

Taking Your Robot For a Walk: Force-Guiding a Mobile Robot Using Compliant Arms

François Ferland, Arnaud Aumont, Dominic Létourneau, François Michaud
Department of Electrical Engineering and Computer Engineering
Université de Sherbrooke, Québec, Canada
Email: {firstname.lastname}@usherbrooke.ca

Abstract—Guiding a mobile robot by the hand would make a simple and natural interface. This requires the ability to sense forces applied on the robot from direct physical contacts, and to translate these forces into motion commands. This paper presents a joint-space impedance control approach that does so by perceiving forces applied on compliant arms, making the robot react as a real-life physical object to a user pulling and pushing on one or both of its arms. By independently controlling stiffness in specific degrees-of-freedom, our approach allows the general position of the arms to change to the preferences of the person interacting with it, a capability that is not possible using a strictly position-based control approach. A test case with 15 volunteers was conducted on IRL-1, an omnidirectional, non-holonomic mobile robot, to study and fine-tune our approach in an unconstrained guiding task, making IRL-1 go in and out of a room through a doorway.

Keywords—*Direct physical interaction; Humanoid robot; Impedance control.*

I. INTRODUCTION

Humans use direct physical contact to influence their motion and postures. For instance, guiding people by their hands or shoulders is a common and natural way of interacting in all kinds of situations such as games, dances, sports, walking assistance, etc. As robots become more present in our environments, having the capability to physically guide a mobile robot would be useful and natural, as observed with children interacting with small mobile robots [1], [2]. However, this may be difficult or impossible to do with medium or large size robots, which are generally actuated using stiff, non backdrivable and constrained actuation mechanisms, limiting their ability to respond to physical contacts and making them potentially dangerous in case of a collision.

For a mobile robot, direct physical interaction (DPI) means being able to push or pull on its body to influence its motion. Humanoid robots equipped with force/torque sensors have recently been introduced. With MS DanceR [3], a force/torque sensor placed between the upper body and lower body allows a user to guide and dance with the robot. A different approach consists of exploiting compliant (i.e., torque-controlled) arm joints, such as the ones on Cody [4], [5], [6], PR2 [7] and Rollin' Justin [8], to physically guide the platforms. This capability has only been validated with Cody, using two 7 degree-of-freedom (DOF) compliant arms actuated using series elastic actuators (SEAs) [9], [10] and 6 axis force/torque sensors at the robot's wrists, installed on a height-adjustable torso and a Segway omnidirectional base. The arms were programmed to keep a single posture, or *home position*: The forearms were

parallel to the ground, with elbows bent at 90° , and each joint acted like a damped spring with a low and constant stiffness. The wrists were held parallel to the forearm and were position-controlled with higher stiffness. Displacement from the arms' home position (left/right) and applied forces measured at the wrist (forward/backward, clockwise/counterclockwise angular velocities, up/down) were linearly mapped to the mobile base or the torso. For instance, pulling on the wrist made the robot go in a straight line, pulling on its upper arm made it move laterally, and turning the arm around its shoulder vertical axis made it rotate. Trials with unexperienced users in a healthcare scenario suggest that such DPI can outperform a gamepad interface according to several objective and subjective measures [4].

Our research interest regarding DPI lies in perceiving direct physical contact from multiple locations on the robot's body. For instance, we developed AZIMUT-3, an omnidirectional non-holonomic four-wheel steerable platform. The use of differential elastic actuators (DEAs [11]) for wheel steering makes it possible to sense force/torque applied on the platform in the horizontal plane, allowing physical interactions from almost anywhere on the platform [12], [13]. Similarly to a SEA, a DEA acts as an active elastic element that can inherently absorb shocks, perceive the forces from the environment on the robot and control the forces applied back, a capability referred to as interaction control [14]. DEAs are conceptually similar to SEAs, but use a differential coupling (harmonic drive) instead of a serial coupling of a high impedance mechanical speed source (an electrical DC brushless motor) and a low impedance mechanical spring (a passive torsion spring). A non-turning sensor connected in series with the spring measures the torque output of the actuator. This results in a more compact and simpler solution for controlling mechanical elasticity and viscosity in accordance with an admittance control scheme [14], [15].

Following an iterative and incremental design methodology, DPI derived from AZIMUT-3's steerable wheels was then validated with the addition of a humanoid torso equipped with 4 DOF compliant arms (also using DEAs), creating a robot platform referred to as IRL-1 [16]. Using a cartesian impedance control scheme, the arms were programmed to be relatively stiff and in extension in front of the robot, with the grippers at 0.45 m from the shoulders as its home position. Torque measurements from DEAs on the arms were not exploited to detect physical interaction: Users could push or pull the arms or the grippers, generating forces on the wheels' contact point and detected by the DEA-steered powered wheels. Trials revealed that it was easy to physically

guide the robot through its grippers to move on a straight path, but it was more difficult to guide it on a curved path or to switch from a left-handed curve to a right-handed one, or vice versa. Torque measurements from DEA’s steering axes are noisy and contain a constant bias difficult to predict caused by the weight of the robot’s torso itself. In addition, part of the forces provided by the user gets absorbed by the DEAs on the arms, and thus users had to provide more force (compared to applying forces to other locations) to physically guide the platform.

To improve sensitivity for DPI applied on the arms, this paper presents an approach that uses joint-space impedance control of the arms’ DEAs to derive user’s intent in guiding the platform. Impedance control has been used before for physical HRI, for instance in collaborative tasks such as lifting objects [17], [18]. More specifically, joint-space impedance control has been used in an imitation learning context for performing a safe contact motion [19]. Our force control approach can be compared to the hybrid controller approach described in [20] and observed in [21], where one or more degrees of freedom are distinguished as being force-controlled rather than position-controlled. Exploiting a force-based approach to motion control extends our previous work with AZIMUT-3 to compliant arms. A test case with users having to physically guide IRL-1 through a doorway demonstrate the feasibility of our approach. Results using a remote gamepad controller also helps identify improvements to be made in preparation of an upcoming usability study.

II. JOINT-SPACE IMPEDANCE CONTROL FOR FORCE ESTIMATION USING COMPLIANT ARMS AND VELOCITY CONTROL

Instead of having the robot’s arms stay in a fixed position for DPI (as with Cody), our goal was to develop an interaction technique in which the height of a robot’s arm could be changed freely by the person interacting with it. This could help interacting with the robot in tight spaces. Using cartesian impedance control, the generalized force applied at the end effector can be estimated with its Jacobian matrix and the effect of gravity on joint torques without using a 6-axis force and torque sensor, for instance when estimating the contact force of a door-opening robot [22]. However, this technique requires at least six non-redundant DOF to fully estimate the force and torque at the end effector. Full estimation is not required to move a mobile platform around, and simplifications can be made using joint-space impedance control.

Figure 1 illustrates the block diagram of our approach. Impedance control of a revolute joint can effectively simulate a torsion spring and damper system, giving the arm a spring-like quality. All joints are controlled to statically compensate gravity depending on the arm’s pose, the centre of gravity of each linkage and their weights. To make our approach as natural as possible, our technique only applies stiffness to a subset of the arm joints. More specifically, the arm is programmed to move freely around its shoulder’s tilt axis, and so each arm can be moved up or down without influencing the robot’s motion. To simplify the problem, we do not estimate the torque applied to the arm’s end point, effectively reducing unknown variables from six (forces and torques in all three components X, Y, Z) to three. Furthermore, because the robot’s

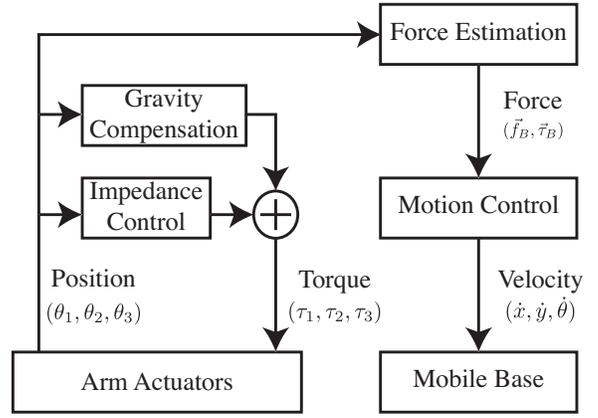


Fig. 1. Block diagram of our joint-space impedance control approach.

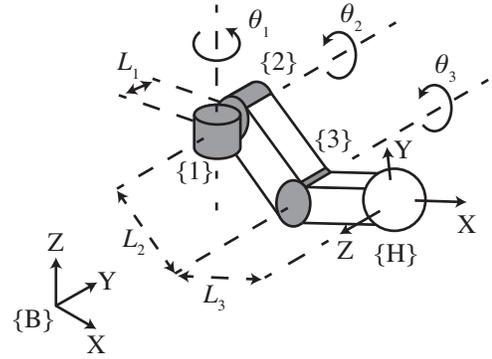


Fig. 2. A 3 DOF arm.

mobile base is constrained to a 2D plane, two out of three torque components applied to it do not influence the robot’s motion.

To explain further the approach, consider the generic 3 DOF arm shown in Fig. 2, with θ_1 and θ_2 the shoulder pan and tilt angles, respectively, and θ_3 the elbow tilt angle. $\{B\}$ is the reference frame of the robot’s mobile base, and $\{H\}$ the reference frame of the arm’s end point. To simplify notation, all vectors in this section are defined in relation to reference frame $\{B\}$. Table I provides the Denavit-Hartenberg [23] configuration of the 3 DOF arm. This notation describes the relation between the reference frames of sequential actuated joints, as follows [24]:

- α_{i-1} : Angle from \hat{Z}_{i-1} to \hat{Z}_i measured along \hat{X}_{i-1} ;
- a_{i-1} : Distance from \hat{Z}_{i-1} to \hat{Z}_i measured along \hat{X}_{i-1} ;
- d_i : Distance from \hat{X}_{i-1} to \hat{X}_i measured along \hat{Z}_i ;
- θ_i : Angle from \hat{X}_{i-1} to \hat{X}_i measured along \hat{Z}_i .

Revolute joints always rotate around their respective \hat{Z}_i axis. Thus, θ_i becomes the position control variable of joint i , and the three other parameters depend on the structure of the robotic arm. In Fig. 2, $d_2 = L_1$ corresponds to the linkage between the first and second joints (measured along \hat{Z}_2 , which is parallel to \hat{Y}_B because $\alpha_1 = 90^\circ$), and $a_2 = L_2$ to the distance between the second and third joint.

TABLE I. DENAWIT-HARTENBERG CONFIGURATION FOR A 3 DOF ARM.

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	90°	0	L_1	θ_2
3	0	L_2	0	θ_3

Joint {1} is programmed with medium to high stiffness by setting impedance coefficient k_1 with a positive value, making Joint {1} simulate a torsion spring. θ_1 is set to 0 so that the shoulder is aligned in its neutral posture with the rest of the torso. External force can then be measured if a moment is applied to the arm. On the other hand, Joint {2} is configured with zero stiffness, allowing it to move in any direction with minimal effort and to keep its pose when no external forces are applied. This imposes the following constraint:

$$\vec{r}_{2H} \times \vec{f}_H = \vec{0} \quad (1)$$

where \vec{r}_{2H} is a vector between the reference frames of Joint {2} and {H}, and \vec{f}_H is the external force applied at the arm's end point. This implies that no external forces can be measured from Joint {2}. Joint {3} is also programmed with medium to high stiffness by setting impedance coefficient k_3 with a positive value, and therefore can provide force measurements. Its position, θ_3 , is set to a positive value so that the arm stays slightly bent in its neutral posture. However, because Joint {2} and Joint {3} are coplanar, the external force's component perpendicular to \vec{r}_{2H} does not generate a moment arm at Joint {3}'s axis. This is especially apparent when the arm is fully extended. When the arm is bent, forces applied parallel to \vec{r}_{2H} produce a moment arm and effectively stretch or compress the arm. This moment arm and its relation to the torsion spring simulated in Joint {3} is described by (2):

$$\|\vec{\tau}_3\| = (\vec{r}_{3H} \times \vec{f}_H) \cdot \hat{Z}_3 = k_3 \Delta\theta_3 \quad (2)$$

where $\|\vec{\tau}_3\|$ is the magnitude of the torque applied at Joint {3}'s axis, \vec{r}_{3H} is the vector between the reference frames of Joint {3} and {H}, \hat{Z}_3 is the normalized rotation axis of Joint {3}, and $\Delta\theta_3$ is the difference between Joint {3}'s current and neutral posture when slightly bent angles. Unfortunately, (2) cannot be solved for every components of \vec{f}_H , because a single torque measurement cannot fully describe the direction of the force applied to the arm's end point. As a solution, to interpret the component of the external force parallel to \vec{r}_{2H} , our approach considers a virtual spring attached between Joint {2} and the arm's end point, as shown in Fig. 3. L_2 is the distance between Joint {2} and Joint {3} (the upper arm), L_3 is the distance between Joint {3} and the arm's end point (the forearm), and l_A is the effective length of the arm when Joint {3} is bent. \vec{f}_A represents the external force applied on the arm's end point.

l_A is evaluated using (3):

$$l_A = \|\vec{r}_{2H}\| = \sqrt{L_2^2 + L_3^2 + 2L_2L_3 \cos(\theta_3)} \quad (3)$$

\vec{f}_A can be evaluated with a simple linear spring relation, as in (4):

$$\vec{f}_A = k_A \Delta l_A \frac{\vec{r}_{2H}}{\|\vec{r}_{2H}\|} \quad (4)$$

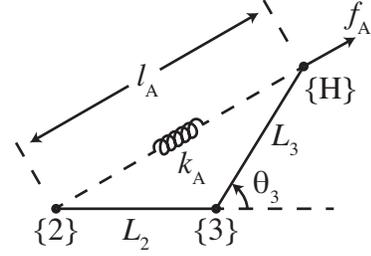


Fig. 3. A virtual spring attached from the arm's shoulder tilt joint (Joint {2}) to the arm's end point ({H}).

where k_A is the virtual spring constant, and Δl_A is the difference in the arm's effective length between its current and neutral posture angles. Because of constraint (1), \vec{f}_H lays on a plane containing both \vec{r}_{1H} and \vec{r}_{2H} . Thus, two orthogonal components can fully describe \vec{f}_H : \vec{f}_A given by (4), and \vec{f}_S which is parallel to both \hat{Z}_2 and \hat{Z}_3 . \vec{f}_H becomes a simple sum of both components:

$$\vec{f}_H = \vec{f}_A + \vec{f}_S \quad (5)$$

$\vec{\tau}_1$, the torque applied at Joint {1}, is given by (6):

$$\|\vec{\tau}_1\| = (\vec{r}_{1H} \times (\vec{f}_A + \vec{f}_S)) \cdot \hat{Z}_1 = k_1 \Delta\theta_1 \quad (6)$$

where $\Delta\theta_1$ is the angle between θ_1 and its neutral position. Since both \vec{f}_A from (4) and the direction of \vec{f}_S are known, (6) can be solved for \vec{f}_S using (7):

$$\vec{f}_S = (k_1 \Delta\theta_1 - \|\vec{r}_{1H} \times \vec{f}_A\|) \hat{Z}_3 \quad (7)$$

The effect of \vec{f}_H can be applied to the mobile base's reference frame to estimate the intended motion using (8):

$$\{\vec{f}_B; \vec{\tau}_B\} = \{\vec{f}_H; \vec{r}_{BH} \times \vec{f}_H\} \quad (8)$$

where \vec{r}_{BH} is the vector linking {B} and {H} reference frames. To simulate the dynamics of the mobile platform reacting to the estimated force and torque, a mass and damper model can be used, as described in [12]. However, to simplify implementation details, we linearly map force and torque to velocity commands, as described by (9):

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} k_f & k_f & 0 & 0 & 0 & 0 \\ k_f & k_f & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & k_\tau \end{pmatrix} \begin{pmatrix} \vec{f}_B \\ \vec{\tau}_B \end{pmatrix} \quad (9)$$

where \dot{x} and \dot{y} are the longitudinal velocity components, and $\dot{\theta}$ the angular velocity component. k_f and k_τ are the coefficients that linearly map force and torque applied to the base to linear and angular velocities, respectively.

III. IMPLEMENTATION ON IRL-1

Figure 4 shows IRL-1, i.e., our omnidirectional platform AZIMUT-3 on which two compliant arms and a humanoid torso are installed. Omnidirectionality of AZIMUT-3 is generated using steerable and drivable wheels with a lateral

offset from its attachment point [25]. Each wheel is made of a propulsion actuator and a steer actuator. Steering is done using DEAs. By having off-centered steerable wheels actuated using DEAs, a force applied at a wheel's contact point can be measured as torque on its steering axis. This is what makes physical interaction possible from almost any point on the platform (as long as the applied force is not parallel to all of the wheels' propulsion axes). The use of off-centered wheels also lowers the height of the chassis, and a passive vertical suspension made of four Rosta springs is used to connect the steerable wheels to the platform's chassis, thus helping to keep the wheels in contact with the ground on uneven surfaces. To make AZIMUT-3 move, all wheels must be precisely coordinated [26]: They must either all be oriented in the same direction, or have all their axes converge toward one point called the Instantaneous Centre of Rotation (ICR) of the platform. The robot's chassis represents a physical constraint on the rotation of the wheels around their steering axis [27]. These constraints introduce discontinuities on the steering angles of some wheels when the ICR moves continuously around the robot. In fact, a small change of the ICR position may require reorienting the wheels, such that at least one wheel has to make a full 180° rotation. This rotation can take some time because the platform needs to come to a stop. To limit such occurrences when the robot is moving at a relatively high linear velocity, in this implementation ICR position is limited to specific zones: Lateral movement was allowed without having to stop to reorient the wheels by restricting the overall direction of AZIMUT-3 to $(-45^\circ, +45^\circ)$ or, in velocity terms, when $|\dot{y}| < |\dot{x}|$.

IRL-1's arms are attached to the torso and have 4 DOF (three in the shoulder and one in the elbow), also actuated using DEAs. Impedance control of each joint enables an infinite combination of arm behaviors, from zero impedance for free movement with gravity compensation, to high stiffness constraining the arms to precise positions or ranges of movement. The arms can also sustain impacts with humans or objects. A gripping tool serves as the arm's end effector. Compared to the 3 DOF described in Section II, the arm's additional joint (shoulder roll) makes the forearm rotate around an axis that goes through the upper arm. To limit the effect of this additional joint on force estimation, it is programmed with high stiffness so that both shoulder and elbow tilt axes stay parallel as much as possible. Both arms can be used for force-guidance: The commands generated from both arms are simply summed, and the final command is limited by the platform's linear and angular velocity thresholds (0.45 m/s and 0.50 rad/s, respectively).

IRL-1's hardware architecture includes 20 distributed controller modules for sensing, power, and low-level control, which communicate with each other through a 1 Mbps CAN bus [28]. A 2.0 GHz Core 2 duo processor with 2 GB of RAM, is located in AZIMUT-3. A 2.67 GHz Core i7 quad core processor with 4 GB of RAM is located in front of the torso and is used for the arm controllers. Our approach for force estimation using compliant arms and velocity control is implemented as a ROS [29] node.

Table II presents the parameters used in our implementation. On IRL-1, k_4 and θ_4 corresponds to k_3 and θ_3 presented in Section II, because Joint $\{3\}$ of IRL-1's arms corresponds to

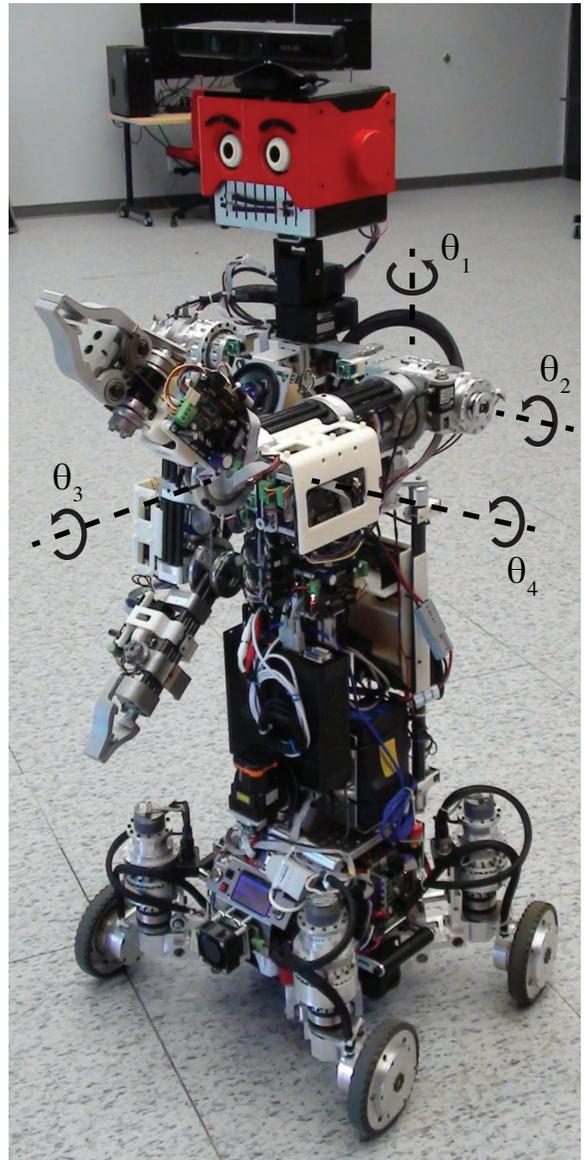


Fig. 4. IRL-1 and its right arm joint axes.

shoulder roll instead of elbow tilt. These parameters were set empirically to provide high sensitivity to small forces while avoiding jerkiness and keep the robot as stable as possible. Their influences were evaluated both by an experienced user and a pilot user who had very little experience with DPI, and who did not take part in the following test case.

TABLE II. SYSTEM PARAMETERS

Parameter	Value	Unit
k_1	10.00	Nm/rad
k_3	5.00	Nm/rad
k_4	5.00	Nm/rad
θ_1	0.00	rad
θ_4	0.90	rad
k_A	150.00	N/m
k_f	0.03	m/Ns
k_r	0.08	rad/Nms

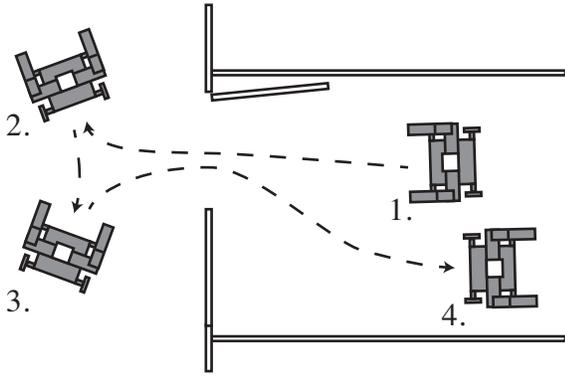


Fig. 5. Illustration of the task for the test case. IRL-1 is shown with its arms extended forward to represent its orientation.

IV. TEST CASE

The study and fine-tune the joint-space impedance control approach on IRL-1, and before conducting a usability study, we decided to use a test case to examine technical considerations regarding how people used DPI to guide IRL-1, which is non-holonomic (i.e., it cannot reorient its wheels instantaneously). Figure 5 illustrates the task used for the test case. It consists of guiding IRL-1 through a doorway by holding its arms' end points. We chose this task because it involves a simple and common scenario of guiding IRL-1 in or out of a room through a doorway. The maneuver requires guiding IRL-1 forward from a hallway (2 m wide) to a sparse room, make the robot turn around and get it through the doorway a second time. The doorway has an opening of 90 cm, and IRL-1 is 60 cm wide with its arms close to its body, leaving at most 15 cm of clearance on both side. The path to take once in the sparse room was not specified: Participants were left to find which path they were the most comfortable with. This allowed us to observe how users would naturally guide the robot, with no specific constraints on the path to produce.

Using this task, two conditions were tested: force-guiding IRL-1 using the approach described in Section II (DPI), or using a wireless gamepad to guide the robot (Gamepad). Both conditions followed the same ICR motion and maximum velocity constraints explained in Section III. Note however that the balance between linear and angular velocities with DPI did not allow the robot to turn on-the-spot, which may limit IRL-1's maneuverability compared to the Gamepad. Note also that using the Gamepad condition, the participants were allowed to move freely around the robot to get a better view of its surroundings. A Logitech Wireless Gamepad model F710 is used: The left analog stick controlled angular velocity ($\dot{\theta}$) by moving left and right, and the right analog stick controlled linear velocity (up/down for \dot{x} and left/right for \dot{y}). The objective pursued with these trials is to evaluate and fine-tune (mainly k_f and k_r) the approach on IRL-1, before engaging into a usability study on DPI. The Gamepad evaluation therefore serves as a reference measure of what can be expected of IRL-1 when longitudinal and angular motion can be precisely controlled.

A convenience sample of 15 participants (engineering researchers aged from 21 to 36, $\bar{x} = 26.2$, $\sigma = 4.2$, 14 male

and 1 female) took part in our test case. Four of those 15 participants were involved 4 months earlier in our previous study, implementing DPI using force/torque sensed by AZIMUT-3's steerable wheels [16]. Participants were allowed up to 5 min to familiarize with each condition before doing their trials. They performed five trials with each condition. The order in which participants used each condition was chosen randomly so that one half used DPI first and then Gamepad, and the other half used Gamepad first and then DPI. Time to complete the task was measured, and participants were asked to evaluate the following sentences for each condition.

- Q1: IRL-1's responsiveness was considered to be very low (1), low (2), correct (3), high (4), or too high (5).
- Q2: It was easy to move the robot on a straight line (1 = strongly disagree, 3 = neutral, 5 = strongly agree).
- Q3: It was easy to reorient the robot (1 = strongly disagree, 3 = neutral, 5 = strongly agree).
- Q4: It was hard to predict the robot's response (1 = strongly disagree, 3 = neutral, 5 = strongly agree).

Participants were also encouraged to leave comments on their questionnaire sheet about their overall experience.

V. RESULTS

Figure 6 illustrates typical paths taken by participants to accomplish the task, which were influenced by user preferences, spatial limitations (e.g., turning while force-guiding IRL-1 requires more space compared to using the Gamepad), and control capabilities. For instance, making IRL-1 change orientation was executed by guiding the robot to make a large loop (always moving forward) or a three-point turn (making the robot move forward and backward), while distinct control of the longitudinal and angular velocities allowed the robot to turn on-the-spot. This influenced the time required to accomplish the task. Figure 7 summarizes the time measurements over all trials, along with the shortest and longest trials recorded per participant. The vertical lines in the boxes represent the medians. Clearly, the task is accomplished faster when using the Gamepad. The average time using the Gamepad is 39.7 s ($\sigma = 10.7$ s) compared to 45.5 s ($\sigma = 19.1$ s) with DPI. The average shortest time over trials per participant is 23.1 s ($\sigma = 6.5$ s) using the Gamepad and 42.3 s ($\sigma = 13.2$ s) with DPI, and the average longest performances is 35.8 s ($\sigma = 12.0$ s) using the Gamepad and 68.3 s ($\sigma = 20.2$) with DPI. Only one participant had its best performance with DPI, with 23.9 s compared to 24.5 s using the Gamepad. The average difference between best performances in both conditions is 19.22 s ($\sigma = 9.9$ s).

Using the Gamepad, the fastest performance was recorded using the path shown in Fig. 6(c), which was the shortest possible one to complete the task: It involved only one stop. However, with DPI, the fastest performance was recorded with 6(a) instead of the three-point turn path represented by Fig. 6(b). The latter requires more frequent stops to reorient IRL-1's wheels, which resulted in longer time to completion. Reorientation of AZIMUT-3's wheels occurred more often

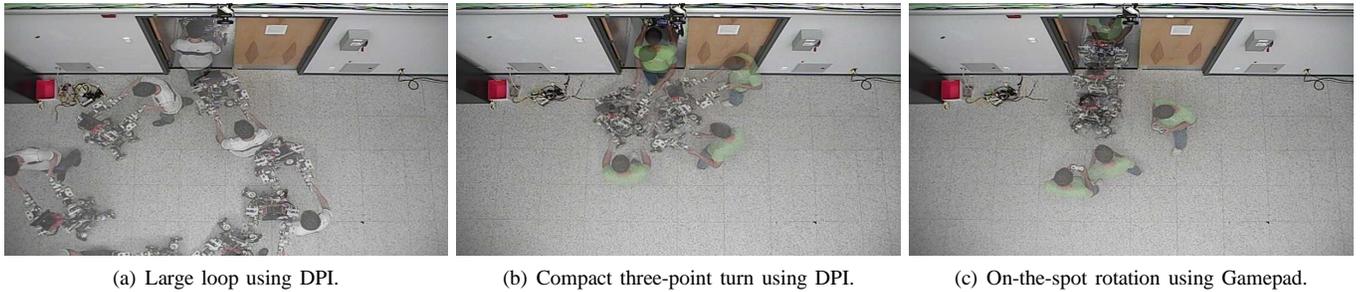


Fig. 6. Three paths used to accomplish the task.

TABLE III. RESULTS FROM THE QUESTIONNAIRE

Question	DPI		Gamepad	
	\bar{x}	σ	\bar{x}	σ
Q1. Responsiveness	2.47	0.64	3.07	0.46
Q2. Easy to move in a straight line	4.13	0.74	4.67	0.49
Q3. Easy to reorient	2.67	0.82	4.40	0.51
Q4. Hard to predict	2.80	1.01	2.07	0.80

when a shorter turning radius was desired, and k_f and k_τ were set to limit such occurrences. The same wheel reorientation problem applied to the Gamepad condition, but it was less apparent because participants rarely combined both linear and angular velocities: Participants made IRL-1 go forward, stop, turn on itself, and go forward again. Thus, stopping to reorient the wheels did not come as a surprise in these situations: Participants already planned to stop the platform. On the other hand, it was more jarring when a stop happened when the linear velocity is progressively diminished while applying constant angular velocity, because IRL-1 interrupted what the user wanted to communicate to the platform, i.e., a smooth and fluid turn motion.

Table III summarizes the results gathered with the questionnaire. Regarding IRL-1’s responsiveness (Q1), most participants found it to be a bit low with DPI, and correct using the Gamepad. To improve responsiveness, k_f and k_τ can be set higher: The robot would then respond with greater motion to smaller force inputs. This would also require increasing maximum velocity limits. In addition, one participant commented that it was much easier to learn IRL-1’s velocity limits using the Gamepad. Indeed, the linear and angular velocity limits are directly tied to the physical limits of the analog sticks of the Gamepad, whereas learning the limits through DPI is more subtle and can be potentially dangerous for the robot. For instance, the linear velocity limit using DPI is clearly felt when an operator pulls the arms harder than necessary, which makes the robot tilt forward since the mobile base cannot compensate for the intended speed. Because such pull is usually only possible when the arms are extended, this puts the centre of gravity of the robot high and forward, increasing the risk of tipping over. This is intrinsically linked to IRL-1’s design, and raises the issue of finding a balance between having linear velocity limits set low for normal control situations, and high to prevent the robot from falling over. Regarding Q2, nobody seemed to experience difficulties moving the robot on a straight line in both conditions, with again a preference for the Gamepad. To reorient IRL-1 however (Q3), most participants found DPI difficult to use. Not being able to use DPI to make

the robot rotate on-the-spot certainly influences this result. But not having to experience reorientation of IRL-1’s wheels as frequently using the Gamepad compared to DPI mostly explains the situation. In fact, most participants complained verbally when the robot had to stop to reorient its wheels, in both conditions. Two also specifically commented these occurrences on their questionnaire sheets. The same reason may explain why five participants found the robot’s actions difficult to predict with DPI compared to only one using the Gamepad (Q4).

Regarding the comments gathered from the participants, two of them mentioned they preferred DPI in tight situations around the doorway. One of them mentioned this was because he could better focus its attention on the robot’s wheels because he knew exactly where its arms were located, while he had to be visually aware of both with the Gamepad. Because of their lateral offset and frequent reorientations at low velocities, AZIMUT-3’s wheels sometimes became (in two or three occasions overall and in both conditions) a moving point of contact for nearby obstacles (e.g., in the doorway). In addition, one participant mentioned he felt nervous at first with DPI, but found that he could rapidly learn to guide it. Finally, participants who took part in our previous study commented that they found the new approach to be better than the previous one. One of them thought that it would have been difficult to go through the doorway with the previous approach, which required much more force to reorient the robot.

VI. DISCUSSION

The test case conducted demonstrates that while our approach works, improvements need to be made before going further with a usability study on DPI.

Contrarily to Chen & Kemp’s findings [4], DPI did not outperform the Gamepad in our trials. Chen et al.’s direct physical interface basically transposes gamepad controls onto the 6 DOF force/torque sensors placed at the arms’ end points, and only uses 1 DOF of the arm to make the robot move sideways. Two hands are required to make the robot turn and move sideways, and also to control the gamepad. Velocities are derived for each arm acting like the gamepad controller would, and the maximum magnitude over both arms is used to command the robot. This made comparison with a gamepad controller simpler, because both followed a similar control policy.

In that regard, our approach is more complex because the user does not control motion velocities independently. It also

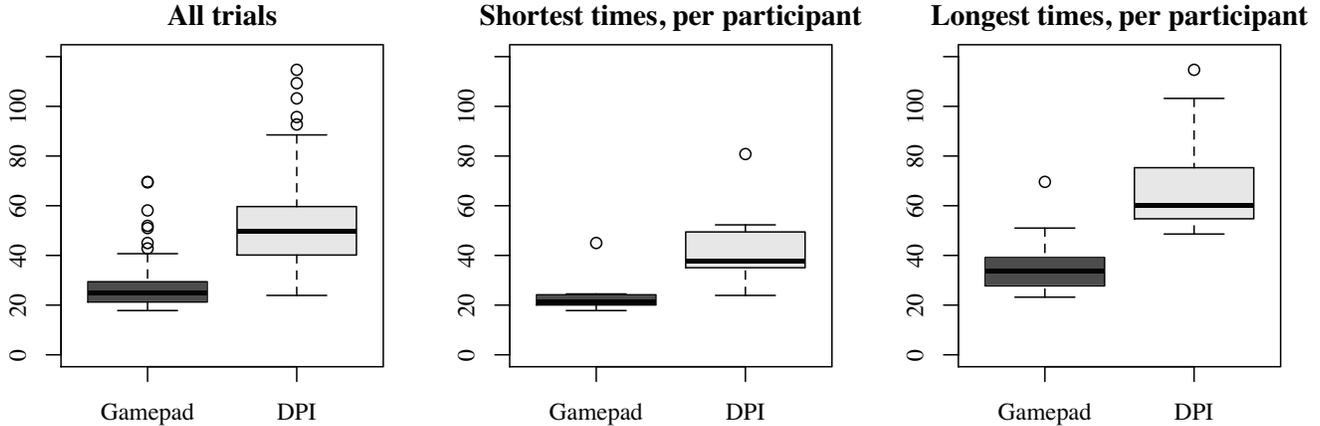


Fig. 7. Time to complete the task, using the Gamepad and with DPI.

does not use a 6-axis force/torque sensors at the arms' end points and can estimate applied forces (not necessarily only the ones applied on the arms' end points but from anywhere on the arms) by controlling stiffness on a subset of impedance-controlled joints. This makes it possible to have the robot react as a real-life physical object to a user pulling and pushing on one or both of its arms. For instance, when pulling on one arm, the robot moves as if it is supported by passive, low-friction castor wheels, and thus rotates until its arm is aligned with its centre of gravity. To move straight ahead requires an equal force applied to both arms, effectively cancelling the torque coming from each side, unless considerable torque is applied around one of the arm's end point. Furthermore, while the shoulder tilt angle (θ_2) is not used to estimate the magnitude of the force applied at the arm's end point, it still has an impact on the direction of \vec{r}_{2H} and thus the orientation of the plane in which \vec{f}_H is constrained. This implies that the magnitude of the produced velocity command tends toward zero when the arm is progressively tilted up or down. We believe this to be a desirable effect of replicating the dynamics of a real life physical object. Otherwise, pulling one arm of the robot toward the ground would make it go forward which is, we find, not particularly intuitive. This is different from Chen & Kemp's approach [4], [5], where pulling straight ahead on the robot's end effector does not produce any rotation even if the arm has a lateral offset from the robot's centre of gravity. In addition, the robot's arms do not have to stay in a home position: An operator can keep them at a comfortable height, and pull them down when they are not needed or would interfere with an obstacle.

Finally, not imposing constraints on the path taken by participants led to significant differences over the trials, but allowing us to observe IRL-1's maneuverability and gain insights regarding experimental conditions for the upcoming usability study. As a consequence, Gamepad results should not be used as a comparison with DPI but rather as a reference to what can be accomplished by the platform, allowing to identify the following adjustments:

- Minimizing having to stop to reorient IRL-1's steerable wheels. This issue is caused because AZIMUT-3 is non-holonomic (while the robot used by Chen &

Kemp [4], [5] is holonomic), and having the robot stop to reorient its wheels influences DPI usability. In [12], lateral velocity was not allowed ($\dot{y} = 0$) and turning radius was limited to a minimum of 35 cm, which eliminated making the platform stop to reorient the wheels. With our approach, lateral velocity control is less restricted ($|\dot{y}| < |\dot{x}|$), but limiting the turning radius to a minimum of 35 cm would help minimize wheels' reorientation.

- Making the robot turn on-the-spot with DPI. Based on a suggestion made by one participant, making the robot rotate on-the-spot could be done by lifting its arm toward the ceiling, like a dancer would do with his partner. This would be possible by monitoring the torque on IRL-1's shoulder pan joint when its shoulder tilt angle is over a certain threshold.

VII. CONCLUSION AND FUTURE WORK

This paper presents a joint-space impedance control approach for force-guiding a humanoid mobile robot using its compliant arms. Our approach allows a user to guide motion of a non-holonomic omnidirectional robot simply by applying forces on one or both arms. The trials conducted demonstrate feasibility of the approach, and identify issues (such as wheel reorientation, on-the-spot control) and fine-tuning to be made in an iterative design methodology. The approach could also be applied on other humanoid mobile robot with compliant arms. In future work, we plan to conduct a usability study with different control schemes (e.g., joint-space impedance control, cartesian impedance control with a fixed home position, direct transposition of the velocity controls on the arms' DOF, Gamepad) going through a constrained obstacle course imposing similar wheel reorientation conditions. The estimation of force/torque derived from the arms could also be combined with the applied force/torque perceived from the steerable wheels, as demonstrated in [12] and [16], to allow more precise DPI from any locations on the robot. We are also currently integrating DPI within our Hybrid Behavior-Based Architecture (HBBA) [16], allowing a person to influence the autonomous navigation capabilities of IRL-1.

ACKNOWLEDGEMENTS

This work is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), the Canada Research Chair (CRC) in Mobile Robotics and Autonomous Intelligent Systems, and the Fonds de recherche du Québec – Nature et technologies (FRQ-NT).

REFERENCES

- [1] F. Michaud, T. Salter, A. Duquette, and J.-F. Laplante, “Perspectives on mobile robots used as tools for pediatric rehabilitation,” *Assistive Technologies, Special Issue on Intelligent Systems in Pediatric Rehabilitation*, pp. 14–29, 2007.
- [2] F. Michaud and C. Theberge-Turmel, “Mobile robotic toys and autism,” in *Socially Intelligent Agents - Creating Relationships with Computers and Robots*, K. Dautenhahn, A. Bond, L. Canamero, and B. Edmonds, Eds. Kluwer Academic, 2002.
- [3] T. Takeda, Y. Hirata, and K. Kosuge, “Dance step estimation method based on HMM for dance partner robot,” *IEEE Transactions on Industrial Electronics*, vol. 54, no. 2, pp. 699–706, april 2007.
- [4] T. L. Chen and C. C. Kemp, “A direct physical interface for navigation and positioning of a robotic nursing assistant,” *Advanced Robotics*, vol. 25, pp. 605–27, 2011.
- [5] —, “Lead me by the hand: Evaluation of a direct physical interface for nursing assistant robots,” in *Proceedings of the 5th ACM/IEEE International Conference on Human-Robot Interaction*, 2010, pp. 367–374.
- [6] A. Jain and C. C. Kemp, “Pulling open novel doors and drawers with equilibrium point control,” in *Proceedings of the IEEE-RAS International Conference on Humanoid Robotics*, 2009.
- [7] K. Wyrobek, E. Berger, H. Van der Loos, and K. Salisbury, “Towards a personal robotics development platform: Rationale and design of an intrinsically safe personal robot,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2008.
- [8] C. Borst, T. Wimbrock, F. Schmidt, M. Fuchs, B. Brunner, F. Zacharias, P. R. Giordano, R. Konietzschke, W. Sepp, S. Fuchs, C. Rink, A. Albuschaffer, and G. Hirzinger, “Rollin’ Justin – Mobile platform with variable base,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2009, pp. 1597–1598.
- [9] M. Williamson, *Series Elastic Actuators*. Master’s thesis, Massachusetts Institute of Technology, Cambridge, Boston, 1995.
- [10] D. Robinson, *Design and Analysis of Series Elasticity in Closed Loop Actuator Force Control*. Doctoral thesis, Massachusetts Institute of Technology, Cambridge, Boston, 2000.
- [11] M. Lauria, M.-A. Legault, M.-A. Lavoie, and F. Michaud, “Differential elastic actuator for robotic interaction tasks,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2008, pp. 3606–3611.
- [12] J. Frémy, F. Michaud, and M. Lauria, “Pushing a robot along – A natural interface for human-robot interaction,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2010, pp. 3440–3445.
- [13] J. Frémy, F. Ferland, L. Clavien, D. Létourneau, F. Michaud, and M. Lauria, “Force-controlled motion of a mobile platform,” in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, oct. 2010, pp. 2517–2518.
- [14] S. P. Buerger and N. Hogan, “Impedance and interaction control,” in *Robotics and Automation Handbook*. CRC Press, 2005, pp. 19.1–19.24.
- [15] N. Hogan, “Impedance control: An approach to manipulation: Part II - Dynamics systems measurement control,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, p. 17, 1985.
- [16] F. Ferland, D. Létourneau, A. Aumont, J. Frémy, M.-A. Legault, M. Lauria, and F. Michaud, “Natural interaction design of a humanoid robot,” *Journal of Human-Robot Interaction*, in press.
- [17] R. Ikeura, T. Moriguchi, and K. Mizutani, “Optimal variable impedance control for a robot and its application to lifting an object with a human,” in *Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication*, 2002, pp. 500 – 505.
- [18] T. Takubo, H. Arai, Y. Hayashibara, and K. Tanie, “Human-robot cooperative manipulation using a virtual nonholonomic constraint,” *The International Journal of Robotics Research*, vol. 21, no. 5-6, pp. 541–553, 2002.
- [19] D. Lee, C. Ott, and Y. Nakamura, “Mimetic communication with impedance control for physical human-robot interaction,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2009, pp. 1535–1542.
- [20] M. Mason, “Compliance and force control for computer controlled manipulators,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 11, no. 6, pp. 418–432, 1981.
- [21] R. Paul, *Modelling, Trajectory Calculation and Servoing of a Computer Controlled Arm*. Doctoral thesis, Stanford University, Stanford, CA, 1972.
- [22] C. Ott, B. Büuml, C. Borst, and G. Hirzinger, “Employing cartesian impedance control for the opening of a door: A case study in mobile manipulation,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems Workshop on Mobile Manipulators: Basic Techniques, New Trends & Applications*, 2005.
- [23] J. Denavit and R. Hartenberg, “A kinematic notation for lower-pair mechanisms based on matrices,” *Journal of Applied Mechanics*, pp. 215–221, June 1955.
- [24] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 3rd ed. Pearson Prentice Hall, 2005, ch. 3, pp. 68–69.
- [25] M. Lauria, I. Nadeau, P. Lepage, Y. Morin, P. Giguere, F. Gagnon, D. Létourneau, and F. Michaud, “Kinematical analysis of a four steered wheeled mobile robot,” in *Proceedings IEEE International Conference on Industrial Electronics*, vol. 4, 2006, pp. 3090–3095.
- [26] L. Clavien, M. Lauria, and F. Michaud, “Instantaneous center of rotation estimation of an omnidirectional mobile robot,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2010.
- [27] S. Chamberland, É. Beaudry, L. Clavien, F. Kabanza, F. Michaud, and M. Lauria, “Motion planning for an omnidirectional robot with steering constraints,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.
- [28] F. Michaud, D. Létourneau, M. Arseneault, Y. Bergeron, R. Cadrin, F. Gagnon, M. Legault, M. Millette, J. Pare, M. Tremblay, P. Lepage, Y. Morin, J. Bisson, and S. Caron, “Multi-modal locomotion robotic platform using leg-track-wheel articulations,” *Autonomous Robots*, vol. 18, no. 2, pp. 137–156, 2005.
- [29] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, “ROS: An open-source Robot Operating System,” in *Open-Source Software Workshop of the International Conference on Robotics and Automation*, 2009.