Feasible Stylized Motion: Robotic Manipulator Imitation of a Human Demonstration with Collision Avoidance and Style Parameters in Increasingly Cluttered Environments

Roshni Kaushik
Carnegie Mellon University
Pittsburgh, PA, USA
roshnika@andrew.cmu.edu

Anant Kumar Mishra
Siemens Corporation
Charlotte, NC, USA
anant.mishra@siemens.com

Amy LaViers
University of Illinois at
Urbana-Champaign
Urbana, IL, USA
alaviers@illinois.edu

ABSTRACT

Socially intelligent robots are a priority for large manufacturing companies that want to deploy collaborative robots in many countries around the world. This paper presents an approach to robot motion generation in which a human demonstration is imitated, collisions are avoided, and a “style” is applied to subtly modify the feasible motion. The framework integrates three subsystems to create a holistic method that navigates the trade-off between form and function. The first subsystem uses depth camera information to track a human skeleton and create a low dimensional motion model. The second subsystem applies these angles to a simulated UR3 robot, modifying them to produce a feasible trajectory. The generated trajectory avoids physically infeasible configurations and collisions with the environment, while remaining as close to the original demonstration as possible. The final subsystem applies four style parameters, based on prior work using Laban Effort Factors, to endow the trajectory with a specific “style”. This approach creates adaptive robot behavior in which one human demonstration can result in many subtly different robot motions. The effectiveness of the hybrid approach, which considers functional as well as expressive goals, is demonstrated on three environments of increasing clutter. As expected, in more cluttered environments, the desired imitation is not as pronounced as in unconstrained environments. Potential applications of this framework include programming robot motion on a factory floor with greater efficiency as well as creating feasible motion on multiple robots with a single demonstration. This quantitative work highlights the Function/Expression duality named in the Laban/Bartenieff Movement System, illuminating how the arts are critical for “practical” spaces like the factory.

CCS CONCEPTS

• **Human-centered computing** → **Collaborative and social computing devices; Interaction design process and methods; Collaborative interaction.**

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

MOCO '20, July 15–17, 2020, Jersey City/ Virtual, NJ, USA

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7505-4/20/07...\$15.00

<https://doi.org/10.1145/3401956.3404188>

KEYWORDS

robotics, imitation, style, coordination, social robots, collaborative robots, manufacturing, Laban Movement Analysis, optimization

ACM Reference Format:

Roshni Kaushik, Anant Kumar Mishra, and Amy LaViers. 2020. Feasible Stylized Motion: Robotic Manipulator Imitation of a Human Demonstration with Collision Avoidance and Style Parameters in Increasingly Cluttered Environments. In *7th International Conference on Movement and Computing (MOCO '20)*, July 15–17, 2020, Jersey City/ Virtual, NJ, USA. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3401956.3404188>

1 INTRODUCTION

With many less degrees of freedom and a differing morphology from a human, robots cannot perfectly replicate human motion. Thus, enabled by this mismatch in dimensionality, a robot can imitate a human’s motion in many different ways with varying success. However, tackling questions such as *which part of human motion is important to imitate* and *how should the robot imitate feasibly* can take us closer to creating a framework for robot imitation. Imitation is an important interaction mode between humans, as discussed below, and developing methods for robot-human imitation will likely contribute significantly towards the goal of humans and robots working and interacting in shared spaces.

Robots are used currently and will be used to an even greater extent in the future to automate various tasks on a factory floor, ranging from dangerous welding tasks to drilling and polishing. To program these robots to perform these tasks effectively, researchers must develop methods for programming robots these tasks. Additionally, in manufacturing environments, robots will be performing tasks side-by-side with humans. We aim to design robots capable of signaling human counterparts through movement that indicates attention, change of task, adaptation to new instructions, and other ambient signals that may facilitate collaboration in parallel to completing functional tasks. The ability to do this in different cultural and contextual situations, such as in factories producing different products in locations around the world, is vital to the success of the robots’ deployment at these facilities.

The combination of the robot performing its function properly, expressing its intent appropriately, and adapting to the human exemplifies the function/expression duality in the Laban/Bartenieff system. This highlights the need for an understanding of human motion and an integration of concepts from the arts even in “practical” spaces like the factory.

In this paper, we address creating feasible, stylized robot motion that imitates a human demonstration. First, we look at creating a model for robot motion from a human motion demonstration, which requires selecting which aspects of human motion to imitate. We then modify this robot trajectory to create feasible robot motion given objects in the environment and the robot’s physical limitations, while keeping close to the original demonstration. Lastly, we incorporate style parameters based on Laban’s efforts to subtly modify the movement and create more possible variability in human perception of the robot’s resulting motion. This framework will demonstrate trade-offs between style and imitation success in environments with different functional requirements due to varied obstacles.

We first overview our methodology for generating feasible, stylized robot motion imitating a human demonstration in Section 3. We include examples of results in three environments and with different styles in Section 4. We conclude and discuss future directions and potential applications of this work in Section 5. In our robot-human imitation framework, we do not claim to create anthropomorphic movement, due to the large disparity between the physical structure and capabilities of a robotic platform and a human. We use the human movement to tap into the movement conventions of human, which the robot can use to appear to adapt to the environment and contextual situation.

The first aspect to this work includes modeling human motion and generating robot motion that imitates that motion. Imitation is an important modality of human interaction, as seen in foundational studies in psychology [3]. Research on human movement during conversation indicates that patterns in the formation and breaking of symmetry in head angle of the two interactants in conversations facilitate human perception of communication; interactants that received artificially interrupted symmetry patterns reported less successful communication [4]. Thus, developing methods for robot-human imitation will likely contribute significantly towards the goal of humans and robots working and interacting in shared spaces.

Robot imitation of human movements has been implemented in many previous works. Prior research used low-degree-of-freedom (low-DOF), non-humanoid robots to imitate human motion capture data using the changing angle of the spine [7, 8]. Other works have emulated human behaviors during social interactions by creating a mapping that virtually connects elements of estimated body pose and points on the robot with translational springs [14]. Optimization subject to dynamic and physical constraints of the robot were used to create a mapping between motion capture and a humanoid robot [19].

Imitation learning is an approach in which a policy is learned by an artificial agent by observing, modeling, and reproducing a desired behavior. By tracking users’ movements and their interactions with the environment, an approach using imitation learning was used for virtual character animation [6]. An imitation learning framework for a humanoid robots was developed using a combined control and learning architecture to produce movement sequences [2]. These methods have been developed for long-term skill development rather than immediate, real-time imitation through vision-based sensors.

An allied field to the work developed in this paper is learning from demonstration (LfD), a technique that uses multiple successful executions of a task to abstract a model of that task. LfD has been used to teach a robot how to draw letters from varying start positions by physically moving a robot through many successful drawings of the letter [16]. Researchers used learning from demonstration to program a virtual dancer by developing an internal model of a human dancer’s movements using Artificial Neural Networks (ANN) and Hidden Markov Models (HMM) and reacting to some movements from a human dancer [12].

Another component of this work is the application of “style” parameters to robot movement. Style is a subjective impression of a movement by the viewer. Quantitative analysis has been performed on styles of human motion, such by comparing the dance forms of Kathak and Flamenco [15].

Other work has investigated the notion of style and character in developing robotic motion for human-facing environments. Researchers used the low-DOF head motions of the Keepon and Nao robots to engage human interactants by using Laban efforts [9, 17]. A methodology for creating correspondences between groups of body parts on different characters re-targeted the movement of virtual characters using a model of style [1]. A mapping between the human poses and the character poses can be generated by using a few key human poses from a motion capture session of a human moving in the style of the character [20].

2 APPROXIMATING THE LABAN EFFORT SYSTEM

This paper leverages a prior mapping from the Laban Movement Analysis system of Effort, a systematic description of motion quality that is part of the Laban/Bartenieff Movement System, to a quantitative model. This underpinning movement theory will be reviewed in this extended background section along with simplifications used to inspired the method presented here. Namely, a prior mapping to an optimal control problem endows a motion trajectory with subtle variation, also discussed as “style” or “quality”, as in [11]; this specific formulation is reviewed in Sec. 3.3.

An important approach to formalizing movement is taken by Rudolf Laban, whose dance notation system explicates principles of the body’s geometry and movement’s dynamic quality [5, 18]. Specifically, we examine his system of movement *Effort* [10]. These parameters are derived from Laban’s *Effort* or *Motion Factors of Space, Weight, Time, and Flow*. A relationship between these parameters is shown in Fig. 1.

The so-called “dynamosphere” in Fig. 1 lists the eight Basic Effort Actions *Dabbing, Gliding, Floating, Flicking, Thrusting, Pressing, Wringing, and Slashing*, and the *Effort Factors* explicate a relationship between these motion qualities. For example, the difference between *Dabbing* and *Flicking* is via the *Space Effort* – the former utilizes a *Direct* relationship to *Space* while the latter utilizes an *Indirect*. We might see a movement like *Dabbing*, with its *Light Weight, Direct Space, and Sudden Time* as the opposite of *Wringing*, with its *Strong Weight, Indirect Space, and Sustained Time*, thus, these qualities are on opposite corners of this cube.

In developing a quantitative framework for generating these ideas, on low-dimensional machines, we must sacrifice and approximate. We note the sense of linearity associated with *Direct*

Figure 1: The dynamosphere, showing the combination of three of Laban’s efforts along the axes of *Space*, *Weight*, and *Time*. These three efforts relate to a single movement, and combined with *Flow*, can describe the quality of a movement sequence [11, 13].

Space, likening this idea to the tracking of the most immediate path between two motion states. We note the sense of energetic engagement in *Strong Space*, likening this idea to a sense of energy, which is related to the acceleration of motion. We also note the large change over a short period of time associated with *Sudden Time*, likening this idea with the magnitude of change in state. Finally, we approximate the concept of *Flow Effort*, a complex phenomenon that is often described as a sense of liveliness and life in living creatures, with the connection between a series of movements, relating this idea to the clarity of where one action stops and another starts. Using these simplified ideas, Sec. 3.3 introduces a quantitative model for “adding” subtle shifts of style through the lens of *Effort*.

3 METHODOLOGY

The goal of this line of work is to develop a comprehensive approach to imitate a human motion demonstration with a robot arm using both collision avoidance and style parameters. Thus, the framework should both imitate the human demonstration, avoid infeasible behaviors for the new, robotic platform, and be able to modify the resulting execution subtly for sending different perceptual cues to human observers. This section describes an initial framework that has several of these features.

The framework is divided into a three subsystem hierarchy (Figure 2). The robot used throughout this work is the UR3 by Universal Robots. The first subsystem (S_1) creates a three degree-of-freedom (DOF) model that imitates a particular aspect of the human motion demonstration. The next subsystem (S_2) modifies this 3-DOF model to generate a robot joint trajectory that avoids self-collisions and obstacles while remaining close to the initial motion. The final subsystem (S_3) applies four style parameters based on Laban motion factors on a robot joint trajectory to change the reference motion quality.

3.1 Subsystem 1: Human Model

The first subsystem (S_1) computes a robot joint trajectory that imitates a human motion demonstration. Using the Microsoft Kinect

Figure 2: An input (orange) of a motion demonstration to a series of subsystems (green) results in an output (orange) of a feasible robot joint trajectory imitating the original human motion with applied style parameters, with intermediate outputs shown in blue. These three subsystems are abbreviated S_1, S_2, S_3 in this paper.

depth camera and the `openni_tracker` ROS package, we first obtained a 15-dimensional skeleton of the human demonstrator.

We calculate three angles ($\theta_1, \theta_2, \theta_3$) to create a low-DOF human model by first choosing four features from the total of fifteen on the skeleton model. These four can be chosen arbitrarily, but based on previous work in [7, 8] and the rough similarities between human and robot arms, we choose four signals corresponding to the left and right arms to compute the model (left arm angles shown in Figure 3). Each point (a, b, c, d) is in $\mathbb{R}^3$, and we define three vectors as the component-wise difference between pairs of points (i.e. $\vec{ab} = b - a, \vec{bc} = c - b, \vec{cd} = d - c$). Other points could be defined on the body, such as using the hip, knee, and foot to mimic changing angles in the leg, but in this implementation, we only use the left arm and right arm.

We then define $g(\vec{v}_1, \vec{v}_2) : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}$ as the angle between two generic vectors $\vec{v}_1, \vec{v}_2$ (Equation 1). We can compute the three angles for the human motion model using Equation 2, where $\hat{i}$ is unit vector in x-direction.

$$g(\vec{v}_1, \vec{v}_2) = \cos^{-1} \left(\frac{\vec{v}_1}{|\vec{v}_1|} \cdot \frac{\vec{v}_2}{|\vec{v}_2|} \right) \quad (1)$$

$$\begin{aligned} \theta_1 &= g(\vec{a}, \vec{b}) \\ \theta_2 &= g(\vec{a}, \vec{c}) \\ \theta_3 &= g(\vec{b}, \vec{c}) \end{aligned} \quad (2)$$

These three angles ($\theta_1, \theta_2, \theta_3$) are the output of S_1 , which takes a human motion demonstration as input to create a 3-DOF model that imitates a part of the human motion.

Figure 3: We choose 4 points (a, b, c, d) on the human skeleton model (left) and compute three angles ($\theta_1, \theta_2, \theta_3$) from the total 15 joints given by the `openni_tracker` package’s processing of the Kinect camera input. These three angles form the bases for the control of three joints (shoulder pan, shoulder lift, elbow) of the UR3 robot (right). These Rviz visualizations are what the user of this framework sees when recording the human motion demonstration (left) and animating robots with the calculated trajectories (right).

3.2 Subsystem 2: Collision Avoidance

In this subsystem, we control three joints (shoulder pan, shoulder lift, and elbow) of the UR3 robot (Figure 3). In our current approach, we set the three wrist joint angles of the robot to 0 for all trajectories, but future work could modify these angles as well, especially with more sophisticated human body tracking.

The human body has differing morphology and mobility from the robot, so a specific joint angle the human reaches may not be feasible for the robot. Additionally, since the human demonstrates the movement in a given space and the robot might imitate the

movement with different obstacles present, collision avoidance must be calculated for the robot in its given environment.

Thus, this subsystem (S_2) modifies an input joint trajectory based on the collected human demonstration (α) for the UR3 robot to avoid collisions while preserving as much as possible of the original input motion. Here α is a combination of identified human motion features we want to imitate, $\theta_1, \theta_2, \theta_3$; thus $\alpha \in \mathbb{R}^3$. Specifically, we subtract π from θ_3 to account for the differences in how the angle is measured on the UR3 before performing the feasibility modification. In summary, θ_1 drives the shoulder pan joint, θ_2 drives the shoulder lift joint, θ_3 (with modification) drives the elbow joint, i.e., $\alpha = [\theta_1, \theta_2, \theta_3 - \pi]^T$.

We use the ROS library Moveit along with a Gazebo model of the UR3 robot in this implementation. Using the `PlanningSceneInterface` and `CollisionObject` classes defined in Moveit, we initialize an environment around the robot with obstacles. We next check the validity of the input joint trajectory (α) using the service `check_state_validity` that outputs whether each element of α is a feasible configuration for the robot without self-collisions or collisions with any obstacles in the environment.

After using this checker, we now may have several elements, or sequences of points, in α that the planner determines are impossible for the robot to execute. We replace each sequence of infeasible points in α with a feasible sequence. Let $\beta \subset \alpha$ contain n infeasible consecutive points with indices $i, i+1, \dots, i+n-1$ in α .

We first consider the case where β does not contain the first or last points of α . Using the Moveit planning interface (OMPL planner), we compute a feasible joint trajectory between the feasible points immediately around β : between α_{i-1} and α_{i+n} . This joint trajectory γ from OMPL avoids any collisions and connects two feasible sequences in α , but it may contain m points where $m \neq n$ and therefore cannot immediately replace β in α .

We resample the points in γ using the Python `scipy` library suite to obtain a new trajectory γ' with n points, which can then replace the infeasible β in our original trajectory α . To handle the endpoint cases where the first or last points in α are infeasible, we trim the trajectory until both endpoints are feasible before proceeding with the infeasible sequence replacement, though extrapolation methods can also be used from interior feasible points to tackle this case.

We lastly smooth the resulting trajectory using a spline to prevent any sudden spikes in either the input (Kinect data can be noisy) or the trajectory splicing just performed. The output of S_2 , $\sigma = [\sigma_1, \sigma_2, \sigma_3]^T$, is a feasible joint trajectory for the UR3 robot that preserves as much as possible of the original input trajectory.

3.3 Subsystem 3: Style

This subsystem (S_3) applies four parameters to a robot joint trajectory that change the quality of the motion. These four parameters correspond to the four categories of Laban *Effort*: *Space*, *Weight*, *Time*, and *Flow*. A motion can combine different values of these parameters, similar to moving in $\mathbb{R}^4$. Changes in these parameters correspond to moving on along an axis between two motion descriptors: flexibility and directness in *Space*, light and strong in *Weight*, sudden and sustained in *Time*, and free and bound *Flow* [11]. The relationship between *Space*, *Weight*, and *Time* is shown in Figure 1. The approach used in this paper to mathematically

describe these motion factors and modify a reference trajectory using optimal control is briefly reviewed below.

Review of mapping between kinesthetic notions of style and optimal control. The mathematical formulation here is reviewed from [11]. Consider a system with input $u = [u_1, \dots, u_m]^T$, state $x = [x_1, \dots, x_n]^T$, and output $y = [y_1, \dots, y_l]^T$ that tracks a reference signal $r = [r_1, \dots, r_l]^T$. The reference is the the input to S_3 , or the output of S_2 .

We create four weight matrices for an LQ optimal control problem that correspond to the four motion factors discussed above. Specifically, each matrix is constrained to be the identity matrix times a scalar parameter where $Q = q\mathbb{I}_{n \times n}$ scales with *Direct Space*, $R = r\mathbb{I}_{m \times m}$ scales with *Strong Weight*, $P = p\mathbb{I}_{n \times n}$ scales with *Sustained Time*, and $S = s\mathbb{I}_{n \times n}$ scales with *Bound Flow*. Thus, $Q \sim \text{direct}$, $R \sim \text{light}$, $P \sim \text{sustained}$, $S \sim \text{bound}$. A quadratic cost function is established as:

$$J_u = \frac{1}{2} \int_0^T \{ (y-r)^T Q (y-r) + u^T R u + \dot{x}^T P \dot{x} \} dt + \frac{1}{2} (y-r)^T S (y-r) \Big|_T \quad (3)$$

to find input u based on the parameters q, r, p, s . We then formulate an optimal control problem:

$$\begin{aligned} \min_u J_u \\ \text{s.t.} \begin{cases} \dot{x} = Ax + Bu & x(0) = x_0 \\ y = Cx \end{cases} \end{aligned} \quad (4)$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, and $C \in \mathbb{R}^{l \times m}$. The optimal y is then based on the weight matrices in Equation 3, where large entries in the weight matrices could result in a more *direct*, *light*, *sustained*, and *bound* quality, and small entries could produce motion that is more *flexible*, *strong*, *sudden*, and *free*. Details on the solution of this optimal control problem and resulting Hamiltonian dynamics, which we use to compute the solution can be found in [11].

Applying optimal control formulation in S_3 . Using this optimal control framework, we take the input trajectory into S_3 as $r \in \mathbb{R}^3$ ($l = 3$ for the three joints). We construct a 6-dimensional system with double-integrator dynamics where $x = [\phi_1, \phi_1, \phi_2, \phi_2, \phi_3, \phi_3]^T$, $u = [u_{\phi_1}, u_{\phi_2}, u_{\phi_3}]^T$

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} u \quad (5)$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} x \quad (6)$$

For our application, we applied this methodology on sequential subsequences of the reference, r , which is set to the output of S_2 , σ , (i.e., for 1 second time intervals). We designate each of these subsequences as “windows.” The initial conditions for the first window are determined by the initial value of σ with zero velocities. The

remaining windows use the final conditions computed from the previous windows as their initial conditions.

The number of windows chosen for a motion sequence is a user-tuned parameter that affects the resulting motion. Using several windows instead of one large window (as used in [11]) is due to the high variability in the input joint trajectory which a single window between the starting point and ending point could not capture. Additionally, if the input joint trajectory has the same starting and end state (i.e. the arm could move in a circle), multiple windows could allow for capturing that variability.

The output of S_3 is a three-dimensional joint trajectory modified by four parameters corresponding to desired variations in the *Space*, *Time*, *Weight*, and *Flow* qualities of the motion. Currently, the framework does not recheck the feasibility of these new trajectories because due to the linear dynamics and quadratic cost structure used here, the variations produced by this subsystem are very small.

4 RESULTS

We have developed a framework using ROS, Gazebo, and Moveit that combines all three subsystems, allowing for the recording of multiple users simultaneously to control multiple robots by specifying the relationship between the demonstrators and robots. A user of this framework first modifies several options for the framework using a Python GUI, with several of the available options listed below:

- Number of users to track
- Number of robots to simulate
- Number of points to record of the demonstration
- For each robot to simulate:
 - Which user to imitate (e.g., 1st, 2nd, etc.)
 - Which part of the user to imitate (e.g., left arm)
 - Style parameters (q, r, p, s)
- Number of windows to include in style modifications

After the options are specified by the user, the framework records the human skeleton information and computes the 3-D human motion model for the specified number of users (with the skeleton appearing in Rviz as similar to Figure 3). Then the trajectory for the specified number of robots is calculated based on the user-defined parameters, and the robot motion is simulated using Gazebo and Moveit and is visualized as shown in Figure 3 (right).

We compared the results using our framework to generate robot motion imitating a single human motion demonstration in three different environments (Figure 4). These three environments, top row, have increasing constrictions of the possible robot motion when moving from left to right, with the plots corresponding to those environments appearing below each environment.

The raw angles (blue) are the angles obtained from S_1 (Section 3.1) from the human model. Using the methodology in Section 3.2 in S_2 , we create a feasible trajectory from the 3-D model to avoid collisions in each environment (orange). We then applied two different sets of style parameters on this feasible trajectory using the method in Section 3.3 from S_3 . Style 1 (gray) has the parameters $q = 0.1, r = 0.1, p = 1, s = 100$, and Style 2 (yellow) has the parameters $q = 1, r = 1, p = 10, s = 100$.

Figure 4: Using the same initial demonstration (blue), our system modifies the robot’s trajectory to avoid collisions in three different environments (orange). Our system also adds various styles to the motion via four different style parameters, and we have plotted two different styles in gray and yellow. The plots under each environment correspond to the three angles used in this framework at various steps. The dashed circles highlight key areas discussed in Sections 4.1 (red) and 4.2 (green)

To quantify the difference between two trajectories, we sum over the 2-norm of the difference at each point. Let the two trajectories to compare be $(\theta_1, \theta_2, \theta_3)$ and $(\theta'_1, \theta'_2, \theta'_3)$. We define the **cumulative distance** between θ and θ' as the sum over all time of $\sqrt{(\theta_1 - \theta'_1)^2 + (\theta_2 - \theta'_2)^2 + (\theta_3 - \theta'_3)^2}$.

4.1 Collision Avoidance in Different Environments

First, when examining the differences between the raw (blue) trajectory and the trajectory modified for collision avoidance (orange), key differences can be seen in the resulting angles. The dotted red circles indicate differences resulting from both the placement of

obstacles in the environment and the robot’s inherent dynamics and morphology. A discussion of these red circles is included below.

- Circles a and b : The difference between the raw angles from the human demonstration (blue) and the robot’s collision avoidant trajectory (orange) is due to the robot’s inherent dynamics and morphology.
- Circles (c_1, c_2) and (d_1, d_2) : Comparing the robot’s trajectory (orange) between the pairs of circles, the differences are due to the increasing environment clutter.

We also computed the **cumulative distance** between pairs of trajectories to add to our qualitative observations. We compared the differences between the raw angles from the human demonstration and the feasible trajectories generated for each environment (Table 1). As the environment becomes more complex (that is, moving

down the first column of Table 1 and from left to right in Figure 4) the difference between the human demonstration and the feasible robot trajectory becomes greater. The increased clutter makes it more difficult for the robot to imitate the human movement closely. This can also be seen by looking at the high **cumulative distance** between Environment 1’s feasible trajectory (6.40) and Environment 3’s feasible trajectory (18.75) in Table 1, indicating that the framework made more modifications to the demonstration with a more complex environment.

Table 1: The cumulative distance between trajectories, showing the effect of collision avoidance in different environments

	Raw	Feasible Env 1	Feasible Env 2
Feasible Env 1	6.38	-	-
Feasible Env 2	6.40	1.24	-
Feasible Env 3	18.75	15.94	15.94

4.2 Stylized Robot Motion in Different Environments

We also compared the resulting trajectories when applying two different sets of style parameters, Style 1 and Style 2. The dashed green circles indicate areas where the two styles deviated from each other slightly.

- Circles e, f, g : Differences between Style 1 (gray) and Style 2 (yellow) due to the varied style parameters.
- Circles (h_1, h_2) : The differences between the pair of circles are due to both the difference in the environment and the varying style parameters.

To quantify these observations further, we also computed **cumulative distance** values with the two stylized trajectories (Table 2). As the environments grow more complex (moving left to right in the table), the cumulative distance between the feasible trajectory and each styled trajectory decreases. This indicates that in more complex environments, the robot has less room for stylistic motions as the trajectory focuses on avoiding obstacles.

Table 2: The cumulative distance between trajectories, showing the effect of “adding” style in different environments

	Env 1	Env 2	Env 3
Feasible vs. Style 1	27.99	27.81	26.88
Feasible vs. Style 2	27.92	27.74	26.49
Style 1 vs. Style 2	1.58	1.25	2.11

In this paper, the feasibility of the stylized trajectories is not checked, which could generate trajectories with collisions after feasible trajectories are modified with the styled parameters. However, as seen in Table 2, the cumulative distance between the feasible and styled trajectories is around 28, which over 500 trajectory points is an average of a $0.056 \text{ rad} = 3.21^\circ$ deviation. In this demonstration, this was within our tolerance for error, but in future work, for more precise applications in complex environments, this framework can re-do the feasibility modification after the style modification, i.e., repeat S_2 after S_3 , to ensure the feasibility of the final trajectory.

5 CONCLUSIONS

In this paper, we developed a framework for generating feasible, stylized robot motion imitating a human demonstration. This was split into three subsystems: (1) creating a 3-DOF model of an aspect of human motion from skeleton data obtained using a depth camera, (2) modifying the trajectory to account for objects in the environment and the robot’s physical limitations, and (3) applying style parameters to allow for subtle changes in the quality of the robot motion; all while imitating the original demonstration closely.

Our results demonstrated the performance of our system in three simulated environments with varying amount of clutter, and the subtle changes in the motion produced by the style parameters. We illustrated the efficacy of the approach by quantitatively comparing the closeness of the imitation in both increasingly complex environments and with applied stylistic parameters. Our framework allows for the control of robots by users with less training. This capability could transform factory floors into spaces where humans and robots communicate through the kind of intuitive nonverbal bodily posture identified in [4]. However, this work does not aim to create anthropomorphic motion but to create distinct motion styles that humans can distinguish to facilitate non-verbal communication. Given the alien shape of some robotic platforms, humans may even need to learn new conventions through interaction in context to correctly interpret this non-verbal signaling.

Future work on this project can proceed in several different directions. Adding feasibility modification after the style modification will ensure that the final trajectory is feasible. Adding detail to the sensing system, e.g., by using hand and wrist motion capture, can increase the complexity of the resulting robot motion, i.e., specifying joint angles of the UR3 that were not utilized here. A real-time implementation of this approach can allow for humans to control a robot’s motion in real-time, for possible teleoperation applications. User studies are required to validate the perceptual response human users may have to the method and whether this form of imitation increases a sense of communication. Through pursuit of these extensions, this framework can be used to develop intuitive, adaptable motion of real robots on a factory floor to allow for easier programming of and collaboration with robots for industrial tasks.

ACKNOWLEDGMENTS

We would like to thank Siemens for providing the funding and resources for this project, which extended prior work conducted at the University of Illinois under DARPA award #D16AP00001. We would also like to thank Mayuri Deshpande (Siemens) for her feedback during this project.

REFERENCES

- [1] M Abdul-Massih, I Yoo, and B Benes. 2017. Motion Style Retargeting to Characters With Different Morphologies. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 86–99.
- [2] Ramesh Amit and Maja Matari. 2002. Learning movement sequences from demonstration. In *Proceedings 2nd International Conference on Development and Learning. ICDL 2002*. IEEE, 203–208.
- [3] Janet Beavin Bavelas, Alex Black, Nicole Chovil, Charles R Lemery, and Jennifer Mullett. 1988. Form and function in motor mimicry topographic evidence that the primary function is communicative. *Human Communication Research* 14, 3 (1988), 275–299.
- [4] Steven M Boker and Jennifer L Rotondo. 2002. Symmetry Building and Symmetry Breaking in Synchronized Movement. *Mirror neurons and the evolution of brain and language* 42 (2002), 163.
- [5] Peggy Hackney. 2003. *Making Connections: Total Body Integration through Bartineff Fundamentals*. Routledge.
- [6] Bernhard Jung, Heni Ben Amor, Guido Heumer, and Matthias Weber. 2006. From motion capture to action capture: A review of imitation learning techniques and their application to VR-based character animation. In *Proceedings of the ACM symposium on Virtual reality software and technology*. 145–154.
- [7] Roshni Kaushik and Amy LaViers. 2018. Imitating Human Movement Using a Measure of Verticality to Animate Low Degree-of-Freedom Non-Humanoid Virtual Characters. In *Social Robotics*, Vol. 11357. Springer International Publishing, Cham, 588–598. https://doi.org/10.1007/978-3-030-05204-1_58
- [8] Roshni Kaushik, Ilya Vidrin, and Amy LaViers. 2018. Quantifying Coordination in Human Dyads via a Measure of Verticality. In *Proceedings of the 5th International Conference on Movement and Computing - MOCO '18*. ACM Press, Genoa, Italy, 1–8. <https://doi.org/10.1145/3212721.3212805>
- [9] Heather Knight. 2016. Expressive Motion for Low Degree-of-Freedom Robots. (2016).
- [10] Rudolf Laban and Frederick Charles Lawrence. 1947. *Effort*. Macdonald & Evans.
- [11] Amy LaViers and Magnus Egerstedt. 2012. Style Based Robotic Motion. In *2012 American Control Conference (ACC)*. IEEE, 4327–4332.
- [12] John McCormick, Kim Vines, Saeid Nahavandi, Douglas Creighton, and Steph Hutchison. 2014. Teaching a digital performing agent: Artificial neural network and hidden markov model for recognising and performing dance movement. In *Proceedings of the 2014 International Workshop on Movement and Computing*. 70–75.
- [13] Jean Newlove and John Dalby. 2004. *Laban for All*. Taylor & Francis US.
- [14] Christian Ott, Dongheui Lee, and Yoshihiko Nakamura. 2008. Motion Capture Based Human Motion Recognition and Imitation by Direct Marker Control. In *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference On*. IEEE, 399–405.
- [15] Miriam Phillips. 2013. Becoming the floor/breaking the floor: Experiencing the kathak-flamenco connection. *Ethnomusicology* 57, 3 (2013), 396–427.
- [16] Bonggun Shin and Sungho Jo. 2011. Pattern-Preserving-Based Motion Imitation for Robots. In *2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. IEEE, 20–25.
- [17] Reid Simmons and Heather Knight. 2017. Keep on Dancing: Effects of Expressive Motion Mimicry. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 720–727.
- [18] K. Studd and L. Cox. 2013. *Everybody is a Body*. Dog Ear Publishing.
- [19] Wael Suleiman, Eiichi Yoshida, Fumio Kanehiro, Jean-Paul Laumond, and André Monin. 2008. On Human Motion Imitation by Humanoid Robot. In *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2697–2704.
- [20] Katsu Yamane, Yuka Ariki, and Jessica Hodgins. 2010. Animating non-humanoid characters with human motion data. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 169–178.