path Integral Control: From Theory to Parallel Computation. *Journal of Guidance, Control, and Dynamics*, 40(2):344–357, 2017.

[37] Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A Data-driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition. *Journal of Nonlinear Science*, 25(6): 1307–1346, 2015.