A Model Predictive Control Approach for Reach Redirection in Virtual Reality

Eric J. Gonzalez
Stanford University
Stanford, CA, USA

Elyse D.Z. Chase
Stanford University
Stanford, CA, USA

Pramod Kotipalli
Stanford University
Stanford, CA, USA

Sean Follmer
Stanford University
Stanford, CA, USA

ABSTRACT

Reach redirection is an illusion-based virtual reality (VR) interaction technique where a user's virtual hand is shifted during a reach in order to guide their real hand to a physical location. Prior works have not considered the underlying sensorimotor processes driving redirection. In this work, we propose adapting a sensorimotor model for goal-directed reach to obtain a model for visually-redirected reach, specifically by incorporating redirection as a sensory bias in the state estimate used by a minimum jerk motion controller. We validate and then leverage this model to develop a Model Predictive Control (MPC) approach for reach redirection, enabling the real-time generation of spatial warping according to desired optimization criteria (e.g., redirection goals) and constraints (e.g., sensory thresholds). We illustrate this approach with two example criteria – redirection to a desired point and redirection along a desired path – and compare our approach against existing techniques in a user evaluation.

CCS CONCEPTS

• **Human-centered computing** → **Virtual reality**; **HCI theory, concepts and models**.

KEYWORDS

Haptic Retargeting, Model Predictive Control, Optimization, Reach, Redirection, Virtual Reality

ACM Reference Format:
Eric J. Gonzalez, Elyse D.Z. Chase, Pramod Kotipalli, and Sean Follmer. 2022. A Model Predictive Control Approach for Reach Redirection in Virtual Reality. In *CHI Conference on Human Factors in Computing Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3491102.3501907

1 INTRODUCTION

Virtual Reality (VR) offers users immersive experiences in environments that differ from the world around them. Beyond audiovisual feedback, haptic feedback can further increase users' sense of immersion in VR [33]. However, accurately representing the physical properties and locations of virtual objects is challenging. Additionally, users' real-world environments often have constraints that further limit their ability to be mapped directly to a virtual scene. To address these issues, researchers have begun exploring ways of leveraging visual dominance [18] — or the tendency for vision to strongly influence perception — to affect users' perceived haptic and proprioceptive sensations in VR.

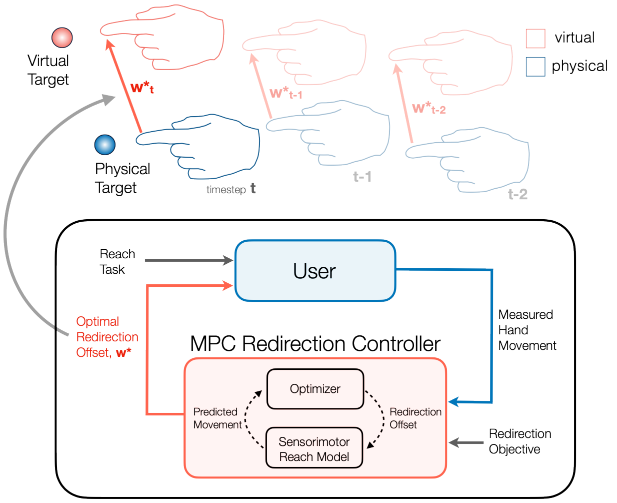

Figure 1: Overview of the proposed Model Predictive Control approach to reach redirection. As the user reaches to a target (top), the optimal virtual hand offset is computed at each timestep based on the redirection objective.

Reach redirection is one such approach that aims to influence the user's hand trajectory as they reach for an object by smoothly offsetting their virtual hand from their real hand [3, 30]. By loosening the requirement of 1-to-1 movement, researchers have shown how redirection can be used to convincingly repurpose physical props [3], generate sensations of weight [40], and influence our perception of interactive devices [2, 20].

While there is much recent work studying the limitations and impacts of redirection on user experience [5, 21, 52], the generation of novel redirection algorithms has been less explored. Some higher-level strategies, such as blink-suppressed hand redirection [53],

CHI '22, April 29-May 5, 2022, New Orleans, LA, USA

ACM ISBN 978-1-4503-9157-3/22/04…$15.00
https://doi.org/10.1145/3491102.3501907

broaden the capabilities of existing redirection techniques, but the low-level generation of real-to-virtual offsets remains largely limited to linear interpolation between known points in space[3, 7, 37].

This predefined spatial interpolation strategy cannot easily adapt to changes in the user's environment, uncertainty about the user's intent, or individual differences between users. Moreover, the psychophysical impacts of redirection are not currently considered in the warping algorithm itself, except through ad hoc constraints. To enable redirection that directly considers the trade-offs between users' qualitative perception and the various goals of a redirected action, a more robust and generalizable framework is needed. Furthermore, to capture how spatial warping more generally impacts user reaching movements, an improved sensorimotor control model of redirection is also needed. Presently, most applications treat redirection as a "black box" applied to achieve a desired result (e.g., guide the user's hand to a specific point), without consideration for the underlying process. However, explicitly considering this process may ultimately lead to better algorithms than interpolation.

Towards this goal, we propose a novel Model Predictive Control (MPC) approach for reach redirection in VR, illustrated in Figure 1. MPC is a general framework for the real-time control of a dynamic system which satisfies a given optimization objective [17]. The MPC framework enables the previously mentioned redirection trade-offs to be intuitively incorporated as optimization costs. Moreover, MPC leverages a dynamic system model to plan optimal actions (in this case, spatial warping) over a horizon of potential future states. Since this process occurs continuously, spatial warping could adapt to changes within the user's environment, such as the introduction of an obstacle that should be avoided.

We first present a practical dynamic model for redirected reaching in VR. Based on well-established sensorimotor control research [43], we model the user's hand motion as a minimum jerk (MJ) controller [16, 26] and incorporate redirection as a sensory bias in the user's estimate of their hand state. This model allows us to simulate a user's hand trajectory given any redirection algorithm and their reach target/time.

Leveraging this model, we then describe our proposed MPC approach for controlling redirection. We explore two different cost functions to illustrate the versatility of this approach, specifically enabling redirection to a desired endpoint and along a desired path. Both cost functions also encouraging the resulting warp to be smooth and minimal. By tuning the parameters within each cost function, it is possible to adjust the relative weight between task achievement (e.g., guide the hand to a precise point) and user comfort (e.g., limit rapid changes in applied warp). Finally, we evaluate our MPC approach against standard redirection methods in a target acquisition study.

1.1 Contributions

- Adapting sensorimotor control theory to model visually-guided redirected reaching in VR
- A novel MPC approach for reach redirection including system model, parameters, sample cost functions, and implementation example
- An evaluation comparing MPC-based reach redirection against traditional redirection methods

2 RELATED WORK

2.1 Hand Redirection in VR

Hand redirection techniques leverage visual dominance [18] to adjust the trajectory of the user's hand [30]. Typically this is done by smoothly offsetting the position of the virtual hand from the measured hand position [32]. Azmandian et al. introduce two such warping methods – body warping and world warping – in their broader Haptic Retargeting technique [3]. Other techniques include Thin Plate Spline warping [31] and functional optimization of 3D space [55]. In each case, however, the warping used is fully defined by the target layout and typically does not update during the reach.

Hand redirection has been used in a wide array of applications, including changing the perceived shape [30, 55] and location [3, 7] of passive haptic props. Researchers have also shown how redirection can be useful for improving the ergonomic layout of targets [14, 37], inducing weight perception [40], and improving the perceived performance and capabilities of haptic devices [2, 20, 54].

Researchers have identified the detection thresholds of hand redirection in a variety of scenarios [5, 13, 21, 52]. Other important factors, such as impact on comfort [7] and task performance [22, 32], have also been explored. Such insights are invaluable to the development of new redirection techniques and applications which must balance their goal (e.g. remapping a set of targets) while maintaining user agency, immersion, and comfort. In our work, we propose an MPC framework as a way to incorporate such perceptual considerations directly into the generation of spatial warping.

Recently, Lebrun et al. proposed a trajectory model for desktop-scale hand redirection based on Bézier curves [35]. While their model captures final trajectory shape, in this work we propose a dynamic model which captures trajectory formation over time.

2.2 Models of Goal-Directed Movement

Flash & Hogan [16] proposed and experimentally-validated one of the earliest mathematical models of human arm movement – the minimum jerk (MJ) model. This model suggests humans minimize the jerk (rate of change of acceleration) of their hand's movement during reaching. Hoff & Arbib [25] formulated the MJ model as a feedback control law which can account for mid-reach target perturbations; follow up work further considers the change in reach time induced by target perturbation [25]. Saunders & Knill [43] extended Hoff & Arbib's feedback control model to include the effect of minor visual perturbations of the hand while reaching. While this approach does not account for trajectory uncertainty or individual differences in strategy between users, the MJ feedback law has a closed-form, analytical solution making it convenient for real-time simulation.

Other models have explicitly considered limb dynamics during movement generation [28]. Uno et al. [48] found that minimizing the overall change in joint torque during reach reproduced experimental trajectories not captured by the MJ model. Such biomechanical models better capture the subtleties of human movement including joint limits, but introduce significant nonlinearities and thus increase computational cost.

More generally, the human sensorimotor system has been shown to be well-modeled using stochastic optimal control methods accounting for motor and sensory noise [49]. Harris and Wolpert

[23] found that open-loop minimization of endpoint variance produced plausible reach trajectories when multiplicative motor noise was accounted for. Todorov [47] generalized this into a closed-loop stochastic optimal control model that also incorporates sensory feedback. Alternatively, Bye & Nielson [6] presented the BUMP model for motor response planning, which incorporates variable horizon predictive control and accounts for both linear [45] and logarithmic [15] speed-accuracy tradeoffs [51]. These models accurately reproduce a broad range of movements, but require significant computation to generate trajectory estimates, making them difficult to integrate into real-time systems.

Recently, HCI researchers have applied control theoretic models to reaching in the context of pointing [38]. Do et al. [9] recently presented a thorough simulation of point-and-click behavior with a mouse based on intermittent optimal control leveraging the BUMP model [6], replicating the trajectory variance and task completion time of real users. Such an approach is promising for the offline simulation of redirection, but is challenging to incorporate within a real-time optimization process. Bachynskyi et al. [4] provide two models for the dynamics of 3D point-to-point hand movements. These models replicate user data well, but do not consider sensory feedback and their parameters must be fit to each user.

Importantly, recent work presented a preliminary analysis comparing redirected velocity profiles to those expected by a naive (non-feedback) MJ model [19]. Their results suggest that incorporating sensory feedback is needed for more accurate redirection modeling. Building on previous work in sensorimotor control, in this work we model redirection as sensory bias within an MJ feedback controller.

2.3 Real-time Optimization in HCI

Prior work in HCI has explored real-time optimization as a tool to enhance interactive systems. Inverse optimal control techniques have been used to infer users' intended targets during mouse pointing [56] and reaching [36]. Nescher et al. apply an MPC framework for planning redirected walking in virtual reality [39]; however, their approach operates at a higher planning level – selecting between predefined strategies – and does not consider the sensorimotor loop. Other work has considered using MPC to provide better haptic guidance [41]. Langerak et al. [34] leverage model predictive contouring control to enable real-time electromagnetic haptic guidance for drawing. We take inspiration from their approach in this work, where instead we utilize a model of redirected reach and consider the warping applied to the virtual hand as an abstract form of guidance.

3 A MODEL OF REDIRECTED REACH

3.1 Overview

Our initial goal is to model the trajectory of the user's real hand in space given a visual target and an applied redirection algorithm. We begin by considering a simplified version of the user's sensorimotor control process, illustrated in Figure 2. An internal motion controller generates a control signal u which drives the user's movement dynamics and updates their true hand state $\mathbf{x}$. This true state is then processed through the sensory system to yield an *estimated* state $\widehat{\mathbf{x}}$. Critically, it is this estimated state which the internal motion controller uses to generate the control signal. Differences between the estimated and true state will therefore impact the generated movement of the hand.

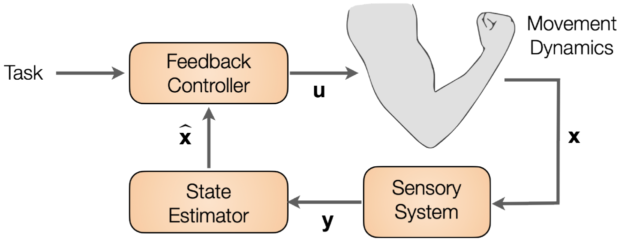

Figure 2: Overview of the sensorimotor control process, where u is the motor command, x is the hand state, y is the sensory measurement, and $\hat{x}$ is the estimated hand state.

Various models for the user's internal motion controller and movement dynamics during reaching have been explored, as described in Section 2. By augmenting such an existing model with sensory feedback that captures redirection as a shift in the visually estimated hand state, we show that we can produce a practical model for redirected reaching. Note that in this approach we consider visual feedback alone in the sensory estimation process, while in reality the sensory system integrates multiple channels (e.g., vision, proprioception, tactile) to estimate the various states of the body. Vision, however, has been shown to play a dominant role in reaching even during rapid movement [42–44], as demonstrated clearly by the success of previous redirection applications.

In the remainder of this section we present the selected user reach model, detail the required adjustments to account for redirection, and present the results of our model simulation and validation.

3.2 User Reach Model

We build our redirected reach model by augmenting an existing model that captures how humans reach under normal conditions. We select the Minimum Jerk (MJ) model [16] for reach trajectory generation for its simplicity and generalizability. The MJ principle states that humans tend to minimize the derivative of acceleration (jerk) when generating reaching movements. Hoff & Arbib formulated this principle as a feedback control law [25], which we leverage as the basis for our redirected reach model. For simplicity, we consider only a single movement dimension in our model derivation. We present the extension to 3D following our derivation.

In this model, position and velocity are integrated from the hand's acceleration, which is driven by a jerk signal u. Let the state variable $\mathbf{x} = [p \;\; v \;\; a \;\; T]^T$ be the position p, velocity v, and acceleration a of the user's hand, along with the target position T, and let the internal control variable u be the commanded jerk. The discrete dynamics of the system with timestep δt are then:

$$\mathbf{x}_{k+1} = \begin{bmatrix} 1 & \delta t & 0 & 0 \\ 0 & 1 & \delta t & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} 0 \\ 0 \\ \delta t \\ 0 \end{bmatrix} u_k = A\mathbf{x}_k + Bu_k, \qquad (1)$$

Given the remaining reach time at timestep k, the MJ principle can formulated as a feedback control law of the form [25]:

$$u_k = K_k \mathbf{x}_k = \begin{bmatrix} \frac{60}{R_k^3} & \frac{-36}{R_k^2} & \frac{-9}{R_k} & \frac{-60}{R_k^3} \end{bmatrix} \mathbf{x}_k \tag{2}$$

where R_k is the time remaining in the reach at timestep k. Note that the gain matrix K_k varies as R_k changes.

Hoff & Arbib used this feedback model to explain the movements caused by shifting the target position T mid-reach. In contrast, we aim to adapt this model to explain the hand movements induced during redirection by considering a shifting estimate of the user's hand state.

3.3 Redirection as a Sensory Bias

Redirection algorithms work by computing and applying a desired offset between the user's physical and virtual hand:

$$p_k^v = p_k + w_k \tag{3}$$

where p^v is the virtual position of the hand, p is the measured position of the physical hand, and w is the computed offset at timestep k.

For our model, we make the assumption that the state of the virtual hand (i.e., its position, velocity, acceleration) forms the user's internal estimate of their hand state $\widehat{\mathbf{x}}$. That is,

$$\widehat{\mathbf{x}} = \mathbf{x}^v = \begin{bmatrix} p^v & v^v & a^v & T^v \end{bmatrix}^T \tag{4}$$

where p^v, v^v, and a^v, are the position, velocity, and acceleration of the virtual hand, respectively, and T^v is the position of the virtual target (i.e., the target which the user is reaching towards). From a state estimation perspective, Eq. 3 illustrates that w can be thought of as a bias in the user's estimate of their hand position.

3.4 Modeling Redirected Trajectories

Our claim is that the dynamics of a redirected reach can be modeled by considering this estimated state as the input to Hoff & Arbib's MJ feedback control law. That is, the dynamics of redirected reach can be expressed as:

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + BK_k\widehat{\mathbf{x}}_k \tag{5}$$

In order to model how all of $\hat{\mathbf{x}}$ changes as a function of p and w, we consider an augmented state $\mathbf{z}_k$ which consists of $\mathbf{x_k}$ (the state of the user's real hand) as well as two steps of position and control (w) history.

$$\mathbf{z}_k = \begin{bmatrix} p_k & v_k & a_k & T_k & p_{k-1} & w_{k-1} & p_{k-2} & w_{k-2} \end{bmatrix}^T \tag{6}$$

We then estimate the velocity and acceleration of the virtual hand using backwards difference in order to relate v^v and a^v to p and w, where δ_t is the discrete timestep:

$$\mathbf{x}_k^v = C\mathbf{z}_k + Dw_k$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{\delta t} & 0 & 0 & 0 & \frac{-1}{\delta t} & \frac{-1}{\delta t} & 0 & 0 \\ \frac{1}{\delta t^2} & 0 & 0 & 0 & \frac{-2}{\delta t^2} & \frac{-2}{\delta t^2} & \frac{1}{\delta t^2} & \frac{1}{\delta t^2} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, D = \begin{bmatrix} 1 \\ \frac{1}{\delta t} \\ \frac{1}{\delta t^2} \\ 0 \end{bmatrix} \tag{7}$$

The dynamics for our augmented state $\mathbf{z}_k$ are then given by:

$$\mathbf{z}_{k+1} = \widetilde{A}\mathbf{z}_k + \widetilde{B}_1 K_k \widehat{\mathbf{x}}_k + \widetilde{B}_2 w_k$$

$$\widetilde{A} = \begin{bmatrix} 1 & \delta t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \delta t & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \widetilde{B}_1 = \begin{bmatrix} 0 \\ 0 \\ \delta t \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tilde{B}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \tag{8}$$

Combining with our estimation equations, we arrive at the complete dynamic model for redirected reaching to a virtual target positioned at T^v:

$$\begin{aligned} \mathbf{z}_{k+1} &= \left(\widetilde{A} + \widetilde{B}_1 K_k C\right)\mathbf{z}_k + \left(\widetilde{B}_1 K_k D + \widetilde{B}_2\right) w_k \\ &= \widehat{A}_k \mathbf{z}_k + \widehat{B}_k w_k \end{aligned} \tag{9}$$

The result is a linear, time-varying dynamic model for state $\mathbf{z}$ with command input w. The time-varying nature of the dynamics are driven by the internal MJ feedback matrix Kwhich varies with the estimated time remaining in the reach R.

3.5 Extending to 3D Trajectory Formation

The preceding sections assumed movement in a single dimension to simplify the derivation. Due to the independence of movement in each dimension suggested by the MJ principle[16], we can readily expand these results to three dimensions as follows, where $\mathbf{z}^i$ contains the i^{th} components of the augmented hand state and w^i is the i^{th} component of the redirection offset.

$$\begin{bmatrix} \mathbf{z}_{k+1}^x \\ \mathbf{z}_{k+1}^y \\ \mathbf{z}_{k+1}^z \end{bmatrix} = \begin{bmatrix} \widehat{A}_k^x & 0 & 0 \\ 0 & \widehat{A}_k^y & 0 \\ 0 & 0 & \widehat{A}_k^z \end{bmatrix} \begin{bmatrix} \mathbf{z}_k^x \\ \mathbf{z}_k^y \\ \mathbf{z}_k^z \end{bmatrix} + \begin{bmatrix} \widehat{B}_k^x & 0 & 0 \\ 0 & \widehat{B}_k^y & 0 \\ 0 & 0 & \widehat{B}_k^z \end{bmatrix} \begin{bmatrix} w_k^x \\ w_k^y \\ w_k^z \end{bmatrix} \tag{10}$$

3.6 Simulation & Comparison with Experimental Data

We validate our model by simulating a set of reaches and comparing with experimental results.

3.6.1 Dataset. We use the dataset of redirected reaches generated by Gonzalez et al. [19] for comparison, which contains a total of 2400 reaches performed by 12 users in VR. Each user completed 200 reaches to 5 targets (2 cm diameter circles) – 100 reaches were redirected to the targets, while the remaining 100 were performed normally. Position trajectories of users' real and virtual hands were recorded for each trial. We compare our model against both redirected and normal, non-redirected reaches.

3.6.2 Target Layout & Redirection Algorithm. Each reach began from a starting point 15 cm above and 40 cm in front of a central reference. The physical target was located either 0, 4, 8, 12, or 16 cm to the left of the central reference. For redirected reaches, the visual target was always located at the central reference. Redirection was performed via Haptic Retargeting as detailed by Cheng et al. [7], where the offset between the real and virtual hand is linearly

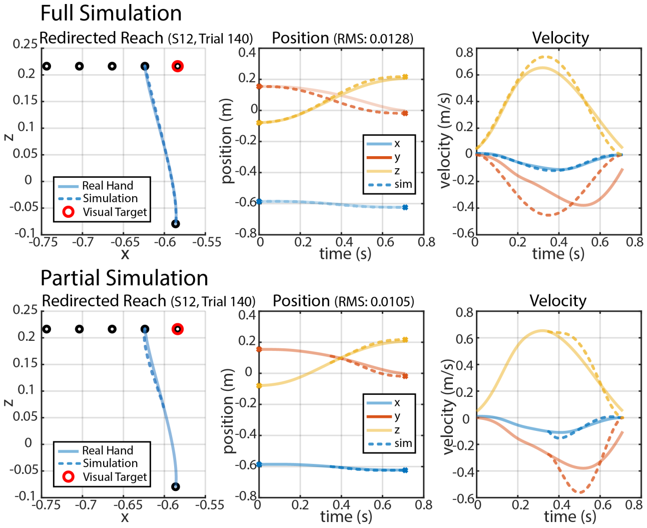

Figure 3: Example comparison of real and simulated redirected reach trajectories to a physical target that is offset 4 cm from the virtual target. The first row of graphs shows a simulated trajectory initialized with the hand state at t = 0. The second row shows a partial simulation of the same reach, initialized with the hand state at t = 0.5 × reach time.

increased to the target value (0 - 16 cm) as the hand approaches the physical target.

3.6.3 Simulation. We initialize our simulation with the starting position, velocity, and acceleration of the real hand and the position of the visually-presented target. The system is discretized with a timestep of 10 ms. At each timestep k, the remaining reach time is used to generate the appropriate $\widehat{A}_k$ and $\widehat{B}_k$ matrices and propagate the state forward in time. In these initial simulations, we assume the reach end time (included in the dataset) is known. To investigate our model's ability to simulate both full and partial reaches, we consider two initial simulation conditions per reach. In the Full condition, the simulation is initialized at the onset of the reach and the entire trajectory is simulated; in the Partial condition, the simulation is initialized at the temporal midpoint of the reach and the remaining half of the trajectory is simulated.

3.6.4 Metrics. For each trial in the dataset, we compute the root-mean square error (RMSE) between the simulated and experimental trajectory. To account for variation in reach time between trials, we resample each trajectory to 100 evenly spaced points between the simulation start and end time prior to computing RMSE.

3.6.5 Results. Figures 3 & 4 show representative examples of fully and partially simulated redirected reaches. The average real and simulated trajectories for all offsets are provided in Supplementary Material. Figure 5 presents the RMSE for fully and partially simulated reaches to each target, both redirected and non-redirected.

The simulation results suggest that our proposed model is effective at modeling the spatial trajectory of both redirected and non-redirected reaches. At a high level, the simulation produces

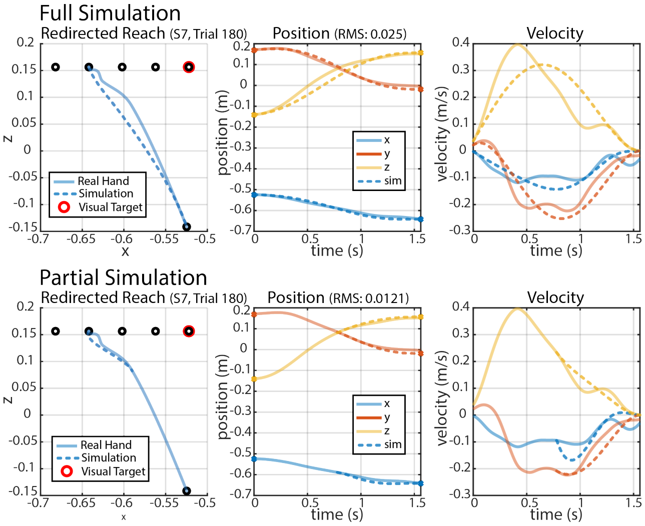

Figure 4: Example comparison of real and simulated redirected reach trajectories to a physical target that is offset 12 cm from the virtual target. The first row of graphs shows a simulated trajectory initialized with the hand state at t = 0. The second row shows a partial simulation of the same reach, initialized with the hand state at t = 0.5 × reach time.

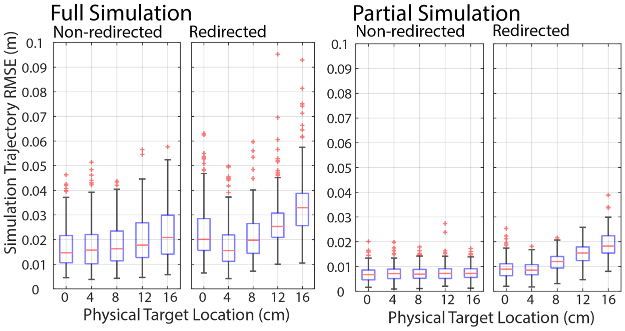

Figure 5: RMSE of fully (left) and partially (right) simulated trajectories for non-redirected and redirected reaches.

behavior that matches real reaches – the physical hand is redirected to the correct target, yielding S-shaped position trajectories and bell-shaped velocity trajectories – given only the hand's initial state and the end time of the reach. We use the reach end time given in the dataset (computed as the time the hand slows to a stop within 2 cm of the target).

For redirected reaches, we see an increase in the model error and its variance as the magnitude of redirection increases, indicating that there may still be features of redirected reaching not fully captured by our model. Importantly, however, we see that model accuracy tends to increase for partial reach simulation, where only the latter half of the reach is simulated. This is beneficial for applications which use models iteratively (such as MPC), as accuracy improves over time.

The increase in model error during redirected reaches may also arise from the task itself, in which users were repeatedly reaching under widely varying levels of redirection. Repeatedly changing users' spatial mapping in this way could have led to the adoption of modified reaching strategies which are captured less by the MJ model for goal-directed reach. Overall, however, the model generates plausible trajectories for redirected reaches.

4 REDIRECTION AS A MODEL-PREDICTIVE CONTROL PROBLEM

Up to this point, we have developed an approximate dynamic model for redirected reaching. While such a model is useful on its own for simulating reach under a given redirection algorithm, we are most interested in exploring how this dynamic model may be leveraged for the generation of novel redirection strategies. Specifically, in what ways might the knowledge of *how* the user may respond to a change in spatial warping be useful for the real-time generation of the warping? Towards answering this question, we explore posing redirection as a Model Predictive Control (MPC) problem.

4.1 Overview

In contrast with previous redirection strategies, the MPC approach aims to determine an optimal physical-to-virtual warping in a real-time, closed-loop fashion. Driving this process is the dynamic model of redirected reaching developed in Section 3.

MPC operates by minimizing a cost function over a receding horizon in order to find a set of optimal system states and control inputs, leveraging a specified dynamic model to inform the controller how potential control inputs may influence future system states. The optimal control input is then applied to the real system, and the process is repeated at each timestep.

In the case of reach redirection, this optimization process should balance two main objectives: (1) guide the user's hand to complete a desired redirection goal (e.g., arrive at a certain real-world target), and (2) ensure warping is applied smoothly and minimally. At a high level, this can be described by the following process, where $\mathbf{z}$ is the user's state and $\mathbf{w}$ is the commanded input:

$$\min_{\mathbf{z},\mathbf{w}} \sum C_{goal}(\mathbf{z},\mathbf{w}) + C_{warp}(\mathbf{z},\mathbf{w}) \tag{11}$$

Here, C_{goal} is the cost associated with straying from a given redirection goal and C_{warp} is the cost associated with warping the user's virtual hand from their real one.

The redirection goal describes the objective of a given redirection task. In this work, we explore two possible redirection goals. We refer to the first as endpoint-based redirection (such as Haptic Retargeting [3]), where the primary goal is for the user's physical hand to arrive at a desired point in space. We refer to the second as path-based redirection, a novel redirection case where the goal is for the user's physical hand to follow a desired path while reaching.

4.2 System Model

We leverage the dynamic model derived in Section 3 for our MPC formulation. Here, however, we redefine our state, control, and dynamics to be in the 3D form. We define the system state used in our MPC formulation ($\mathbf{z}$) as,

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}^x & \mathbf{z}^y & \mathbf{z}^z \end{bmatrix}^T \tag{12}$$

where $\mathbf{z}^x$, $\mathbf{z}^y$, $\mathbf{z}^z$ are defined according to Eq. 6 for the x, y, and z dimensions, respectively. This contains the position, velocity, and acceleration of the user's real hand, as well as the position of their reach target (and two steps of position & control history). We define the control variable $\mathbf{w}$ as the vector offset added to the physical hand to generate the virtual hand position at a given timestep:

$$\mathbf{w} = \begin{bmatrix} w^x & w^y & w^z \end{bmatrix}^T \tag{13}$$

We define the system dynamics as:

$$\mathbf{z}_{k+1} = \widehat{A}_k \mathbf{z}_k + \widehat{B}_k \mathbf{w}_k \tag{14}$$

where, without loss of generality, $\widehat{A}_k$ and $\widehat{B}_k$ are redefined as the block diagonal A and B matrices in Eq. 10.

4.3 Warping Costs

We first consider the costs associated with warping the user's virtual hand from their real hand. It is intuitive that the larger the offset $\mathbf{w}$ is between the real and virtual hand, the less tolerable and more noticeable the redirection will be [7, 52]. To encourage the magnitude of the applied warping at any time to be minimal, we consider the following cost:

$$C_w(\mathbf{w}) = \|\mathbf{w}\|^2 \tag{15}$$

which penalizes the magnitude of any applied offset. Furthermore, to encourage warping to be applied smoothly [55], we also penalize the change in applied offset from one timestep to the next:

$$C_{\Delta w}(\mathbf{z},\mathbf{w}) = \|\mathbf{w}_k - \mathbf{w}_{k-1}\|^2 \tag{16}$$

Additionally, we penalize the difference in velocity between the user's real and virtual hand. This penalty explicitly encourages the motion of the virtual hand to follow that of the real hand. As an extreme example, we want our controller to discourage the virtual hand from moving in the opposite direction of the real hand, as this would likely be frustrating to the user and detract from their sense of agency and embodiment in the virtual environment [29].

$$C_v(\mathbf{z},\mathbf{w}) = \ \mathbf{v}^v - \mathbf{v} \ ^2 \tag{17}$$

Here $\mathbf{v}$ and $\mathbf{v}^v$ are the velocity vectors of the real and virtual hands, respectively. While the real hand velocity is contained with the state $\mathbf{z}$, the virtual hand velocity is computed according to Eq. 7.

4.4 Redirection Goals

It remains to formulate an appropriate cost function that captures the redirection goal. That is, what is the desired result of warping the user's virtual hand? In the majority of prior work [3, 7, 19, 22], this goal is to bring the user's hand to a particular point in space as they reach towards a specific virtual object. We refer to this as an endpoint-based redirection goal. Applications of endpoint-based redirection range from enabling users to reach haptic props [7] to improving the ergonomics of a particular action [14, 37]. In this section, we formulate this goal as an optimization cost.

However, to demonstrate the generalizability of an MPC approach to reach redirection, we also consider a second redirection goal which, to the best of our knowledge, has not been previously

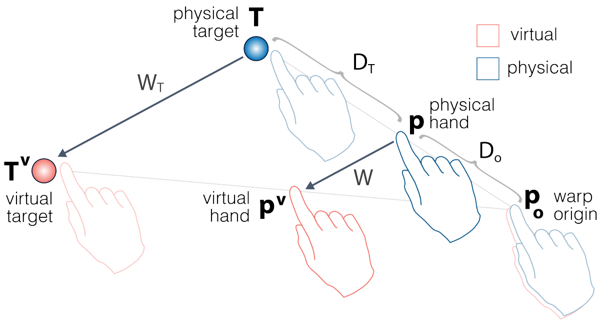

Figure 6: Diagram of endpoint-based redirection

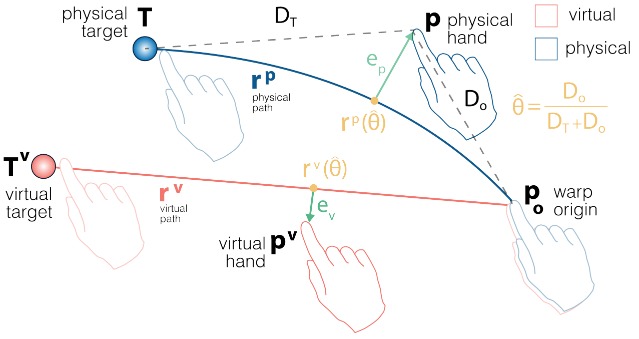

Figure 7: Diagram of path-based redirection

explored. In addition to endpoint-based redirection, we explore a cost formulation to enable path-based redirection, where the goal is to guide the user's hand along a particular path in space. If successful, such an approach could be useful in a variety of scenarios, such as to obstacle avoidance within the user's physical surroundings.

4.4.1 Endpoint-based Redirection. The goal of endpoint-based redirection is for the physical hand to arrive at a specific physical point, $\mathbf{T}$, when the virtual hand arrives at a specific virtual point, $\mathbf{T}_v$. Furthermore, at the point where redirection is initiated ($\mathbf{p}_o$), no warping should be present. This process is illustrated by Figure 6, where $\mathbf{p}$ is the position of the physical hand, and $\mathbf{p}_v$ is the position of the virtual hand.

The diagram demonstrates that we want to encourage the applied offset $\mathbf{w}$ to approach $\mathbf{w}_T = \mathbf{T}_v - \mathbf{T}$ as the physical hand approaches $\mathbf{T}$. Similarly, $\mathbf{w}$ should be encouraged to be near zero when the physical hand is near the warp origin $\mathbf{p}_o$. Let D_o be the current distance from $\mathbf{p}$ to $\mathbf{p}_o$, and let D_T be the current distance from $\mathbf{p}$ to $\mathbf{T}$. We can then formulate a cost which penalizes deviation from $\mathbf{w}_T$ more heavily when D_T is small, and similarly penalizes deviation from zero when D_o is small.

$$C_{re}(\mathbf{z}, \mathbf{w}) = \frac{1}{(D_T)^2} \|\mathbf{w} - \mathbf{w}_T\|^2 + \frac{1}{(D_o)^2} \|\mathbf{w}\|^2 \tag{18}$$

4.4.2 Path-based Redirection. For path-based redirection, the goal is to have the physical hand follow a desired path as the user is reaching towards a virtual object. Let $\mathbf{r}^p$ be the desired path of the physical hand, parameterized by $\theta \in [0, 1]$ (shown in Figure 7). Here θ represents progress along the path; $\mathbf{r}^p(0)$ indicates the path starting point, while $\mathbf{r}^p(1)$ indicates the endpoint. Similarly, let $\mathbf{r}^v$ be the straight-line path from the reach starting point to the virtual target, also parameterized by $\theta \in [0, 1]$. We can think of this as the expected path of the virtual hand as the user reaches towards the target.

Following a desired reach path requires minimizing the distance between the physical hand position $\mathbf{p}$ and the nearest point on $\mathbf{r}^p$ at any given time. However, because finding the nearest point on a curve is an optimization problem in itself, we find an approximate value for this point leveraging the progress term θ. We estimate the user's current progress along the path as,

$$\widehat{\theta} = \frac{D_o}{D_o + D_T} \tag{19}$$

where D_o is the distance from $\mathbf{p}$ to the reach starting point $\mathbf{p}_o$, and D_T is the distance from $\mathbf{p}$ to the desired reach path endpoint T. We therefore formulate the following cost to encourage the physical hand to follow $\mathbf{r}^p$.

$$C_{rp}(\mathbf{z}) = \quad \mathbf{p} - \mathbf{r}^p(\hat{\theta}) \quad^{2} \tag{20}$$

Finally, in order to ensure the virtual hand arrives at the virtual target, we include an additional similar cost on the virtual hand's deviation from $\mathbf{r}^v$.

$$C_{rv}(\mathbf{z}, \mathbf{w}) = \quad \mathbf{p}^v - \mathbf{r}^v(\hat{\theta}) \quad^{2} \tag{21}$$

Importantly, because the MPC approach operates continuously in real-time, the desired path $\mathbf{r}^p$ can be adjusted at each timestep. This enables redirection along a path that updates dynamically, for example, in response to changes in the user's physical environment (such as another user entering the space) or shifting uncertainty in the user's intended target. Furthermore, in order to simplify computation, $\mathbf{r}^p$ can be a local fit to a global reference trajectory.

4.5 Optimization

Combining each warping cost, the total warping stage cost is,

$$J_k^w = \alpha_w C_w(\mathbf{w}_k) + \alpha_{\Delta w} C_{\Delta w}(\mathbf{z}_k, \mathbf{w}_k) + \alpha_v C_v(\mathbf{z}_k, \mathbf{w}_k) \tag{22}$$

where the weights α_w, $\alpha_{\delta w}$, and $\alpha_v \geq 0$ govern the relative influence of each cost term. For the endpoint-based redirection case, the redirection goal cost is

$$J_k^r = \alpha_{re} C_{re}(\mathbf{z}_k, \mathbf{w}_k) \tag{23}$$

where $\alpha_{re} \geq 0$ governs the relative influence of the endpoint redirection goal. In the case of path-based redirection, the total redirection goal cost is,

$$J_k^r = \alpha_{rp} C_{rp}(\mathbf{z}_k) + \alpha_{rv} C_{rv}(\mathbf{z}_k, \mathbf{w}_k) \tag{24}$$

where α_{rp} and $\alpha_{rv} \geq 0$ govern the relative influence of the path redirection goal costs. We report the values of all weights used in our experiments in the following section.

The total cost is computed by summing each stage cost over N future timesteps. At each time t, a sequence of optimal inputs $\mathbf{w}$ is computed by solving an N-step finite horizon nonlinear optimization problem. Following each computation, the first offset in the optimal input sequence ($\mathbf{w}_0$) is applied to the real hand to determine the position of the virtual hand. The user then adjusts the motion of their hand in response, updating the system state $\mathbf{z}$. This process is iteratively repeated at each timestep, enabling modeling and approximation errors to be continually compensated.

The final optimization problem (in both the endpoint-based and path-based redirection cases) is then

$$\min_{\mathbf{z},\mathbf{w}} \sum_{k=1}^{N} J_k^r + J_k^w$$

Subject to:

$$\begin{aligned} \mathbf{z}_{k+1} &= \widehat{A}_k \mathbf{z}_k + \widehat{B}_k \mathbf{w}_k && \text{(state dynamics)} \\ \mathbf{z}_0 &= \mathbf{z}(t) && \text{(initial state)} \\ \mathbf{z}_k &\in Z && \text{(state constraints)} \\ \mathbf{w}_k &\in W && \text{(input constraints)} \end{aligned} \tag{25}$$

At each time t, the optimization problem is initialized based on the current state of the hand, which we assume is known. This assumption is reasonable as spatial tracking of controllers (and accordingly hands) is common in VR systems. Note that $\widehat{A}_k$ and $\widehat{B}_k$ can vary from one stage of the optimization problem to the next based on the estimated time remaining in the reach at timestep k – we describe our reach time approximation process in the following Implementation Section. Larger horizons tend to yield better performance but at the cost of longer computation times; we have found N=10 to be a suitable balance between performance and computation in our implementation. Constraints can also be used to build limits directly into the MPC redirection algorithm. For example, state constraints can represent the physical limits of the user's workspace, and input constraints can represent desired limits on the magnitude of applied offsets. In the present work, we do not implement any constraints on the state or control inputs.

5 IMPLEMENTATION

In this section we describe our real-time implementation of MPC-based reach redirection in VR. We implement two example controllers - one for endpoint-based redirection (which we refer to as MPC-E) and one for path-based redirection (MPC-P). We detail our iterative approximation for remaining reach time, the selected optimization weights used in our evaluations, and the software implementation of the MPC controllers.

5.1 Remaining Reach Time Approximation

The redirected reach model used in the MPC strategy requires an estimate of the remaining time in the reach to inform the time-varying dynamics ($\widehat{A}_k$, $\widehat{B}_k$). Importantly, because the optimization problem is solved at each timestep, we require only an approximation for the time remaining in the reach. To generate this approximation at each timestep, we assume the hand's velocity toward the reach target is constant. We first smooth the velocity vector of the real hand using a 2nd order Savitzky-Golay filter and then project it in the direction of the target. The remaining reach time R is then approximated as,

$$R = D_T / \|\mathbf{v}_T\| \tag{26}$$

where D_T is the hand's distance to the physical target, and $\mathbf{v}_T$ is the smoothed velocity vector projected in the direction of the target T.

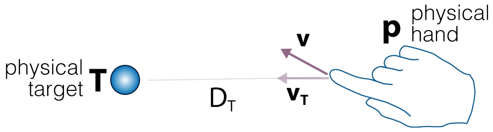

Figure 8: Remaining reach time is approximated as distance-to-target divided by the hand's speed in the target direction.

5.2 Software & Hardware

Our implementation is run in Unity 2017.3.1f1 on a laptop PC (Intel Core i7-7700 CPU @ 2.8 GHz). Optimization solvers for MPC-E and MPC-P were generated using ForcesPro [1], which creates C code for each solver. A C# wrapper allows the generated solvers to be used in Unity. Both solvers accept the optimization weights, physical target position, virtual target position, and reach origin as input parameters; additionally, MPC-P accepts the desired physical path as an input. The optimization weights used in our experiments are listed in Table 1 and were empirically selected by balancing user preference and MPC performance in pilot studies. Note that α_v was not found to influence MPC-E, and so was set to 0 for efficiency. In contrast, this parameter had a significant effect on MPC-P, so we explore two potential weight values in this case:

Table 1: Optimization Weights

	α_{re}	α_{rp}	α_{rv}	α_w	$\alpha_{\Delta w}$	α_v
MPC-E	1	-	-	0.001	1	0
MPC-P	-	1	0.1	0.001	1	0.1/0.05

Because the MPC must run in real-time, it is critical that the solver is fast and efficient. The mean solve time for an MPC-E problem instance is 2.5 ± 1.2 ms, while the mean solve time for MPC-P is 5.3 ± 1.0 ms. We found these values to be consistent across different users and reaches. Our algorithm runs in a Unity thread with a 10 ms loop time – no trials in our experiments led to solve times exceeding 10 ms.

We use an HTC Vive Pro Eye head-mounted display (HMD) to render our VR environment. The HMD and two standard Vive controllers are tracked by two IR base stations. In the current implementation, users hold the controller in their hand to enable tracking of hand motion.

6 EVALUATION

We designed a two part VR target acquisition study to evaluate the effectiveness of MPC compared to existing redirection techniques. In the Endpoint Study, we compare our MPC-E against standard Haptic Retargeting (HR) as implemented by Cheng et al. [7], where users reach to a physical target that is offset by a specified angle. In the Path Study, we investigate desired reach paths of different curvatures; here we compare our MPC-P to the thin-plate spline warping (TPS) technique used by Kohli et al. in *Redirected Touching* [31]; TPS smoothly interpolates between pairs of fixed "landmarks" in physical and virtual space [11]. The impact of each of these methods were compared using quantitative reach characteristics as well as self-reported user experience.

6.1 Participants

We recruited 10 right-handed participants to complete the study; however, 2 participants were removed due to data collection issues. Thus, we report on 8 participants (4 F, 4 M) between ages 19-28 (μ = 22, σ = 2.9). Participants received a $15 gift card for their time.

6.2 Conditions

6.2.1 Endpoint Study. This within-subjects study has two independent variables: warp type (MPC-E vs. HR) and angle offset. For each reach, the physical target is offset from the virtual target by either 0°, 4°, 8°, or 16°(Figure 9b). We selected these levels to be below (0°, 4°), slightly above (8°), and significantly above (16°) the expected detection threshold of 4.5°as reported by Zenner et al. [52]. This results in a 2 × 4 study design.

6.2.2 Path Study. This study compares MPC-P against TPS for redirecting the user’s hand along a desired path. We also investigate the impact of α_v on the performance of MPC-P, as we hypothesize this weight balances the trade-off between path-following and detection of the warp. From pilot testing, we selected to compare α_v values of 0.05 and 0.1. Therefore we investigate three warp types in Path Study (MPC-P$_{0.05}$, MPC-P$_{0.1}$, and TPS) and four offsets for a 3 × 4 within-subjects study design.

In these trials, the physical target always coincides with the virtual target, while the curvature of the desired path is varied. Four paths are tested, with constant normalized curvatures of 0, 0.2, 0.4, and 0.6 (Figure 9b). A normalized curvature of 0 indicates a straight line, while 1 would indicate a semicircle. In TPS trials, 30 evenly-spaced points on the desired physical trajectory are used as physical landmarks, while 30 evenly-spaced points on the straight line between the origin and target are used as virtual landmarks.

6.3 Measures

For each trial, we quantify reach performance by the error from the reach endpoint to the physical target, total reach time, reach smoothness, and root mean squared error (RMSE) between the measured and desired trajectory (for curved path conditions only).

We also solicit ratings about how noticeable the warping was for each reach. Additionally, for each warp type we record user ratings about overall noticeability, how tolerable the warping was, and the perceived task difficulty.

6.4 Procedure

Before beginning the study, we explained the experimental protocol and received each participant’s consent to be in the study. Participants were seated and wore a VR headset (HTC Vive Pro Eye) while holding two HTC Vive VR controllers. In the virtual scene, a starting zone (orange sphere, 3 cm dia.) and reach target (2 cm dia.) were positioned in front of the participant. Following the setup of Zenner et al. [52], the starting zone was positioned 20 cm below the user’s headset and 20 cm away from the body, while the target was positioned 40 cm out from the starting zone (Figure 9a). A small red sphere (1 cm dia.) was displayed to represent the participant’s virtual hand.

The overall study was organized into two sections, Endpoint Study and Path Study, counterbalanced between participants. Each section was divided into blocks (grouped by warp type) presented in randomized order. Endpoint Study had 2 blocks (MPC-E, HR), while Path Study had 3 blocks (MPC-P$_{0.1}$, MPC-P$_{0.05}$, TPS). Each block (i.e. warp type) consisted of 16 randomly presented reach trials (4 offset levels × 4 repetitions).

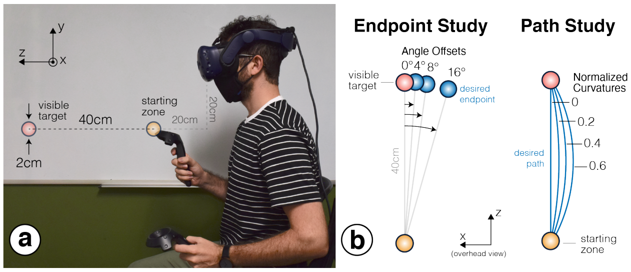

Figure 9: (a) Experimental setup and virtual target layout for study. Participants wore a VR headset and reached the controller from the orange starting zone to a red target. (b) Redirection offsets tested in Endpoint and Path Studies.

At the start of each trial, participants were asked to reach from the starting zone to the target as accurately as possible with their right hand. Note that no time limit or timing instruction was given. After each reach, participants were asked to respond to this statement on a 5-point Likert scale:

Q1. *The movement of the red sphere matched my real hand.*
(0 = Strongly Disagree, 4 = Strongly Agree)

Upon completion of a block (one complete warp type), participants were asked to rate their agreement (via 5-point Likert scale) with three additional statements about their overall experience:

Q1. *Overall, the movement of the red sphere matched my real hand.*
(0 = Strongly Disagree, 4 = Strongly Agree)

Q2. *To what extent were any differences in the movement of the red sphere and your real hand tolerable.*
(0 = Very Intolerable, 4 = Very Tolerable)

Q3. *Rate the overall difficulty of completing this block of tasks.*
(0 = Very Difficult, 4 = Very Easy)

7 RESULTS

Across 8 participants, 640 trials were recorded (Endpoint Study: 256, Path Study: 384). In order to determine the effects of warp type and magnitude, we fit a linear mixed effects model to the data for each dependent measure. For per-trial measures (endpoint-error, reach time, RMSE, smoothness, per-trial reported mismatch), warp type and offset magnitude were set as fixed effects with an interaction effect between them. For per-block measures (overall reported mismatch, tolerability, and task difficulty/ease), warp type was the only fixed effect. In all models, participant was included as a random effect.

An ANOVA was used to determine the significant factors within each model. For significant parameters ($p < 0.05$), an additional Bonferroni-corrected post-hoc test was carried out in order to determine pairwise significance – for space, these are reported via

Table 2: Endpoint Error Effects

Study	Factor	df_{Num}	df_{Den}	F	p	η_p^2
Endpoint	Warp	1	241	0.46	0.496	0.00
Endpoint	Offset	3	241	3.90	**0.009**	0.05
Endpoint	W*O	3	241	2.23	0.085	0.03
Path	Warp	2	365	5.85	**0.003**	0.03
Path	Offset	3	365	13.10	**<.0001**	0.10
Path	W*O	6	365	1.31	0.252	0.02

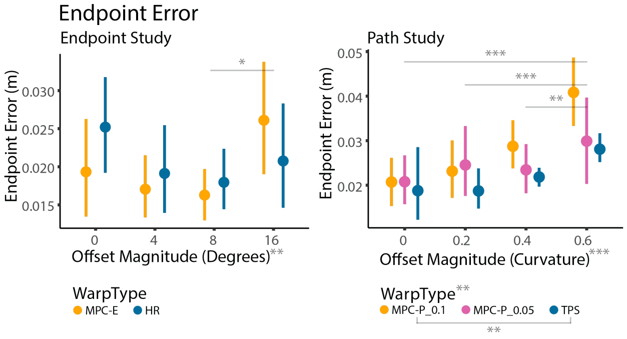

Figure 10: Endpoint error across different offset magnitudes and warping types. Means and 95% CI shown.

Table 3: Reach Time Effects

Study	Factor	df_{Num}	df_{Den}	F	p	η_p^2
Endpoint	Warp	1	241	4.32	**0.039**	0.02
Endpoint	Offset	3	241	3.18	**0.025**	0.04
Endpoint	W*O	3	241	0.76	0.515	0.01
Path	Warp	2	365	23.32	**<.0001**	0.11
Path	Offset	3	365	8.76	**<.0001**	0.07
Path	W*O	6	365	1.05	0.085	0.02

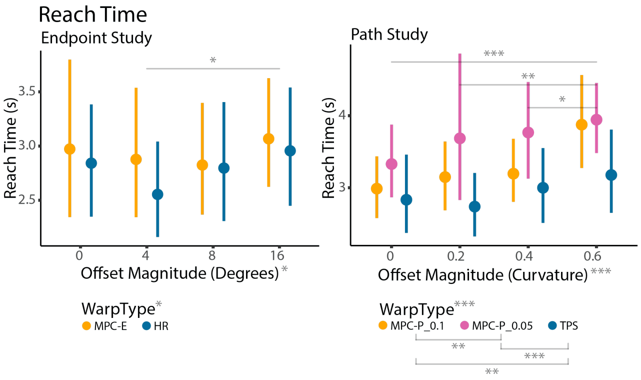

Figure 11: Reach time across different offset magnitudes and warping types. Means and 95% CI shown.

Table 4: RMSE Effects

Study	Factor	df_{Num}	df_{Den}	F	p	η_p^2
Path	Warp	2	372	5.80	**0.003**	0.03
Path	Offset	3	372	2.77	**0.041**	0.02
Path	W*O	6	372	2.37	0.227	0.02

significance markings on the figures (*: p < 0.05, **: p < 0.01, ***: p < 0.001). The results are shown visually in Figures 10 - 17. Complete statistical results for all pairwise contrasts are provided in Supplementary Material.

7.1 Endpoint Error

During each trial, the main goal was to reach the target as accurately as possible; we assess this by computing the displacement magnitude between their hand's final position and the target. In Endpoint Study, warp type did not significantly impact endpoint error (Figure 10); however there was a main effect of offset magnitude (Table 2), with the difference arising between 8°and 16°.

In Path Study (Figure 10), both warping type and offset magnitude significantly affect the endpoint error (Table 2). The significant differences in the warp type were between the MPC methods, with MPC-P$_{0.1}$ resulting in higher endpoint errors. Additionally, endpoint error significantly increased with offset magnitude.

7.2 Reach Time

We also examine the total time taken to complete each reach (Figure 11). In both studies, warping type and offset magnitude were significant factors (Table 3). The significantly different pair in Endpoint Study was between 4°and 16°of offset. In Path Study, all three warp types differed significantly, with MPC-P$_{0.05}$ having the longest reach times and TPS having the shortest. Increasing path curvature also significantly increased reach time.

7.3 RMSE

To see how well the different warp types encourage users to follow a particular curved trajectory (Path Study), we calculated the root mean squared error (RMSE) between the measured and desired trajectory. There were significant effects of both warp type and offset magnitude on RMSE (Table 4). Within warp type, TPS resulted in lower RMSE values compared to each of the MPC-P methods (0.1 and 0.05). RMSE values tended to increase with offset magnitude.

7.4 Smoothness

Beyond task performance, we were also interested in the smoothness of the user's reach. We compute the dimensionless squared jerk (DSJ) of each reach, which has been shown to properly characterize smooth and jerky motions [27]. DSJ was calculated as follows, where $D = t_0 - t_f$ is the reach duration and v_{mean} is the mean reach velocity:

$$\left(\int_{t_o}^{t_f} \ddot{x}(t)^2 \, dt\right) * \left(D^3/v_{mean}^2\right) \tag{27}$$

In Endpoint Study, there were no significant differences between warp type or offset magnitude, however a significant interaction effect was found (Table 5) – this is highlighted between the warp types in the 8°condition (Figure 13). In Path Study, both warp type and offset magnitude were significant factors (Table 5). Within warp type, there were significant differences between both MPC-P

Table 5: DSJ Main Effects

Study	Factor	df_{Num}	df_{Den}	F	p	η_p^2
Endpoint	Warp	1	239	0.13	0.723	0.00
Endpoint	Offset	3	239	2.28	0.080	0.03
Endpoint	W*O	3	239	3.44	**0.018**	0.04
Path	Warp	2	364	7.19	**<.0001**	0.04
Path	Offset	3	364	7.23	**<.0001**	0.06
Path	W*O	6	364	0.88	0.510	0.01

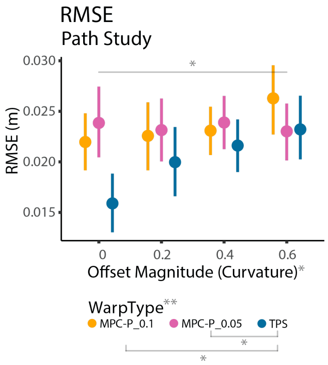

Figure 12: Root mean squared error (RMSE) between real and desired path. Means and 95% CI shown.

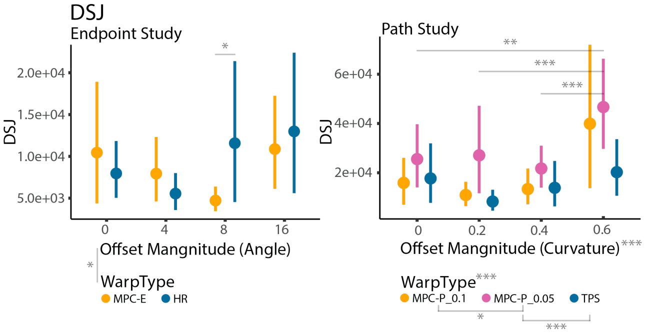

Figure 13: Dimensionless squared jerk (DSJ) across offset magnitude and warp type. Means and 95% CI shown.

methods, with the jerkiest user hand movements occurring during MPC-$P_{0.05}$ trials.

7.5 Qualitative

7.5.1 Per Trial Measures. After each reach, users were asked to rate how well their physical and virtual hands matched.

For Endpoint Study, only offset magnitude made a significant impact on the score (Table 6); larger magnitude yielded lower scores (Figure 14). In Path Study, both warp type and offset magnitude were significant factors, as well as their interaction (Table 6). Figure 14 highlights the relevant interactions – in which TPS performs better than both MPC-P methods at 0 and 0.2 curvatures, better than MPC-$P_{0.05}$ at 0.4, and no different at 0.6. MPC-$P_{0.1}$ also received higher hand match scores compared to MPC-$P_{0.05}$ in all conditions except 0.6.

Table 6: Trial Hand Match Effects

Study	Factor	df_{Num}	df_{Den}	F	p	η_p^2
Endpoint	Warp	1	241	1.31	0.254	0.01
Endpoint	Offset	3	241	39.43	**<.0001**	0.33
Endpoint	W*O	3	241	1.35	0.260	0.02
Path	Warp	2	365	45.35	**<.0001**	0.20
Path	Offset	3	365	118.77	**<.0001**	0.49
Path	W*O	6	365	3.22	**0.004**	0.05

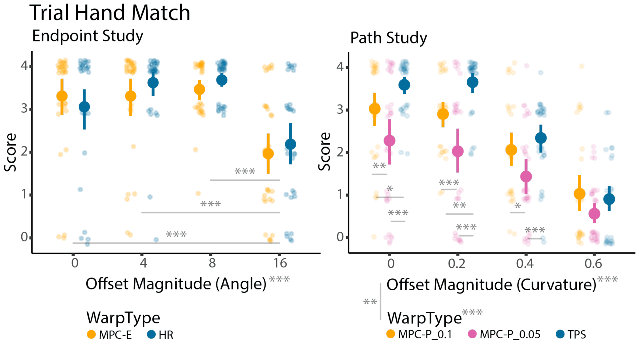

Figure 14: Self-reported hand match ratings per trial across offset magnitude and warp type. Means and 95% CI shown.

7.5.2 Per Block Measures. All trials of a given warp type were grouped into a single block, after which participants were asked to respond to three questions outlined in Section 6.4.

Q1 considered how well the virtual hand movement matched overall. In Endpoint Study, the differences in scoring between HR and MPC-E were not significant; however, in Path Study, warping type was significant (Table 7). There were marginally significant differences between MPC-$P_{0.05}$ and MPC-$P_{0.01}$ (Figure 15). MPC-$P_{0.01}$ performed similarly to TPS, with MPC-$P_{0.05}$ receiving the lowest scores.

Q2 asked participants to rate how tolerable any mismatch was to their experience. There was no significant difference between warping types in Endpoint Study. But, in Path Study there was a significant difference (Table 7), specifically between MPC-$P_{0.05}$ and TPS with TPS receiving higher ratings of tolerability.

Q3 examined the self-reported ease with which participants could complete the block. In Endpoint Study, there was a significant difference in warping type (Table 7), where HR was rated as easier for task completion compared to MPC-E. In Path Study, however, there were no significant differences between the three warping methods (Table 7).

7.6 Visualizing Applied Warping

To provide an intuition about the warping generated by each method, we illustrate the average magnitude of applied warping (i.e. $||\mathbf{w}||$) as a function of progress along the reach. Figure 16 shows this across various offset magnitudes. We see strong similarity in the warpings

Table 7: Block Questions Main Effects

Q	Study	Factor	df_{Num}	df_{Den}	F	p	η_p^2
1	Endpoint	Warp	1	7	0.13	0.732	0.02
1	Path	Warp	2	14	4.83	**0.025**	0.39
2	Endpoint	Warp	1	7	0.00	1.000	0.00
2	Path	Warp	2	14	5.65	**0.016**	0.43
3	Endpoint	Warp	1	7	7.00	**0.033**	0.44
3	Path	Warp	2	14	2.24	0.144	0.22

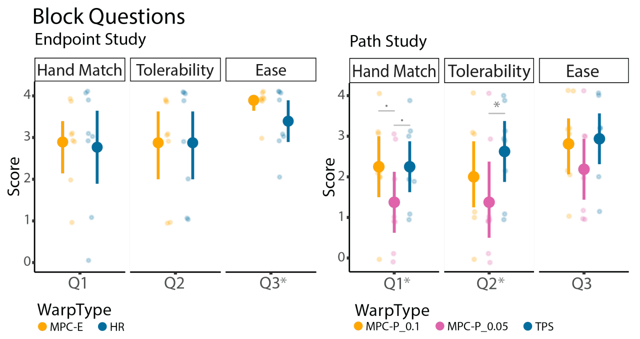

Figure 15: Self-reported subjective ratings per block. Means and 95% CI shown.

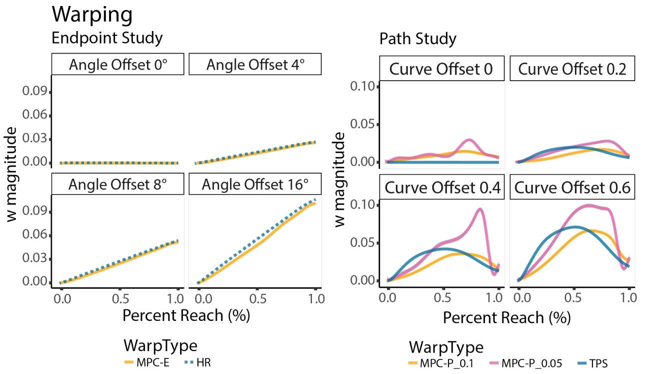

Figure 16: Mean redirection magnitude vs. proportion of reach completed, grouped by offset magnitude and averaged across all participants.

applied by MPC-E and HR, supporting the lack of many significant differences found between the two in our results. In Path Study, the effect of MPC is much clearer. In both MPC-P cases, we see that less warp tends to be applied at the onset of reach compared to TPS, due to the cost of rapidly shifting the virtual hand. Compared to MPC-P$_{0.1}$, the lower penalty of MPC-P$_{0.05}$ yields warping that is applied more rapidly but with less stability.

Additionally, Figure 17 shows an overhead view of the average reach trajectory performed by each participant within the Path Study conditions. This highlights the strong between-user variability in the performance of any given warp (even with fixed warps such as TPS), further motivating the ability to fine-tune a warping method.

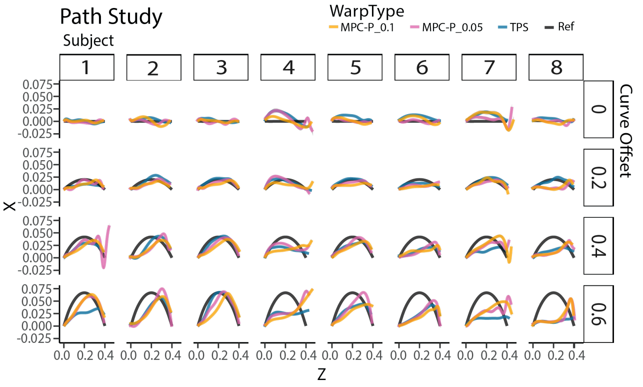

Figure 17: Mean reach trajectories for each participant (col.) and curve offset (row) in Path Study, in horizontal (xz) plane.

8 DISCUSSION AND LIMITATIONS

Our evaluation suggests that for endpoint-based redirection, MPC-E broadly matched the performance of Haptic Retargeting (HR) in terms of endpoint error, trajectory smoothness, and users' qualitative ratings. This is supported by the similarity and consistency of warping generated by the two techniques (Figure 17). We note that for 16°offsets, redirection was significantly more noticeable (lower Hand Match rating) than for 0°-12°offsets, and resulted in significantly larger endpoint error (Figure 10).

For path-based redirection, Thin Plate Spine warping (TPS) tended to perform significantly better than MPC-P$_{0.05}$ by the same measures, as well as in terms of path RMSE. In contrast, no significant differences were found between TPS and MPC-P$_{0.1}$ in terms of smoothness, RMSE, or endpoint error. However, TPS yielded significantly less noticeable warping than both MPC-P cases, particularly at lower curvatures. This important finding suggests that MPC-P as implemented may produce noticeable warping even when the desired redirection offset is small; TPS does not suffer from this issue since the warp is statically determined by the desired path. Per block, however, no significant differences were found between TPS and MPC-P$_{0.1}$; this suggests that overall, user's subjective experience may be comparable among these warp types.

While MPC-P$_{0.1}$ tended to outperform MPC-P$_{0.05}$ in most metrics, MPC-P$_{0.05}$ yielded lower endpoint error and RMSE for larger curvature offsets at the expense of increased reach time and lower self-reported measures. This suggests that there is a trade-off between the amount of warp applied and the user's comfort and cognitive load, which prior redirected reaching studies have found [32]. However, the MPC-P$_{0.05}$ controller may also have suffered from instabilities for large curvature offsets.

On the whole, we see a number of trade-offs between the more traditional redirection techniques and MPC approaches. As discussed, there are performance differences, though less so for endpoint-based redirection and for lower amounts of warping. MPC approaches have a higher computational cost, but are more generalizable and can be modified for different redirection tasks. Additionally, the MPC approach is solved continuously, which means that the goals can be easily updated in real-time; reach targets, cost weights, and other parameters can be updated from one timestep to the next.

This may make it easier to support real-world reaching tasks, where users can change their target goal on the fly. Beyond target changes, other costs can be added - for example to prevent the user from entering into a region or helping them by avoiding a moving obstacle. Perhaps most interestingly, we see the costs and constraints of MPC as a natural way to incorporate the findings from psychophysical studies of redirection (e.g. detection thresholds) directly into the generation of the warping itself. Future work should explore other such applications to better understand how an MPC approach could benefit user reach in VR.

While this work presents an important step towards more generalizable redirection techniques, there are several important limitations to consider:

8.0.1 Study Limitations. In order to maintain a practical study length, we evaluated a limited range of offsets and redirection configurations. Assessing the impact of vertical gain redirection [52], different virtual target locations, and combining angular and curvature offsets would give a broader picture of the capabilities of our MPC approach. Furthermore, the given study task was to reach the target as accurately as possible, with no instructions given about reach timing. Given the known speed-accuracy trade-off in human movement [15], this likely resulted in reduced movement speeds which may not have been captured by the present model. Because this was not tested against alternative instruction (e.g., reach the target as quickly as possible), further work is needed to explore which interaction condition MPC performs best in.

Path Study aimed to guide the user's hand along a path during reach, but their task goal remained simply to reach the end target accurately. Our results (Figure 17) suggest strong variation in the effectiveness of path-based redirection across participants. This may be because curvature offsets did not directly impact task-relevant error [50], as evidenced by Figure 10 where no significant differences in target endpoint error occur between curvatures of 0, 0.2, or 0.4 across all warp types. An interaction task where the user is instructed to follow a virtual path with their hand would likely result in the physical hand following the desired path more closely, since curvature offsets would have a stronger impact on the user's task goal. This interaction is less common than reaching for an object, however, and furthermore is not captured by the point-to-point goal directed movement models considered here.

Furthermore, our interpretation of Path Study is limited by the fact that curvature offset and path-based redirection have not been greatly explored to date, so less is known about their relevant perceptual thresholds. Additionally, our redirected reach model was only validated on linear reach due to the available dataset. Future work should better explore path redirection and validate the redirected reach model for this type of redirection task.

8.0.2 Model Limitations. The simplicity of the sensorimotor redirection model used also presents important limitations. While the Minimum Jerk model provides a convenient closed-form feedback law, it does not account for variation between users, motor and sensory noise, or the potential for different task-based reward functions. More comprehensive approaches such as stochastic [47] or intermittent [6, 9] optimal control present a more complete representation of the sensorimotor process and could perhaps be integrated within future MPC-based redirection iterations. However, it is not yet clear whether such complex optimization-based reach models can be efficiently integrated within a larger MPC framework.

Additionally, the present model considers only visual feedback while in reality the sensory system integrates multiple cues (e.g., proprioception) to estimate the position and movement of the hand in space [46]. Furthermore, recent work has highlighted the integration of hand velocity and orientation when estimating contact with a target in VR [10]. By expanding our model to handle multisensory integration through a Kalman filter [8] or maximum likelihood estimation [12], we may better capture the user's internal state estimates and thus improve model accuracy and MPC performance. Another limitation is the requirement of a reach time estimate. While estimation errors are partially mitigated as the controller recomputes the time estimate in each timestep, overall, it creates issues with inconsistent reaches. Rather than the simplified constant speed assumption used presently to roughly estimate remaining reach time, other methods based on the task itself (such as Fitts's Law[15]) may provide more accurate estimates without introducing too much complexity. Other reach models such as the Minimum Jerk-Minimum Time model [24] or Minimum Variance model [23] do not require the explicit assumption of reach time, but are comparatively more complex and require a different set of biological assumptions.

8.0.3 MPC Limitations. Overall, our MPC approach yielded comparable performance to HR in the endpoint condition and slightly lower performance than TPS in the path condition, particularly in terms of noticeability. The main source of this difference lies in the inherent variability of the warping produced by MPC. In HR and TPS, warping is determined entirely by the hand's position in space. With MPC, the optimization process balances redirection goals (e.g. desired spatial offsets) and warping costs, so the applied warping depends on previous states in addition to the current hand position. The smoothness of the warping applied (i.e., its rate of change over time/space) is then dependent on the costs and parameters selected, whereas the smoothness of traditional techniques is determined entirely by the distance between fixed physical and virtual landmarks. The selection of appropriate cost function parameters is critical for proper performance of an MPC approach, and the increased noticeability of MPC-P suggests additional parameter tuning may be needed to improve performance.

While this inherent variability may make MPC more challenging initially (as evidenced by our Path Study results), in theory it also enables continuous real-time adaptability of warping in response to shifting interaction goals (e.g. updating confidence in the user's target or avoiding dynamic physical obstacles). In contrast, the fixed spatial interpolation methods of traditional redirection techniques do not have a direct method of adapting to the user and their environment, meaning they must be updated ad hoc [7].

The computational cost of our present MPC approach also limits the time horizon on which we can evaluate potential future steps; it is possible that planning further ahead may improve performance. Thus, while the specific implementation of our MPC controllers are not perfect, we are optimistic about the potential benefits of this MPC approach from a high level - improved models and optimization costs could easily be fit into this overall framework.

9 CONCLUSION

In this paper, we introduced a Model Predictive Control approach for reach redirection in VR. We first adapted a Minimum Jerk reach model to capture redirection by incorporating an appropriate sensory bias in the user's estimate of their hand state. Our simulations show that this model predicts the trajectory of redirected reaches with similar errors compared to non-offset reaching, though errors increase with greater redirection, especially beyond known perceptual thresholds. We then applied this model as part of a Model Predictive Control framework for generating online redirected reaching. We described the formulation of two different cost functions for two reach redirection goals, endpoint-based and path-based redirection. Our evaluation results show that our MPC approach preforms well compared to traditional redirection methods for endpoint-based redirection, though it performs worse in path-based redirection tasks. We believe the MPC approach can have benefits beyond the demonstrated goals, especially in scenarios where reaches need to be adapted on the fly. Future work should explore other models and cost functions towards improved performance. However, these initial results suggest that control theoretic approaches to modeling and controlling reach redirection have promise.

ACKNOWLEDGMENTS

The authors would like to thank Thomas Langerak for helpful discussions on the implementation of our MPC solvers. This work was supported by the Stanford DARE Fellowship (E.J.G.), the Stanford Graduate Fellowship (E.D.Z.C), and the Alfred P. Sloan Research Fellowship grant no. FG-2021-15851 (S.F.).

REFERENCES

[1] [n. d.]. ForcesPro. https://www.embotech.com/products/forcespro/overview/

[2] Parastoo Abtahi and Sean Follmer. 2018. *Visuo-Haptic Illusions for Improving the Perceived Performance of Shape Displays*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3173574.3173724

[3] Mahdi Azmandian, Mark Hancock, Hrvoje Benko, Eyal Ofek, and Andrew D. Wilson. 2016. *Haptic Retargeting: Dynamic Repurposing of Passive Haptics for Enhanced Virtual Reality Experiences*. Association for Computing Machinery, New York, NY, USA, 1968–1979. https://doi.org/10.1145/2858036.2858226

[4] Myroslav Bachynskyi and Jörg Müller. 2020. Dynamics of Aimed Mid-air Movements. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–12.

[5] Brett Benda, Shaghayegh Esmaeili, and Eric D Ragan. 2020. Determining detection thresholds for fixed positional offsets for virtual hand remapping in virtual reality. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 269–278.

[6] Robin T Bye and Peter D Neilson. 2008. The BUMP model of response planning: Variable horizon predictive control accounts for the speed–accuracy tradeoffs and velocity profiles of aimed movement. *Human movement science* 27, 5 (2008), 771–798.

[7] Lung-Pan Cheng, Eyal Ofek, Christian Holz, Hrvoje Benko, and Andrew D. Wilson. 2017. *Sparse Haptic Proxy: Touch Feedback in Virtual Environments Using a General Passive Prop*. Association for Computing Machinery, New York, NY, USA, 3718–3728. https://doi.org/10.1145/3025453.3025753

[8] Frédéric Crevecoeur, Douglas P Munoz, and Stephen H Scott. 2016. Dynamic multisensory integration: somatosensory speed trumps visual accuracy during feedback control. *Journal of Neuroscience* 36, 33 (2016), 8598–8611.

[9] Seungwon Do, Minsuk Chang, and Byungjoo Lee. 2021. A Simulation Model of Intermittently Controlled Point-and-Click Behaviour. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–17.

[10] Seungwon Do and Byungjoo Lee. 2020. Improving Reliability of Virtual Collision Responses: A Cue Integration Technique. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–12.

[11] David Eberly. 2020. Thin-Plate Splines. https://www.geometrictools.com/Documentation/ThinPlateSplines.pdf. Accessed: 2021-09-04.

[12] Marc O Ernst and Martin S Banks. 2002. Humans integrate visual and haptic information in a statistically optimal fashion. *Nature* 415, 6870 (2002), 429–433.

[13] Shaghayegh Esmaeili, Brett Benda, and Eric D Ragan. 2020. Detection of scaled hand interactions in virtual reality: The effects of motion direction and task complexity. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 453–462.

[14] Tiare Feuchtner and Jörg Müller. 2018. Ownershift: Facilitating Overhead Interaction in Virtual Reality with an Ownership-Preserving Hand Space Shift. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) *(UIST '18)*. Association for Computing Machinery, New York, NY, USA, 31–43. https://doi.org/10.1145/3242587.3242594

[15] Paul M Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology* 47, 6 (1954), 381.

[16] Tamar Flash and Neville Hogan. 1985. The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of Neuroscience* 5, 7 (1985), 1688–1703. https://doi.org/10.1523/JNEUROSCI.05-07-01688.1985 arXiv:https://www.jneurosci.org/content/5/7/1688.full.pdf

[17] Carlos E Garcia, David M Prett, and Manfred Morari. 1989. Model predictive control: Theory and practice—A survey. *Automatica* 25, 3 (1989), 335–348.

[18] James J Gibson. 1933. Adaptation, after-effect and contrast in the perception of curved lines. *Journal of experimental psychology* 16, 1 (1933), 1.

[19] Eric J Gonzalez, Parastoo Abtahi, and Sean Follmer. 2019. Evaluating the Minimum Jerk Motion Model for Redirected Reach in Virtual Reality. In *The Adjunct Publication of the 32nd Annual ACM Symposium on User Interface Software and Technology*. Association for Computing Machinery, New York, NY, USA, 4–6.

[20] Eric J Gonzalez, Parastoo Abtahi, and Sean Follmer. 2020. Reach+ extending the reachability of encountered-type haptics devices through dynamic redirection in vr. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 236–248.

[21] Eric J Gonzalez and Sean Follmer. 2019. Investigating the detection of bimanual haptic retargeting in virtual reality. In *25th ACM Symposium on Virtual Reality Software and Technology*. 1–5.

[22] Dustin T Han, Mohamed Suhail, and Eric D Ragan. 2018. Evaluating remapped physical reach for hand interactions with passive haptics in virtual reality. *IEEE transactions on visualization and computer graphics* 24, 4 (2018), 1467–1476.

[23] Christopher M Harris and Daniel M Wolpert. 1998. Signal-dependent noise determines motor planning. *Nature* 394, 6695 (1998), 780–784.

[24] Bruce Hoff. 1994. A model of duration in normal and perturbed reaching movement. *Biological Cybernetics* 71, 6 (1994), 481–488.

[25] Bruce Hoff and Michael A Arbib. 1993. Models of trajectory formation and temporal interaction of reach and grasp. *Journal of motor behavior* 25, 3 (1993), 175–192.

[26] Neville Hogan. 1984. An organizing principle for a class of voluntary movements. *Journal of neuroscience* 4, 11 (1984), 2745–2754.

[27] Neville Hogan and Dagmar Sternad. 2009. Sensitivity of smoothness measures to movement duration, amplitude, and arrests. *Journal of motor behavior* 41, 6 (2009), 529–534.

[28] Mitsuo Kawato. 1996. Trajectory formation in arm movements: minimization principles and procedures. *Advances in motor learning and control* (1996), 225–259.

[29] Konstantina Kilteni, Raphaela Groten, and Mel Slater. 2012. The sense of embodiment in virtual reality. *Presence: Teleoperators and Virtual Environments* 21, 4 (2012), 373–387.

[30] Luv Kohli. 2010. Redirected touching: Warping space to remap passive haptics. In *2010 IEEE Symposium on 3D User Interfaces (3DUI)*. 129–130. https://doi.org/10.1109/3DUI.2010.5444703

[31] Luv Kohli. 2013. *Redirected touching*. Ph. D. Dissertation. The University of North Carolina at Chapel Hill.

[32] Luv Kohli, M. Whitton, and F. Brooks. 2012. Redirected touching: The effect of warping space on task performance. *2012 IEEE Symposium on 3D User Interfaces (3DUI)* (2012), 105–112.

[33] Julian Kreimeier, Sebastian Hammer, Daniel Friedmann, Pascal Karg, Clemens Bühner, Lukas Bankel, and Timo Götzelmann. 2019. Evaluation of different types of haptic feedback influencing the task-based presence and performance in virtual reality. In *Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments*. 289–298.

[34] Thomas Langerak, Juan José Zárate, Velko Vechev, David Lindlbauer, Daniele Panozzo, and Otmar Hilliges. 2020. Optimal control for electromagnetic haptic guidance systems. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 951–965.

[35] Flavien Lebrun, Sinan Haliyo, and Gilles Bailly. 2021. A Trajectory Model for Desktop-Scale Hand Redirection in Virtual Reality. In *IFIP Conference on Human-Computer Interaction*. Springer, 105–124.

[36] Mathew Monfort, Anqi Liu, and Brian Ziebart. 2015. Intent prediction and trajectory forecasting via predictive inverse linear-quadratic regulation. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[37] Roberto A Montano Murillo, Sriram Subramanian, and Diego Martinez Plasencia. 2017. Erg-O: Ergonomic optimization of immersive virtual environments. In *Proceedings of the 30th annual ACM symposium on user interface software and technology*. 759–771.

[38] Jörg Müller, Antti Oulasvirta, and Roderick Murray-Smith. 2017. Control theoretic models of pointing. *ACM Transactions on Computer-Human Interaction (TOCHI)* 24, 4 (2017), 1–36.

[39] Thomas Nescher, Ying-Yin Huang, and Andreas Kunz. 2014. Planning redirection techniques for optimal free walking experience using model predictive control. In *2014 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 111–118.

[40] Michael Rietzler, Florian Geiselhart, Jan Gugenheimer, and Enrico Rukzio. 2018. Breaking the tracking: Enabling weight perception using perceivable tracking offsets. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–12.

[41] Ali Safavi and Mehrdad H Zadeh. 2015. Model-based haptic guidance in surgical skill improvement. In *2015 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 1104–1109.

[42] Jeffrey A Saunders and David C Knill. 2003. Humans use continuous visual feedback from the hand to control fast reaching movements. *Experimental brain research* 152, 3 (2003), 341–352.

[43] Jeffrey A Saunders and David C Knill. 2004. Visual feedback control of hand movements. *Journal of Neuroscience* 24, 13 (2004), 3223–3234.

[44] Jeffrey A Saunders and David C Knill. 2005. Humans use continuous visual feedback from the hand to control both the direction and distance of pointing movements. *Experimental brain research* 162, 4 (2005), 458–473.

[45] Richard A Schmidt, Howard Zelaznik, Brian Hawkins, James S Frank, and John T Quinn Jr. 1979. Motor-output variability: a theory for the accuracy of rapid motor acts. *Psychological review* 86, 5 (1979), 415.

[46] Samuel J Sober and Philip N Sabes. 2003. Multisensory integration during motor planning. *Journal of Neuroscience* 23, 18 (2003), 6982–6992.

[47] Emanuel Todorov. 2005. Stochastic optimal control and estimation methods adapted to the noise characteristics of the sensorimotor system. *Neural computation* 17, 5 (2005), 1084–1108.

[48] Yoji Uno, Mitsuo Kawato, and Rika Suzuki. 1989. Formation and control of optimal trajectory in human multijoint arm movement. *Biological cybernetics* 61, 2 (1989), 89–101.

[49] Daniel M Wolpert and Zoubin Ghahramani. 2000. Computational principles of movement neuroscience. *Nature neuroscience* 3, 11 (2000), 1212–1217.

[50] Daniel M Wolpert, Zoubin Ghahramani, and J Randall Flanagan. 2001. Perspectives and problems in motor learning. *Trends in cognitive sciences* 5, 11 (2001), 487–494.

[51] Robert Sessions Woodworth. 1899. Accuracy of voluntary movement. *The Psychological Review: Monograph Supplements* 3, 3 (1899), i.

[52] André Zenner and Antonio Krüger. 2019. Estimating Detection Thresholds for Desktop-Scale Hand Redirection in Virtual Reality. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 47–55. https://doi.org/10.1109/VR.2019.8798143

[53] André Zenner, Kora Persephone Regitz, and Antonio Krüger. 2021. Blink-Suppressed Hand Redirection. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. 75–84. https://doi.org/10.1109/VR50410.2021.00028

[54] André Zenner, Kristin Ullmann, and Antonio Krüger. 2021. Combining Dynamic Passive Haptics and Haptic Retargeting for Enhanced Haptic Feedback in Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics* 27, 5 (2021), 2627–2637. https://doi.org/10.1109/TVCG.2021.3067777

[55] Yiwei Zhao and Sean Follmer. 2018. A functional optimization based approach for continuous 3d retargeted touch of arbitrary, complex boundaries in haptic virtual reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–12.

[56] Brian Ziebart, Anind Dey, and J. Andrew Bagnell. 2012. Probabilistic Pointing Target Prediction via Inverse Optimal Control. In *Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces* (Lisbon, Portugal) *(IUI '12)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/2166966.2166968