Human-Informed Robot Agility: Understanding Human Pushing Interactions for Skill Transfer to Humanoids

Cornelia Bauer and Nancy S. Pollard

***Abstract*— This research focuses on the analysis of human dynamic interactions and their applicability to robot systems. Studying human mechanics and control strategies in these interactions provides valuable insights for developing control approaches for dynamic interactions in robots. This work captures and analyzes wall push-offs as a specific interaction, and develops a skill model capturing push-off behaviors adaptable to different robot embodiments. Pushing skills are modeled as position trajectories relative to the center of mass, implicitly encoding interaction forces. We show that this approach can be scaled and adapted through optimization, and present results for transferring wall push-offs to two different robot platforms. Videos and additional data are available on the project website (https://sites.google.com/andrew.cmu.edu/human-robot-agility).**

I. INTRODUCTION

Dynamic interactions play a fundamental role in human capabilities, enabling us to achieve a wide range of tasks such as moving heavy objects, manipulating our surroundings, and changing directions rapidly and safely. In contrast, most conventional robotic systems lack this level of agility and cannot perform dynamic interactions, limiting their potential in practical applications. Recognizing the value of dynamic interactions as crucial skills for robots, this research endeavors to analyze human skills, and explore their applicability to robot systems. By studying the biomechanics and control strategies employed by humans during agile tasks, our goal is to unlock valuable insights that can guide the development of control approaches to perform dynamic interactions with robots.

In this work, we capture and analyze how humans utilize wall push-offs to quickly change their direction of motion. We present a skill model that captures push-off behaviors and can be adapted and employed for different robot embodiments. We model pushing behaviors as hand or end-effector position trajectories relative to the body's center of mass (CoM), and our model implicitly encodes interaction forces through position offsets. Using this framework, pushing skills can be adjusted to different robots by varying skill parameters. To trigger pushing actions in robots, we adopt a reflex-based controller approach that activates upon detecting a contact event. To our surprise, this simple model generates human-like push-off behaviors in robots, even without any adjustments. We further improve the robot behaviors by finding optimal skill parameters through optimization in simulation, and present results for optimized push-off skills for two different robot platforms, the CMU Ballbot [1] and the PushBot [2].

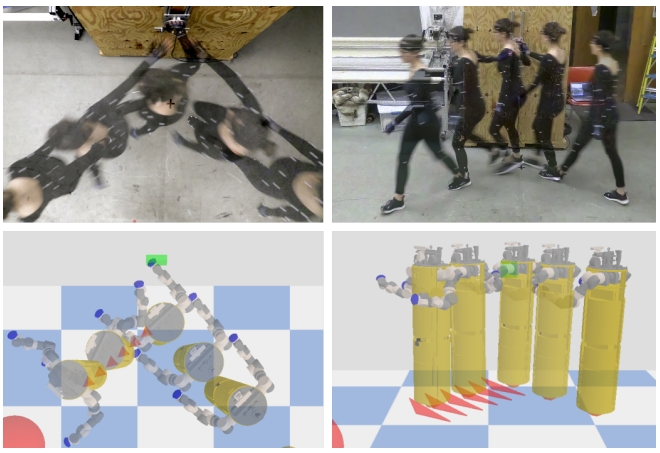

Fig. 1. Human push-off experiment example sequence (top) and skill transfer to the Ballbot humanoid robot (bottom).

II. RELATED WORKS

Analysis of dynamic human motion has long been of interest in the field of bio-mechanics and medicine. For example, Jansen et al. [3] conducted a study on human obstacle avoidance strategies under vision impairment, while other work [4], [5] used motion capture (MoCap) to analyze Parkour landings to understand human strategies for injury prevention and angular momentum regulation.

Humans are exceptionally talented at executing highly dynamic motions and can easily recover from mistakes. As a result, capturing and analyzing human motion to transfer skills to robots has long been of interest to the robotics community as well. Several MoCap datasets have been released that provide recordings of humans performing basic skill tasks such as walking, running, grasping, or demonstrating more specialized skills such as playing sports, dancing, or working in a kitchen [6], [7], [8]. The majority of high-fidelity datasets focus on the kinematics of human motion and are recorded using the Vicon motion capture system. However, capturing dynamic motions that involve interaction forces requires additional instrumentation. Maldonado et al. [9] capture and analyze the kinematics and dynamics of a kong-vault jump (a parkour technique for overcoming obstacles) by using a Vicon motion capture system, two force plates, and two handlebar sensors.

Early works that use human motions to generate synthesized character motions can be found in computer graphics.

*This research was supported by the National Science Foundation award CMMI-1925130.

Cornelia Bauer and Nancy S. Pollard are with the Robotics Institute, Carnegie Mellon Univ, 5000 Forbes Ave, Pittsburgh, PA 15213, USA, cornelib@andrew.cmu.edu, nsp@andrew.cmu.edu

Arikan et al. [10] use motions of a real person responding to being pushed to synthesize new character motions. To rapidly edit highly dynamic motion capture data, Abe et al. [11] use an optimization algorithm that can transform the captured motion so that it satisfies high-level user constraints while enforcing that the linear and angular momentum of the motion remains physically plausible. Lui et al. [12] use single motion capture examples to learn the skills required by real-time physics-based avatars to perform parkour-style fast terrain crossings.

In addition to transferring human motion to human characters, studies have also transferred motions to humanoid robots with various embodiments. For instance, Miura et al. [13] generate walking and turning motions of a humanoid robot based on human motion capture data. Amor et al. [14] learn human-robot interaction skills from captured human-human interactions using dynamic motion primitives.

Despite extensive analysis of dynamic human motion and the development of approaches to transfer these skills to simulated or physical agents, dynamic interactions have received little attention. This is especially true for human pushing maneuvers, despite their widespread use and practical utility (e.g., navigating cluttered environments and workspaces). We build on our prior work [2] where synthetic push-off behaviors were generated for a robot that approaches the wall straight on and pushes off with both hands. We successfully transferred these behaviors to a real robot using a reflex-based controller, without the need for further retraining. In this current work, we eliminate the need for hand-crafted policies and instead create pushing models from human demonstrations. This allows us to produce refined pushing behaviors for a wider range of entry and exit angles and pushes with just one arm. Compared to our previous work, which solely focused on reaching a goal position after the push, the human data incorporates features that encourage efficient pushes through redirecting the momentum and improving maneuverability after the push by trying to match the robot heading with the direction of motion. We show that our approach scales well to complicated humanoid robots, and find that a simple model effectively captures human push-off motions. This simplicity enables a straightforward process to transfer pushing motions to very different robots.

III. HUMAN MOTION CAPTURE EXPERIMENTS AND ANALYSIS

We conduct a motion capture experiment to study how humans push off from a wall with their hand in order to dynamically change their direction of motion.

The experimental setup, shown in fig. 2, consists of a wall that has a 6-axis force-torque (FT) sensor attached for measuring reaction forces. Study participants were instructed to approach the wall at a walking (jogging) pace and use the FT sensor to push off to move towards the goal position marked with a cone. The study protocol can be found on the project website. We track human motions using a Vicon motion capture system with the marker set placed as in [15] and record the forces and torques exerted on the wall during interaction using the FT sensor. A human push-off demonstration is shown in fig. 1.

Fig. 2. Motion capture setup: Study participants wear a full-body suit with markers that are tracked by a Vicon motion capture system. A wall-mounted force-torque sensor is used to record reaction forces and cones are placed at the respective entry and exit angles for visual guidance during the motion.

A. Data Collection

The study included 11 participants (5 male and 6 female). For each, we recorded 54 pushes resulting in a total of 594 pushing motions. We choose two approach speeds (*walking, running*), two approach angles ($0°, 45°$) and five exit angles ($0°, 30°, 45°, 60°, 90°$). We note that for a $0°$ degree approach angle, we did not record $0°$ exit angles because this combination would not require any change of motion. The complete list of pushing motions can be found in table I. fig. 3 visualizes the center of mass (CoM) trajectories of all participants for the two different entry angles ($0°$, $45°$).

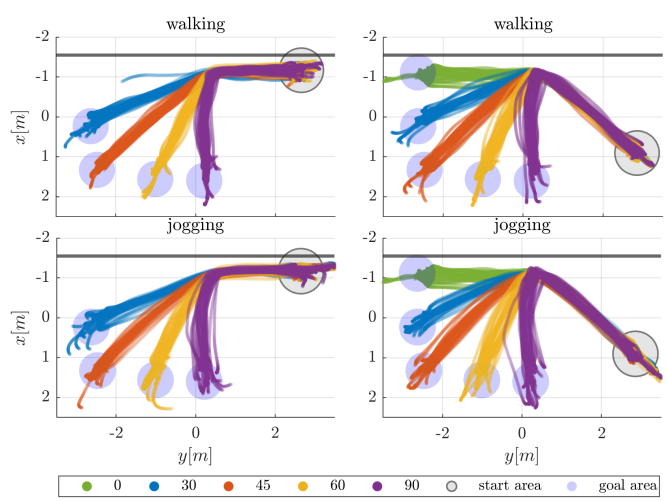

Fig. 3. Top view of center of mass trajectories for $0°$(left), $45°$(right) entry angles and varying exit angle, shown for walking speed (top), and jogging speed (bottom); x points perpendicular to the wall, y parallel to it.

B. Mass model and Body Center of Mass

We take inspiration from existing work [16] to model the human body and the distribution of mass and inertia across body segments. Starting from the skeleton model of the MoCap data format, we assign a mass m_i to every segment i. Individual segments are shown in fig. 4, labels and mass values can be found on the project website. Each segment is

modeled as a cylinder, where the length l_i equals the length of the MoCap model bone segment length. We assume a uniform density of $\rho = 1\text{g/cm}^3$ across all body segments. The radius of the segment cylinder can then be found by $r_i = \sqrt{m_i/(\pi l_i \rho)}$. The CoM $\boldsymbol{x}_{com,i}$ of each segment is located at the center of the cylinder, and the total body CoM is given by the sum of individual CoM positions, weighted by the individual masses: $\boldsymbol{x}_{com} = \frac{1}{m_t}\sum_{i=1}^{n} m_i \boldsymbol{x}_{com,i}$

The total body CoM coordinate frame uses the convention x-right, y-front, z-up. This mass model is scaled to individual study participant body weight estimates. The resulting cylindrical segment model is shown in fig. 4.

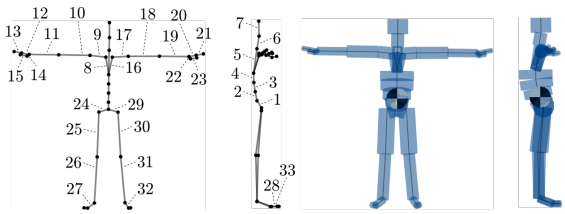

Fig. 4. Left: MoCap skeleton bone segments (labels on project website). Right: Each bone segment corresponds to a cylindrical link scaled based on bone segment length and mass value. The center of mass is marked by a black-and-white sphere.

C. Momentum and Impulse during Push-Off Tasks

To analyze how much the push-off component contributes to the motion during the task, we calculate the linear momentum $\boldsymbol{p}$ given by $\boldsymbol{p} = m\boldsymbol{v}$ throughout push-off tasks, where m is the body's total mass and $\boldsymbol{v}$ denotes its linear velocity. By measuring the reaction force $\boldsymbol{F}_w$ with the wall surface, we can calculate the impulse applied to the body at the hand-wall contact:

$$\boldsymbol{J}_w = \int_{t_0}^{t_f} \boldsymbol{F}_w dt$$

where t_0, t_f mark the start and end of the contact. The impulse-momentum theorem states that the change in momentum of a body $\Delta\boldsymbol{p}_{total}$ equals the impulse applied to it,

$$\int_{t_0}^{t_f} (\boldsymbol{F}_w + \boldsymbol{F}_u)dt = m\boldsymbol{v_f} - m\boldsymbol{v_0} = \Delta\boldsymbol{p}_{total}$$

where $\boldsymbol{F}_w$ is the force acting on the hand at contact with the wall, $\boldsymbol{F}_u$ are other (unknown) forces acting on the body, e.g foot contact forces, and $\boldsymbol{v}_0$ and $\boldsymbol{v}_f$ are initial and final linear velocity of the body, respectively. For each of the entry-exit angle combinations we captured, table I lists the maximum force measured at the wall $\boldsymbol{F}_w$, the push-off impulse normal to the wall J_n, the total change in linear momentum in the direction normal to the wall Δp_{tn}, and the contribution of the push-off impulse to the total change in linear momentum ($J_n/\Delta p_{tn} \cdot 100\%$). As shown in the example plots fig. 5, for shallow push-off angles, the magnitude of the linear momentum over time stays relatively constant. The change in momentum is the greatest while the hand is in contact with the wall and can primarily be observed perpendicular to the wall. For more acute entry-exit angle combinations (e.g. $45° - 90°$), we observe a temporary dip in momentum magnitude during the hand-wall contact, as subjects slow down, and regain momentum as they accelerate again when exiting the push-off.

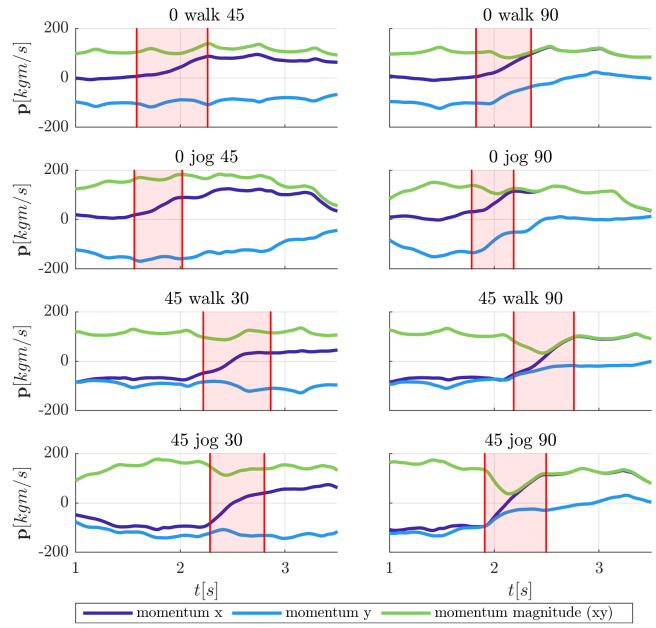

Fig. 5. Momentum time series for different push-off cases, labeled with 'entry angle' 'speed' 'exit angle'. Time frames where the hand is in contact with the wall are highlighted in red.

We find that the impulse from the arm push-off contributes substantially ($39\% - 91\%$) to the overall change in momentum both for walking and jogging and across all entry-exit angle combinations. We observe that the measured force at the wall F_w increases with increasing exit angles, with the highest forces observed for acute angles (e.g. $45° - 60°$). At the same time, the overall contribution of the arm push-off to the change in momentum decreases with increasing exit angles from the wall suggesting that the push-off motion contributes proportionally less, while the importance of foot contact forces increases. This can be attributed to leg muscles generally being much stronger than arm muscles [17]. While the majority did not apply forces $> 300N$ we observed large variations among participants.

Additional data from the study, and the MoCap dataset are available for reference on the project website.

D. Modeling Pushing Motions and Forces for Skill Transfer

For each push-off motion, both the arm and body motion as well as the applied forces are important. We create separate models for representing motions and applied forces that can be combined into a *pushing skill*. To extract the models from the motion capture study data, we employ the following procedure. First, we calculate the position of the human body's CoM in each time step based on the cylindrical bone model shown in fig. 4. We then examine the trajectories of the arms and the wall interaction forces in the CoM frame for each participant. Next, we focus on the right-hand trajectory w.r.t. the CoM frame. We segment and extract the portions of the trajectory where the hand is in contact with the wall. Simultaneously, we also segment and extract

TABLE I

ENTRY ANGLE, EXIT ANGLE, MAXIMUM PUSHING FORCE PER PARTICIPANT $\boldsymbol{F}_{w,max}$, PUSHING IMPULSE J_n, CHANGE IN MOMENTUM PERPENDICULAR TO WALL Δp_{tn}, AND CONTRIBUTION OF PUSH IMPULSE TO IT IN PERCENTAGE ($J_n/\Delta p_{tn} \cdot 100\%$)

angle (°)		$\mathbf{F}_{w,max}$ (N)		J (Ns)	Δp_{tn} (Ns)	$\frac{J_n}{\Delta p_{tn}}(\%)$
in	out	mean	max	mean (std)	mean (std)	mean (std)
walking						
0	30	141.2	218.7	31.0 (13.5)	34.8 (10.1)	91.0 (35.4)
0	45	160.8	249.1	37.4 (12.2)	54.6 (15.0)	71.4 (24.7)
0	60	175.7	247.3	43.1 (11.6)	73.1 (19.2)	60.4 (13.8)
0	90	189.4	285.0	47.2 (12.8)	79.1 (20.9)	61.6 (17.0)
45	0	163.8	230.9	41.0 (14.4)	52.0 (18.9)	83.4 (30.1)
45	30	187.1	256.8	49.0 (12.3)	77.3 (23.8)	67.4 (22.4)
45	45	208.5	292.8	54.6 (15.6)	97.5 (26.9)	58.0 (16.9)
45	60	222.6	304.3	60.6 (15.8)	114.7 (31.5)	55.3 (16.7)
45	90	224.8	382.9	65.3 (13.8)	124.1 (36.0)	56.3 (17.7)
jogging						
0	30	181.6	244.9	32.8 (10.7)	48.2 (16.4)	72.6 (28.9)
0	45	207.3	281.7	36.0 (10.4)	67.4 (19.5)	55.5 (16.9)
0	60	216.1	335.8	38.0 (11.3)	84.9 (19.6)	45.7 (12.2)
0	90	244.5	360.7	42.7 (13.2)	93.5 (28.7)	46.8 (11.9)
45	0	191.3	280.3	33.0 (10.0)	70.3 (24.2)	49.0 (15.4)
45	30	231.9	383.1	44.7 (13.3)	106.4 (32.5)	43.2 (11.5)
45	45	269.3	481.0	49.2 (13.7)	118.0 (33.4)	42.6 (9.5)
45	60	273.2	470.3	55.5 (15.0)	146.6 (49.9)	39.3 (8.3)
45	90	267.1	457.0	62.0 (18.6)	153.0 (52.1)	42.2 (11.0)

the corresponding time series of reaction forces experienced during the hand-wall contact.

To ensure the duration of each push is uniformly scaled between participants and pushes, we normalize the duration of each push $\tilde{t} \in [0, 1]$ with $\Delta\tilde{t} = \frac{1}{N-1}$ and N the number of time steps for a push per recording.

We model the hand trajectory $\boldsymbol{\xi} = (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})^T$ during pushes by fitting a third-degree polynomial to each of the three components $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ of the segmented right-hand trajectories. An example of right-hand trajectories and the fitted model are visualized in fig. 6 (center) from a top-view perspective. Furthermore, we also fit a three-term Fourier series model to the x, y, and z-components of the segmented force time series. After evaluating various models (e.g. polynomial, Gaussian), we concluded that the Fourier series model provided the most accurate representation of the force data. An example of recorded force time series and the resulting fitted force model is shown in fig. 6 (left) for a push with a $0°$ entry and $45°$ exit angle at walking speed.

In order to combine the information about hand trajectories and exerted wall forces into a single model, we first translate the force information into an equivalent displacement $\boldsymbol{d} = \frac{1}{k}\boldsymbol{F}$ with k the stiffness coefficient and $\boldsymbol{F}$ the force time series. We can then superimpose the force displacement with the hand trajectory data to represent a *pushing skill*

$$\boldsymbol{\xi}_{push} = \boldsymbol{\xi}_{hand} + \boldsymbol{d} = \boldsymbol{\xi}_{hand} + \frac{1}{k}\boldsymbol{F}.$$

The original hand trajectory $\boldsymbol{\xi}_{hand}$, force displacement $\boldsymbol{d}$ and resulting push trajectory $\boldsymbol{\xi}_{push}$ are visualized in fig. 6 (right).

IV. RETARGETING DYNAMIC PUSHING MOTIONS TO ROBOTS

Our goal is to re-target the push-off motions observed in our study from humans to mobile robots. We develop a transfer approach that is applicable to arbitrary robots with at least one arm. In the following sections, we detail the transfer process and demonstrate its effectiveness through two robot examples: The CMU Ballbot and the PushBot.

The **Ballbot** is a unique, human-sized robot that balances on a single spherical ball. By tilting its body in the desired direction, the Ballbot creates a driving force that propels it forward. This lean-based locomotion allows the robot to move smoothly and agilely in any direction. The Ballbot is equipped with two articulated arms, each with seven degrees of freedom (DoF), providing a wide range of motion.

The **PushBot** is a small omnidirectional robot platform which uses a hovercraft-like lift system, creating a low friction air cushion that allows the robot to move freely in all directions. The PushBot has two planar 2-DoF arms, and moves by interacting with its environment.

A. Simulation environment

In this section we describe how we set up the simulation environment for each of the robots, what assumptions are made, and how the robots are controlled. In our study, we employ PyBullet to create a three-dimensional simulation test setup that includes a rigid wall object from which robots can push off.

The **Ballbot's** inverse mouse-ball drive mechanism (IMBD) is modeled using a spherical joint. Additionally, a separate controllable yaw DoF allows for body rotation. Each arm of the Ballbot features seven DoF, and the end-effectors are knob hands that are not articulated.

The body controller features a staged combination of a low-level balancing controller with center of mass compensation and an outer control loop that accepts lean angle commands, enabling the robot to lean and accelerate toward a desired direction. Each arm controller is realized as a cartesian impedance controller, with feed-forward terms for gravity, nonlinear terms, and a null-space projection keeping the arm joints close to a *neutral* configuration (elbows bent) and away from joint limits.

The omnidirectional free-floating hovercraft body of the **PushBot** is modeled through a fixed virtual base and two prismatic joints in the x and y-direction, and an unactuated yaw joint aligned with the z-axis. This setup allows the body to move freely in the xy-plane, and rotate about the vertical axis. The 2 DoF arms of the Pushbot are controlled by a cartesian impedance motion controller described in the following section.

B. Cartesian Impedance Motion Controller

We control the arms of the Ballbot and PushBot through a cartesian impedance motion controller without inertia shaping, that effectively emulates a spring-damper system. This controller receives commands w.r.t. the shoulder frame.

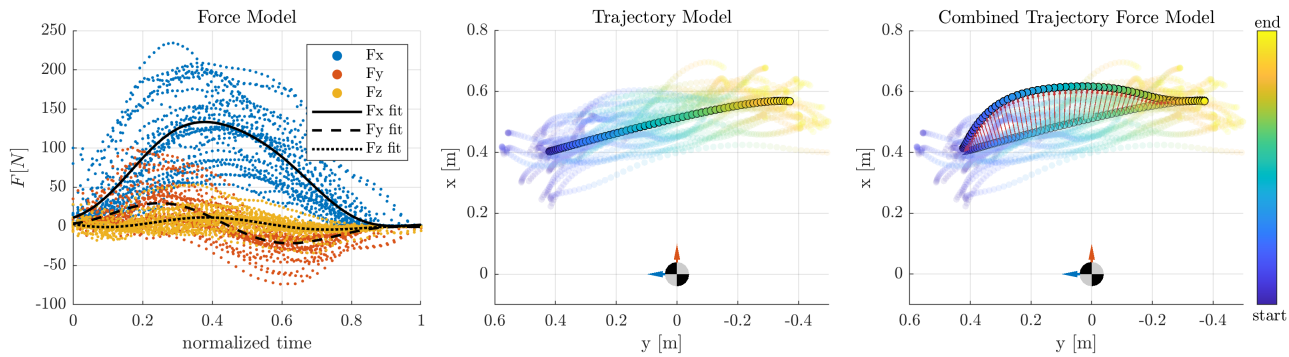

Fig. 6. Pushing skill model example ($0° - 45°$ at walking speed). Left: Force profiles over duration of the push, human data and fitted force model. Forces are shown w.r.t. center of mass coordinate frame (x-right, y-front, z-up). Center: Top-view of right hand trajectory profiles (transparent), and fitted trajectory model (black edges) w.r.t. center of mass, plotted at the origin. Blue arrow points toward heading, and red arrow points toward right side of human body model. Right: Combined trajectory force model; at each trajectory point, a scaled force offset is added to the position, indicated by red arrows.

Based on the current arm configuration, we calculate the position error at the end effector $\boldsymbol{e}_x$ and its derivative $\dot{\boldsymbol{e}}_x = \frac{d}{dt}\boldsymbol{e}_x$. The desired task force $\boldsymbol{f}_d$ is given by

$$\boldsymbol{f}_d = \ddot{\boldsymbol{x}}_d + \boldsymbol{K}_{p,x}\boldsymbol{e}_x + \boldsymbol{K}_{d,x}\dot{\boldsymbol{e}}_x$$

with desired acceleration $\ddot{\boldsymbol{x}}_d$, stiffness and damping matrices $\boldsymbol{K}_{p,x}, \boldsymbol{K}_{d,x}$, and this force is mapped to the joint torques based on

$$\boldsymbol{\tau}_{cmd} = \boldsymbol{J}^T\boldsymbol{f}_d = \boldsymbol{J}^T(\ddot{\boldsymbol{x}}_d + \boldsymbol{K}_{p,x}\boldsymbol{e}_x + \boldsymbol{K}_{d,x}\dot{\boldsymbol{e}}_x).$$

We compute the feedforward terms $\boldsymbol{\tau}_f$ as the sum of the coriolis and centrifugal terms $\boldsymbol{C}$, and the gravity terms $\boldsymbol{G}$.

$$\boldsymbol{\tau}_f = \boldsymbol{C}(q, \dot{q}) + \boldsymbol{G}(q)$$

To keep the arm close to a desired *neutral* configuration $\boldsymbol{q}_d$ (bent elbows) and away from joint limits, we add an additional joint space torque

$$\boldsymbol{\tau}_q = \boldsymbol{K}_{p,q}\boldsymbol{e}_q + \boldsymbol{K}_{d,q}\dot{\boldsymbol{e}}_q$$

with gain matrices $\boldsymbol{K}_{p,q}, \boldsymbol{K}_{d,q}$, and joint position error $\boldsymbol{e}_q$.

In the case of the Ballbot arms, we project this component into the null-space using

$$\boldsymbol{\tau}_n = (\boldsymbol{I} - \boldsymbol{J}^{\#}\boldsymbol{J})\boldsymbol{\tau}_q$$

with pseudoinverse $\boldsymbol{J}^{\#}$. Finally, all torques are combined into the final control torque

$$\boldsymbol{\tau} = \boldsymbol{\tau}_f + \boldsymbol{\tau}_{cmd} + \boldsymbol{\tau}_n.$$

For the PushBot, since the 2 DoF arms are not redundant, we simply omit the nullspace projection and instead add the resulting joint space torque, $\boldsymbol{\tau_n} = \boldsymbol{\tau_q}$ in the control law. The controller gains are listed in table II.

TABLE II
CARTESIAN IMPEDANCE CONTROLLER GAINS

	$K_{p,x}$	$K_{d,x}$	$K_{p,q}$	$K_{d,q}$
Ballbot	4000	400	25	2
PushBot	100	1	0.2	0.02

C. Reflex-Based Pushing Skill Controller

A high-level skill controller switches between different modes, and executes the *pushing skill* when the hand makes contact with an obstacle (trigger event). At each timestep, this controller outputs a cartesian end-effector position and sends it as a command to the arm task-space impedance controller. The controller output changes based on the current mode. The different modes of the skill controller are:

0) *Freemotion*: Hand positioned in an anticipating configuration. This is the average position of the hand w.r.t. the CoM at the contact event from all motion capture recordings.
1) *Push*: Hand executes pushing skill trajectory generated by the process described in section III-D.
2) *Recovery*: End-effector moves from the current position back to the freemotion position on a linear trajectory.

Figure 7 shows the mode sequence for the pushing skill controller. The controller launches in *freemotion* mode. Contact at the end effector triggers the switch to *push*, which lets the arm execute the push. After the pushing skill trajectory is completed, the controller switches into *recovery* mode. When the end effector completes the recovery trajectory and is back at the initial position, the controller switches back to *freemotion*.

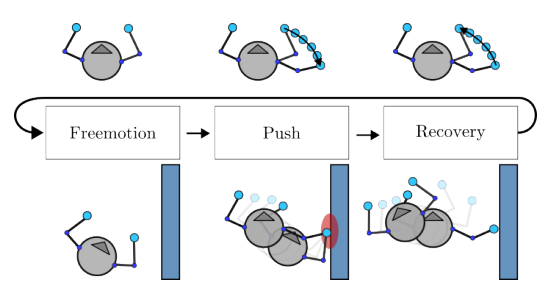

Fig. 7. Reflex-based skill controller. The controller starts in *freemotion* mode, contact at the end effector triggers the switch to *push*. After the pushing skill trajectory is completed, the controller switches into *recovery* mode, and after completing the recovery trajectory back to *freemotion*.

D. Transfer and Optimization

The approach detailed in section III-D generates pushing skill trajectories that are described with respect to the center of mass frame of the human body, as shown in fig. 8 (right). We can transfer these trajectories to a robot as follows: We first copy them into the CoM frame of the robot. We then transform the trajectories into the shoulder frame and scale them to lie within the robot arm workspace.

However, simply copying and scaling the trajectories is not enough to transfer the skill properly to robots, as shown in our experiments in section V. To account for different dynamic properties between robots and humans, we introduce skill parameters that allow us to further modify the push.

We parameterize the push in terms of pushing angle ϕ, and stiffness coefficient k. The pushing angle ϕ allows for rotating the complete pushing trajectory, including force displacements, in the xy - plane around the vertical axis of the shoulder origin. Varying the stiffness coefficient k scales the force displacement $\boldsymbol{d}$ as described in section III-D.

Fig. 8. Pushing skill model transfer from human to robots, for the example of a $0° - 45°$ push at walking speed: We fit the pushing skill model to the data captured in our MoCap study. The original human trajectories for this push are shown as partially transparent lines next to the human model (right), the resulting model is highlighted in solid color. The skill model is transferred to the Ballbot (center), and the PushBot (left). Trajectories are shifted w.r.t. the CoM of the different embodiments, and scaled to lie within arm workspace.

To find suitable pushing skill parameters $\theta = (\phi, \frac{1}{k})$ we use Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [18]. We evaluate the fitness of pushing parameter values based on the following cost terms.

- Position cost c_p: the distance of the CoM from the target position, $2s$ after initial wall contact.
- Push-off velocity angle cost c_a: cumulative difference between angle of current CoM velocity direction and target push-off angle, within $2s$ after initial wall contact.
- Final heading cost c_h: difference between current robot heading and target heading, $2s$ after initial wall contact.

The total cost is given by the weighted sum of all cost terms, $c_t = w_p c_p + w_a c_a + w_h c_h$. We obtain average target locations, angles and heading values for each push from our motion capture study described in section III. For our optimization experiments, we set the weights to $w_p = 100, w_a = 0.01, w_h = 1$.

V. EXPERIMENTS AND RESULTS

When transferring pushing skills from humans to robots, we observe that simply copying and scaling the pushes according to our model surprisingly already produces push-off behaviors in robots as shown in our video submission; however, in most cases, the resulting push-off direction is not aligned with the goal. We optimize pushing skills for both the Ballbot and the PushBot in simulation for two speeds, two entry angles, 4 and 5 exit angles, respectively, with a total of 18 different test cases per robot. The resulting skill parameters, cost values, push impulses and maximum forces are listed in table III, and the CoM trajectories are shown in figs. 10 and 11.

The total runtime of the optimization for all test cases on an 8-core (3.6GHz) CPU is 125 min for the Ballbot and 88 min for the PushBot.

The lower and upper bounds for the skill parameters are set to $\theta_l = (-90°, 0.0), \theta_{u,bb} = (90°, 0.002)$ for the Ballbot, and $\theta_{u,pb} = (90°, 0.02)$ for the PushBot. Given the pushing skill model described in section III-D, the bounds for the inverse stiffness coefficient $\frac{1}{k}$ can be interpreted in the following way: At the lower bound $\frac{1}{k} = 0$, the stiffness coefficient k is infinite, and the force-displacement becomes null, resulting in purely matching the modeled hand trajectory. At the upper bound, a maximum force-displacement of $0.002\boldsymbol{F}$ is added to the pure hand trajectory model. This corresponds to a stiffness of 500N/m, and with the mean maximum forces observed in our study ranging within $[141N, 274N]$, results in a maximum position-offset of $|\boldsymbol{d}| < 0.548m$.

When comparing the stiffness coefficients $K_{p,x}$ used in our cartesian impedance controller (see table II) for the two robots, we find that the difference in order of magnitude can be attributed to the difference in total body mass (Ballbot: 94.952kg, PushBot: 0.666kg). The magnitude of scaling aligns with previous work [19] which has shown that scaling mass and stiffness coefficients can be estimated by $\frac{m_1}{m_2} = L^3$, $\frac{K_1}{K_2} = L^2$. In our case, this would yield a stiffness scaling factor of 27.3 from PushBot to Ballbot, which is in the same order of magnitude as our scaling factor of 40.0.

We find that overall the skill transfer is successful for both robots and produces human-like push-off motions. There are certain commonalities observed in the pushes executed by both robots and human, such as a similar CoM trajectories. Additionally, in both human and robot pushes, the arm induces a rotation in the direction of motion and yaw.

However, there are also notable differences between robot and human pushes. Due to the lack of feet, the Ballbot exerts higher impulses and forces with the arm when compared to humans. In contrast, humans utilize their legs to absorb impact and redirect momentum, particularly for acute angles and higher velocities. In the absence of legs, the optimization yields a strategy where the Ballbot utilizes both arms to

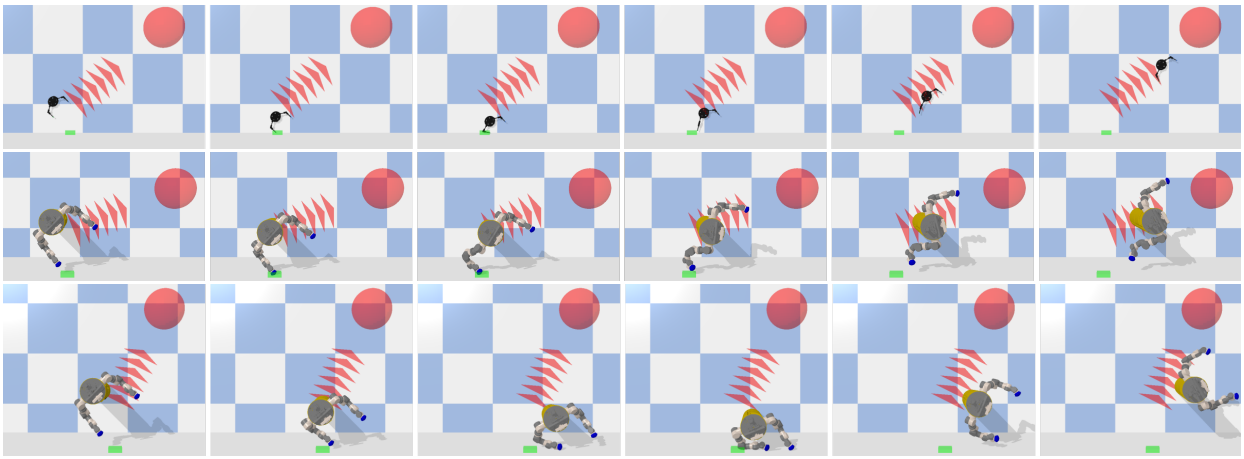

Fig. 9. Examples of optimization results for the PushBot (top) and the Ballbot (center, bottom). The target position is marked by a red circle, and red arrows visualize the push-off angle.

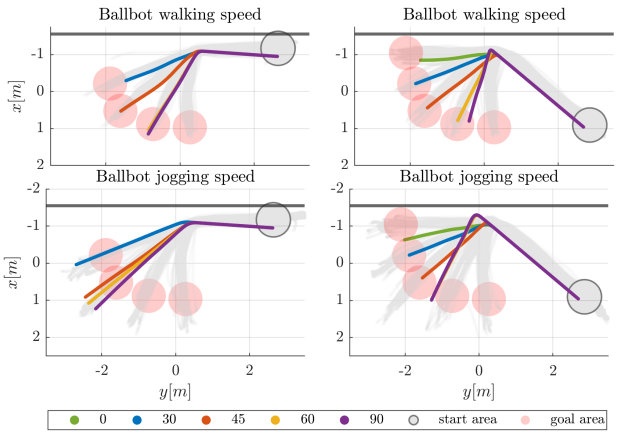

Fig. 10. Resulting CoM trajectories for optimized models for the Ballbot.

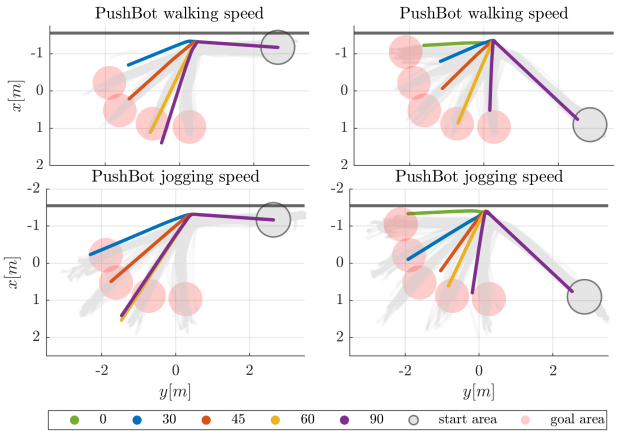

Fig. 11. Resulting CoM trajectories for optimized models for the PushBot.

absorb the initial impact in similar scenarios, as depicted in fig. 9 (bottom).

Furthermore, we observed that when the exit angle of the push is 90°, both humans and robots struggle to fully redirect the push using their arms alone. Instead, humans rely on their feet to correct their direction after the push. This can be seen in the *purple* CoM trajectories illustrated in fig. 3 which show that the exit angle right after the push is not exactly 90°, resulting in a slightly curved trajectory. In contrast, the Ballbot and PushBot can not significantly correct their direction after losing contact with the wall, which results in an exit angle that is less than 90° as shown in figs. 10 and 11. We hypothesize that the performance for acute angles can be improved by implementing different strategies, such as slowing down before the push or using both arms.

Additionally, we tested an alternative skill model which models trajectory and force separately. However, incorporating the modeled force as an additional term in the desired task force calculation (IV-B) did not produce favorable results in terms of cost and the motion's visual appearance.

Overall, our experiments demonstrate that the simple parameterization and transfer of pushing skills w.r.t the CoM is able to capture the human demonstrations well, and generate effective pushes to redirect the robot's motion. In our prior work, the PushBot was able to reach the goal, but consistently approached the wall at a 90° entry angle, used both arms, and relied on a handcrafted push-off model where forces are applied along the hand trajectory line. This limited the achievable exit angles to a more narrow range $[-26°, 26°]$, and resulted in less human-like motions after the push, with the robot sliding backwards and not changing its heading to face the goal. In contrast, our current study observes force profiles in human pushes that differ significantly from the prior model's assumptions, with forces primarily perpendicular to the pushing trajectory. By incorporating these observed force profiles into our modeling, we achieve motions that closely resemble the original human movements and effectively capture the change in heading after the push.

VI. CONCLUSION AND FUTURE WORK

This work presents an analysis of how humans use dynamic interactions for tasks, exemplified by wall push-offs. We captured a total of 594 interactions which we used as the basis for creating a human-based push-off model. Further, we demonstrate the successful transfer of this human model to mobile robots with arms.

Another potential application of our model lies in its suitability for unmanned aerial vehicles (UAVs) that navigate

TABLE III

OPTIMIZATION RESULTS FOR PUSHBOT, BALLBOT. ENTRY ANGLE, EXIT ANGLE, OPTIMIZED PUSHING ANGLE ϕ^*, OPTIMIZED INVERSE STIFFNESS COEFFICIENT $\frac{1}{k^*}$, TOTAL COST c_t, POSITION ERROR e_p, PUSHING IMPULSE J, MAXIMUM WALL CONTACT FORCE F_+

in	out	ϕ^*	$\frac{1}{k^*}$	c_t	$e_p(m)$	$J(Ns)$	$F_+(N)$
PushBot walking speed							
0	30	27.31	0.0007	84.9	0.70	0.05	3.07
0	45	-85.82	0.0009	59.1	0.40	0.47	10.28
0	60	-89.80	0.0129	43.7	0.25	0.97	14.92
0	90	-88.24	0.0200	304.1	0.87	1.03	14.76
45	0	-25.60	0.0135	102.7	0.52	0.48	25.18
45	30	43.17	0.0002	215.8	0.92	0.05	3.42
45	45	36.25	0.0006	201.7	0.76	0.14	8.04
45	60	49.04	0.0151	80.6	0.03	0.69	15.87
45	90	57.94	0.0093	167.3	0.46	1.32	15.98
PushBot jogging speed							
0	30	-57.21	0.0004	64.3	0.42	0.45	9.07
0	45	23.39	0.0102	22.6	0.14	0.86	15.19
0	60	-13.23	0.0038	195.5	0.99	1.30	16.55
0	90	-29.31	0.0054	575.3	1.79	1.23	15.67
45	0	-14.70	0.0044	138.4	0.34	0.50	15.89
45	30	43.66	0.0112	194.1	0.12	0.87	14.96
45	45	85.77	0.0035	187.1	0.65	0.18	17.63
45	60	68.64	0.0124	70.8	0.28	0.43	12.84
45	90	66.41	0.0059	214.2	0.48	1.51	49.29
Ballbot walking speed							
0	30	-20.70	0.0008	67.8	0.44	31.99	285.89
0	45	-82.39	0.0015	49.5	0.01	36.31	378.18
0	60	-17.83	0.0020	87.5	0.20	57.11	367.22
0	90	-16.95	0.0020	386.4	1.14	52.68	372.61
45	0	-45.00	0.0012	74.1	0.44	44.72	518.32
45	30	-50.22	0.0019	26.2	0.04	51.58	453.24
45	45	-14.49	0.0020	38.6	0.11	71.07	825.98
45	60	43.97	0.0020	36.8	0.10	123.99	866.30
45	90	58.95	0.0020	209.6	0.69	118.01	844.40
Ballbot jogging speed							
0	30	34.38	0.0020	106.9	0.85	47.49	465.74
0	45	-7.39	0.0020	152.5	0.93	44.86	431.34
0	60	-10.37	0.0020	409.0	1.64	50.86	439.04
0	90	-10.34	0.0020	700.5	2.45	68.43	448.56
45	0	1.40	0.0020	139.7	0.43	87.26	965.03
45	30	29.43	0.0018	30.7	0.00	95.08	817.26
45	45	42.89	0.0020	44.7	0.15	118.49	855.32
45	60	90.00	0.0008	93.2	0.56	66.05	731.23
45	90	88.43	0.0006	407.9	1.55	57.15	666.67

through confined spaces, where precise and efficient maneuvers are vital. UAVs equipped with an arm could harness our approach to push off to make turns or to stop, enhancing their performance in such situations. Our approach could also be adapted to emulate human-like motions for legged robots, given that a well-defined gait policy is available.

Our prior work has shown that optimized skills can be used as a basis for more general models to interpolate to unseen states. Future work involves modifying our model to interpolate between angles captured in our motion capture study, to produce push-off skills that cover a continuous entry-exit angle space. These skills could then be integrated into a higher-level task and skill planning algorithm, as another step towards robots that can effortlessly navigate and interact with the world around them.

ACKNOWLEDGMENT

The authors thank Justin Macey, Melanie Danver, and Dominik Bauer for their assistance in conducting experiments and all study participants for their contribution.

REFERENCES

[1] R. Shu and R. Hollis, "Development of a humanoid dual arm system for a single spherical wheeled balancing mobile robot," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, 2019, pp. 499–504.

[2] C. Bauer, D. Bauer, A. Allaire, C. G. Atkeson, and N. Pollard, "Learning to navigate by pushing," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 171–177.

[3] S. E. Jansen, A. Toet, and P. J. Werkhoven, "Human locomotion through a multiple obstacle environment: strategy changes as a result of visual field limitation," *Experimental Brain Research*, vol. 212, pp. 449–456, 2011.

[4] B. Dai, J. S. Layer, T. J. Hinshaw, R. F. Cook, and J. S. Dufek, "Kinematic analyses of parkour landings from as high as 2.7 meters," *Journal of Human Kinetics*, vol. 72, p. 15, 2020.

[5] G. Maldonado, F. Bailly, P. Souères, and B. Watier, "Angular momentum regulation strategies for highly dynamic landing in parkour," *Computer methods in biomechanics and biomedical engineering*, vol. 20, no. sup1, pp. S123–S124, 2017.

[6] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber, "Documentation mocap database hdm05," Universität Bonn, Tech. Rep. CG-2007-2, June 2007.

[7] C. Mandery, O. Terlemez, M. Do, N. Vahrenkamp, and T. Asfour, "Unifying representations and large-scale whole-body motion databases for studying human motion," *IEEE Transactions on Robotics*, vol. 32, no. 4, pp. 796–809, 2016.

[8] A. Szczesna, M. Błaszczyszyn, and M. Pawlyta, "Optical motion capture dataset of selected techniques in beginner and advanced kyokushin karate athletes," *Scientific Data*, vol. 8, no. 1, p. 13, 2021.

[9] G. Maldonado, F. Bailly, P. Souères, and B. Watier, "Inverse dynamics study of the parkour kong-vault during take-off," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 22, no. sup1, pp. S331–S333, 2019.

[10] O. Arikan, D. A. Forsyth, and J. F. O'Brien, "Pushing people around," in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '05. New York, NY, USA: Association for Computing Machinery, 2005, p. 59–66.

[11] Y. Abe, C. K. Liu, and Z. Popović, "Momentum-based parameterization of dynamic character motion," *Graphical Models*, vol. 68, no. 2, pp. 194–211, 2006, special Issue on SCA 2004.

[12] L. Liu, K. Yin, M. van de Panne, and B. Guo, "Terrain runner: control, parameterization, composition, and planning for highly dynamic motions." *ACM Trans. Graph.*, vol. 31, no. 6, pp. 154–1, 2012.

[13] K. Miura, M. Morisawa, S. Nakaoka, F. Kanehiro, K. Harada, K. Kaneko, and S. Kajita, "Robot motion remix based on motion capture data towards human-like locomotion of humanoid robots," in *2009 9th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2009, pp. 596–603.

[14] H. B. Amor, G. Neumann, S. Kamthe, O. Kroemer, and J. Peters, "Interaction primitives for human-robot cooperation tasks," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 2831–2837.

[15] "Cmu motion capture database," http://mocap.cs.cmu.edu/, accessed: 2023-07-09.

[16] W. L. Wooten, *Simulation of leaping, tumbling, landing, and balancing humans*. Georgia Institute of Technology, 1998.

[17] D. G. Candow and P. D. Chilibeck, "Differences in Size, Strength, and Power of Upper and Lower Body Muscle Groups in Young and Older Men," *The Journals of Gerontology: Series A*, vol. 60, no. 2, pp. 148–156, 02 2005.

[18] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary computation*, vol. 9, no. 2, pp. 159–195, 2001.

[19] M. H. Raibert and J. K. Hodgins, "Animation of dynamic legged locomotion," in *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, 1991, pp. 349–358.