sensors

MDPI

Article

Robot Imitation Learning of Social Gestures with Self-Collision Avoidance Using a 3D Sensor

Tan Zhang [1,2,*], Wing-Yue Louie [1,3], Goldie Nejat [1] and Beno Benhabib [4]

[1] Autonomous Systems and Biomechatronics Lab, Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, ON M5S3G8, Canada; geoffrey.louie@utoronto.ca (W.-Y.L.); nejat@mie.utoronto.ca (G.N.)
[2] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China
[3] Intelligent Robotics Lab, Department of Electrical and Computer Engineering, Oakland University, Rochester, MI 48309, USA
[4] Computer Integrated Manufacturing Lab, Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, ON M5S3G8, Canada; benhabib@mie.utoronto.ca
* Correspondence: tan.zhang@utoronto.ca; Tel.: +1-3063411528

Received: 25 June 2018; Accepted: 18 July 2018; Published: 20 July 2018

check for updates

Abstract: To effectively interact with people, social robots need to perceive human behaviors and in turn display their own behaviors using social communication modes such as gestures. The modeling of gestures can be difficult due to the high dimensionality of the robot configuration space. Imitation learning can be used to teach a robot to implement multi-jointed arm gestures by directly observing a human teacher's arm movements (for example, using a non-contact 3D sensor) and then mapping these movements onto the robot arms. In this paper, we present a novel imitation learning system with robot self-collision awareness and avoidance. The proposed method uses a kinematical approach with bounding volumes to detect and avoid collisions with the robot itself while performing gesticulations. We conducted experiments with a dual arm social robot and a 3D sensor to determine the effectiveness of our imitation system in being able to mimic gestures while avoiding self-collisions.

Keywords: imitation learning; social robots; multi-jointed arm gestures; self-collision avoidance

1. Introduction

Social robots are currently being developed to interact with humans by perceiving human behavior [1] and displaying natural human-like gestures such as waving to greet humans or pointing to objects to guide their visual focus of attention [2–4]. Trajectories to generate these human-like gestures are often programmed and implemented by expert roboticists. Imitation learning provides a natural mode of human-robot interaction for non-experts to teach a robot arm gestures by having a robot observe, using visual sensors, the teacher's physical demonstrations of the gestures and then map these observed motions onto the robot's own embodiment [5–7].

1.1. Related Work

1.1.1. Imitation Learning

A common approach for imitation learning is to sense the 3D arm joint positions of a demonstrator and solve for the inverse kinematics problem to determine the joint angles between each of the demonstrator's arm segments [8]. The demonstrator joint angles can then be directly mapped to the robot's arms for robots with the same degrees of freedom (DOF) of a human (e.g., the Nao robot). The benefit of utilizing an inverse kinematics model for trajectory representation is that all arm segments are represented at once. These models can be solved using an analytical approach [9], iterative Jacobian [10], or neuro-fuzzy system methods [10]. It was found in [10] that analytical approaches and iterative Jacobian methods require more complex models that cannot be solved in real-time, while neuro-fuzzy system methods require large amounts of training data to generate accurate models. The geometrical approach in [11] utilizes triangles to model the joint angles for human arm motions, which can be used for directly mapping human arm motions to a robot. The geometric-based imitation learning system presented in this paper is inspired by the triangle-based approach presented in [11].

1.1.2. Imitation Learning with Self-Collision Avoidance

Imitation learning alone is not sufficient to mimic human arm gestures on a robot as differences in the embodiment can lead to self-collisions (which can cause damage to the robot) and the inability to implement certain movements. Hence, self-collision avoidance techniques are necessary to generate collision-free trajectories that are close approximations of a demonstrator's arm gestures. Presently, there is some work that detects if self-collisions will occur given a motion trajectory [12,13]; however, they are not capable of generating alternative trajectories to avoid these self-collisions. These self-collision approaches model a robot's body, arms and/or head using bounding volumes, such as spheres and circular capsules (i.e., cylinders capped with spheres) and calculate the distance between the bounding volumes. Collisions are defined as scenarios where the distance between two bounding volumes is less than the sum of their two radii. The work in [12] proposed a collision detection method for a vine pruning robot by modeling plant canes and the robot arm as capsules to check for collisions between the robot arm and the vine. This approach only considers the collisions between two capped ends and the coincidence between two capsules, without taking into account the collisions between two cylinders, whose spatial relations are parallel, perpendicular, or skew. Similarly, in [13] the robot's arms are modeled as circular capsules while elliptical capsules (i.e., elliptical cylinders capped by ellipsoids) were used to fit the humanoid torso shape. Namely, spatial relations between capsules were determined by modeling each capsule as a straight line of infinite length and verifying if the lines intersect. However, such an approach can lead to incorrect collision detections as the capsules have finite length.

More recently, some preliminary work has been done to integrate self-collision avoidance techniques into robot imitation learning systems [14–18]. In [14], continuous null-space projections were used to avoid the potential collisions between two Kuka LWR robotic arms being controlled through an imitation learning system using a Kinect sensor. Namely, self-collision avoidance was accomplished by constraining arm joint movements within the null-space projection of the collision-free arm configuration. However, the approach was limited to self-collision avoidance between only two arms and did not account for self-collision avoidance with multiple body parts (e.g., head, arms, and torso). In [15], an imitation learning system with self-collision avoidance utilized six optimal cameras and 38 markers attached to a demonstrator to sense his/her joint positions during a movement. An inverse kinematics-based approach was used to determine demonstrator joint angles from the sensed joint positions and a collision-free motion trajectory was mapped to the robot by solving for a trajectory that meets the robot's joint angle and self-collision constraints. Namely, self-collisions were detected by determining the distance between any two 3D points on the robot. However, such a self-collision detection approach is computationally expensive and would not work in real-time as it requires a pairwise distance calculation between all 3D points on a robot. In [16], a Kinect sensor-based imitation learning system with self-collision avoidance was developed for a robot with dual arms mounted on a torso. The sensed 3D joint positions of the demonstrator were mapped directly onto a robot by inputting the positions into a cartesian impedance controller. Self-collision avoidance for this imitation learning system was accomplished by modeling the robot's arms as two-line segments and the torso as a rectangular plane. Namely, the minimum distance between the two-line segments and all the distances from the line segments to the plane were continuously monitored to detect collisions between the robot's body parts (i.e., arms and torso). Detected collisions were then used to modify the robot motion trajectories by moving the body parts in the opposite direction of the contact at a speed proportional to the distance of the body parts. However, such a technique has the potential to reduce the similarity of the robot poses to the demonstrator and can result in jerky motions due to rapid changes in speed when avoiding collisions. In [17], a Kinect sensor-based imitation learning system with self-collision avoidance was developed for the humanoid robot Nao. Namely, an inverse-kinematics-based approach was applied to determine demonstrator joint angles from sensed joint positions and directly map them to the Nao robot's joints. A self-collision avoidance technique was also presented that models the robot links as cylinders with spherical ends. The criterion for a collision was defined as situations where the shortest distance between two-line segments was less than the sum of the cylinder radii. When a collision was detected, the robot just stopped moving. However, for such a collision avoidance technique, the demonstrator's poses need to be sampled at a fast-enough rate in order to avoid the motion between two collision free poses introducing a collision. In [18], a similar Kinect sensor-based imitation learning system with self-collision was developed for the humanoid robot Nao. Namely, the robot's workspace was first empirically modeled off-line by manually sampling robot joint poses to identify if a collision would occur. The workspace model was then utilized online to solve for collision-free joint poses to imitate a demonstrator's poses by determining if a solution was within the collision-free workspace of the robot. However, the accuracy was limited to the number of samples used to model the workspace and could not account for all possible collision/collision-free poses.

To date, imitation learning systems with self-collision avoidance are limited as they are: (1) capable of modelling only collisions between two body parts [14]; (2) computationally expensive and therefore cannot be used in real-time [15]; (3) unable to account for all collisions/collision-free poses during a demonstration [17,18]; and (4) unsuitable for gesture demonstrations due to rapid joint speed changes as a result of the collision avoidance technique [16]. Therefore, in this paper, we propose a real-time gesture imitation learning system that can address these limitations by detecting all self-collisions for a robot with multiple body parts. The novelty of this approach is that it optimizes a collision-free motion trajectory while minimizing the distance from the demonstrator's trajectory. The system utilizes a geometric-based approach to map human joint angles to robot joint angles and a kinematical-based approach for self-collision avoidance.

2. Gesture Imitation Learning System

The proposed gesture imitation learning system is presented in Figure 1. The gesture imitation leaning system has two main modules: (1) a skeleton motion mapping module; and (2) a self-collision detection and avoidance module. Namely, the skeleton motion mapping module takes as input a sequence of demonstrator poses (i.e., joint positions in 3D space), herein referred to as a demonstration trajectory, and solves for the sequence of demonstrator joint angles executed to attain this sequence of poses. The self-collision detection and avoidance module then detects for potential self-collisions when the sequence of joint angles is directly mapped to the robot. If a collision is detected for a robot joint angle configuration in the sequence it solves for a collision-free joint angle configuration that closely imitates the human demonstrator. Finally, the sequence of collision-free joint angle configurations is outputted to be executed by the robot.

2.1. Skeleton Motion Mapping Module

The skeleton motion mapping module first obtains the demonstration trajectory and applies a geometric-based approach to each of the demonstrator's poses, $\boldsymbol{x}_{human}$, to obtain a mapping to the robot's pose, $\boldsymbol{q}_{mimic}$. As shown in Figure 2, the demonstrator's arm is modeled with four degrees of freedoms (DOFs). There are three DOFs in the shoulder and one DOF in the elbow. We first determine the shoulder roll (θ_1), pitch (θ_2), and yaw (θ_3) angles using the following equations:

$$\theta_1 = arcsin(\frac{\overrightarrow{l1y}}{\sqrt{\overrightarrow{l1x}^2+\overrightarrow{l1y}^2+\overrightarrow{l1z}^2}}) \tag{1}$$

$$\theta_2 = arcsin(\frac{\underset{d1x}{\rightarrow}}{\sqrt{\underset{l1x}{\rightarrow}^2 + \underset{l1y}{\rightarrow}^2 + \underset{l1z}{\rightarrow}^2}}) \tag{2}$$

$$\theta_3 = arcsin(\frac{\underset{d2x}{\rightarrow}}{\sqrt{\underset{l2x}{\rightarrow}^2 + \underset{l2y}{\rightarrow}^2 + \underset{l2z}{\rightarrow}^2}}) \tag{3}$$

where $\vec{l}_{1x}$, $\vec{l}_{1y}$, and $\vec{l}_{1z}$ are the projections of l_1 on the axes of x, y and z, respectively. Similarly, $\vec{l}_{2x}$, $\vec{l}_{2y}$, and $\vec{l}_{2z}$ are the projections of l_2 on the axes of x, y and z, respectively. $\vec{l}_1$, $\vec{l}_2$, and $\vec{l}_3$ are obtained as follows:

$$\vec{l}_1 = \vec{P}_e - \vec{P}_s \tag{4}$$

$$\vec{l}_2 = \vec{P}_w - \vec{P}_e \tag{5}$$

$$\vec{l}_3 = \vec{P}_w - \vec{P}_e \tag{6}$$

where $\vec{P}_w$, $\vec{P}_e$, and $\vec{P}_s$ are the positions of the wrist, elbow and shoulder, and $\boldsymbol{x}_{human}$ = $\{\vec{P}_w, \vec{P}_e, \vec{P}_s\}$.

We then determine the elbow roll angle, θ_4, with the following equation:

$$\theta_4 = \arccos\left(\frac{|l_3|^2 - |l_1|^2 - |l_2|^2}{2|l_1||l_2|}\right) \tag{7}$$

where $|\vec{l}_1|$ and $|\vec{l}_2|$ are the upper arm and forearm of the left/right arm, respectively. $|\vec{l}_3|$ is the length from the shoulder to the elbow, as shown in Figure 2.

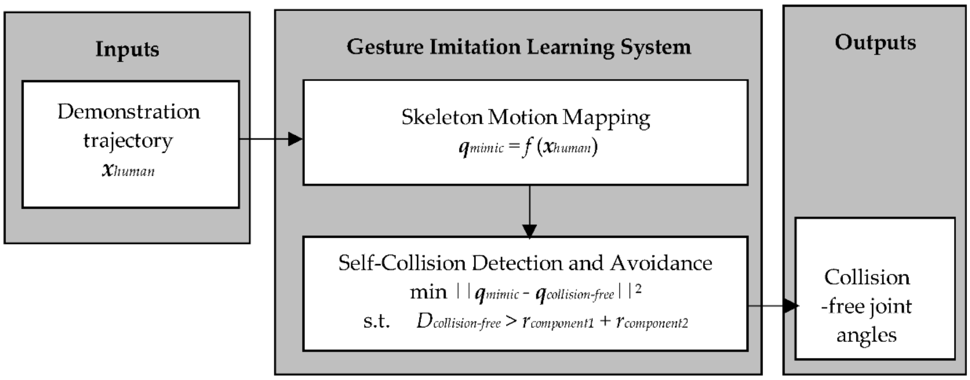

Inputs
Demonstration trajectory
x_{human}
Gesture Imitation Learning System
Skeleton Motion Mapping
$q_{mimic} = f\,(x_{human})$
Self-Collision Detection and Avoidance
min $||q_{mimic} - q_{collision\text{-}free}||^2$
s.t. $D_{collision\text{-}free} > r_{component1} + r_{component2}$
Outputs
Collision -free joint angles

Figure 1. Diagram of the gesture imitation learning system.

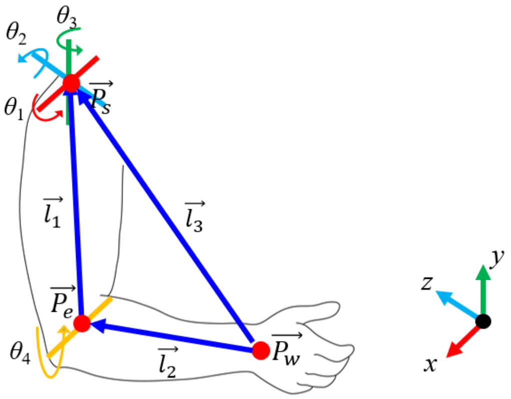


Figure 2. The human arm model.

The shoulder and elbow angles of the demonstrator are then mapped directly to robot's pose, $\boldsymbol{q}_{mimic}$, since we are focusing on imitation learning for robots with a humanoid upper torso. Namely, the robot's pose can be defined as the set $\boldsymbol{q}_{mimic} = \{\theta_1, \theta_2, \cdots, \theta_u\}$, and u is the number of the robot joints. The robot's pose is then input to the self-collision detection and avoidance module to avoid self-collisions due to differences in embodiment between the robot and human demonstrator.

2.2. Self-Collision Detection and Avoidance Module

The self-collision detection and avoidance module is represented by two submodules: (1) the self-collision detection submodule that utilizes a bounding-volume-based approach to detect collisions between any two body parts for a robot pose; and (2) the collision avoidance submodule that solves for the closest collision-free joint angles that mimic the demonstrator's poses.

To avoid the possible collision points, the module first checks the distance between any two components. Using a bounding-volume-based approach, the components are modeled as spheres or capsules with a radius r. Our approach obtains the position of each joint based on $\boldsymbol{q}_{mimic}$ by forward kinematics, and the distance D between two components whose radii are r_1 and r_2. If $D > r_1 + r_2$, there is no collision between the two components. If there is a collision between two components, the module finds a set of new angles $\boldsymbol{q}_{collision\text{-}free}$ that has a minimum difference with $\boldsymbol{q}_{mimic}$, to ensure that there is a minimum distance D between the two volumes.

$$\begin{aligned} \text{minimize} \qquad & ||\boldsymbol{q}_{mimic} - \boldsymbol{q}_{collision\text{-}free}||^2 \\ \text{subject to} \qquad & D > r_1 + r_2 \end{aligned} \tag{8}$$

Finally, $\boldsymbol{q}_{collision\text{-}free}$ is sent to the robot to implement.

2.2.1. Self-Collision Detection

The self-collision detection submodule is used to determine whether two body parts of the robot will be in collision given a robot pose $\boldsymbol{q}_{mimic}$. Herein, we model a robot's body parts as either capsules (e.g., arms, torso) or spheres (e.g., head). A sphere is defined using its centroid and radius. A capsule is a cylinder with two half spheres at its two ends, which is defined by the positions of its two end points and the radius of the cylinder. We define a robot pose to have no self-collisions when the distance between two body parts is larger than the summation of the radii of the two body part models:

$$D_{kj} - (r_k + r_j) > 0, k = 1, \ldots, n; j = 1, \ldots, n \tag{9}$$

where D_{kj} is the shortest distance between two modeled robot body parts, body parts k and j. r_k and r_j are the radii of robot body parts k and j, respectively. n is the total number of robot body parts.

Collision Checks between a Sphere and a Capsule

The collision checks of a sphere and a capsule are represented by the distance between a point and a line segment in 3D space. Let P_s, M_t, and M_b represent the center of sphere S, and the two end points of capsule M, respectively, as shown in Figure 3. $\overrightarrow{M_t M_b}$ is the axis of capsule M. H is the foot of the common normal of point P_s on $\overrightarrow{M_t M_b}$.

The following different cases are considered for calculating the distance between a sphere and a capsule:

(1) If $\overrightarrow{P_s M_t} \cdot \overrightarrow{M_t M_b} \leq 0$, the intersection of the perpendicular line lies on the extended line of $\overrightarrow{M_t M_b}$ near Point M_t, and the shortest distance is $|\overrightarrow{P_s M_t}|$.

(2) If $\overrightarrow{P_s M_t} \cdot \overrightarrow{M_t M_b} \geq |\overrightarrow{M_t M_b}|$, the intersection of the perpendicular line lies on the extended line of $\overrightarrow{M_t M_b}$ near Point M_b, and the shortest distance is $|\overrightarrow{P_s M_b}|$.

(3) If $0 \leq \overrightarrow{P_sM_t} \cdot \overrightarrow{M_tM_b} \leq |\overrightarrow{M_tM_b}|$, the intersection of the perpendicular line lies on $\overrightarrow{M_tM_b}$, and the shortest distance is $|\overrightarrow{P_sH}|$.

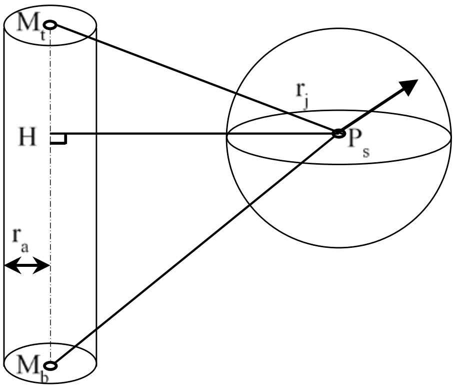


Figure 3. Collison check between a sphere and a capsule.

Collision Checks between Two Capsules

The collision checks of two capsules become the distance between two line-segments in 3D space. Let M_t, M_b, N_t and N_b represent the two end points of capsule M and capsule N, respectively. $\overrightarrow{M_tM_b}$ is the axis of M and $\overrightarrow{N_tN_b}$ is the axis of N.

The following are several cases for calculating the distance between two capsules:

(1) If $\overrightarrow{M_tM_b} \cdot \overrightarrow{M_tN_t} \cdot \overrightarrow{M_tN_b} \neq 0$, the two lines $\overrightarrow{M_tM_b}$ and $\overrightarrow{N_tN_b}$ are skew lines, as shown in Figure 4. There are two situations as follows:

(i) If both the intersections M_h and N_h lie on the two segments, i.e., $|\overrightarrow{M_tM_h}| + |\overrightarrow{M_hM_b}| = |\overrightarrow{M_tM_b}|$ and $|\overrightarrow{N_tN_h}| + |\overrightarrow{N_hN_b}| = |\overrightarrow{N_tN_b}|$, then the shortest distance is the perpendicular distance $|\overrightarrow{M_hN_h}|$.

(ii) If one of the two intersections, M_h or N_h, do not lie on the segment, i.e., $|\overrightarrow{M_tM_h}| + |\overrightarrow{M_hM_b}| \neq |\overrightarrow{M_tM_b}|$ or $|\overrightarrow{N_tN_h}| + |\overrightarrow{N_hN_b}| \neq |\overrightarrow{N_tN_b}|$, the shortest distance is the smallest among $|\overrightarrow{M_hN_h}|$, and d_1, d_2, d_3 and d_4, as shown in Figure 4. d_1, d_2, d_3 and d_4 are the perpendicular distance from point M_t to $\overrightarrow{N_tN_b}$, the perpendicular distance from point M_b to $\overrightarrow{N_tN_b}$, the perpendicular distance from point N_t to $\overrightarrow{M_tM_b}$, and the perpendicular distance from point N_b to $\overrightarrow{M_tM_b}$, respectively.

(2) If the two lines $\overrightarrow{M_tM_b}$ and $\overrightarrow{N_tN_b}$ are proportional, i.e., $\overrightarrow{M_tM_b} = k \cdot \overrightarrow{N_tN_b}$, where $k \neq 0$, the two lines are parallel, as shown in Figure 5. The shortest distance will be the smaller of d_1 and d_2.

(3) If the two lines $\overrightarrow{M_tM_b}$ and $\overrightarrow{N_tN_b}$ are neither skew nor parallel, the two lines are intersected at P, as shown in Figure 6. There are two situations as follows:

(i) If P lies on the two segments, i.e., $|\overrightarrow{PM_t}| + |\overrightarrow{M_tM_b}| = |\overrightarrow{PM_b}|$ and $|\overrightarrow{PN_t}| + |\overrightarrow{N_tN_b}| = |\overrightarrow{PN_b}|$, the shortest distance is zero. Let the two lines $\overrightarrow{M_tM_b}$ and $\overrightarrow{N_tN_b}$ be represented as vectors $\vec{m}(t) = \vec{M_t} + \overrightarrow{M_tM_b} \cdot t$ and $\vec{n}(t) = \vec{N_t} + \overrightarrow{N_tN_b} \cdot s$, the intersection of the two segments, i.e., the position of P is the solution of $\vec{m}(t) = \vec{n}(t)$.

(ii) If P does not lie on one of the two segments, i.e., $|\overrightarrow{PM_t}| + |\overrightarrow{M_tM_b}| \neq |\overrightarrow{PM_b}|$ or $|\overrightarrow{PN_t}| + |\overrightarrow{N_tN_b}| \neq |\overrightarrow{PN_b}|$, the shortest distance is the smallest among d_1, d_2, d_3 and d_4.

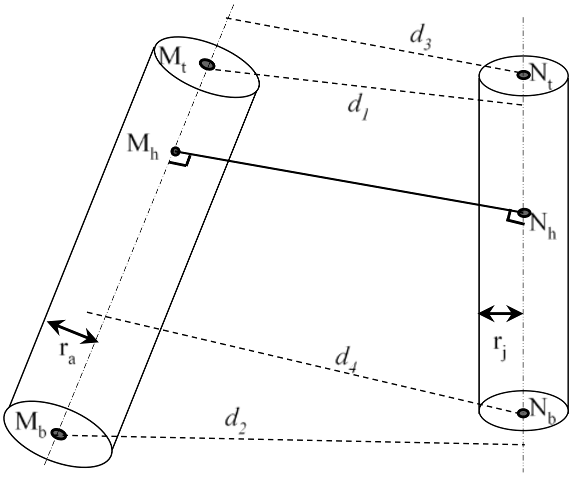


Figure 4. Collison checks between two capsules (two axes are skew).

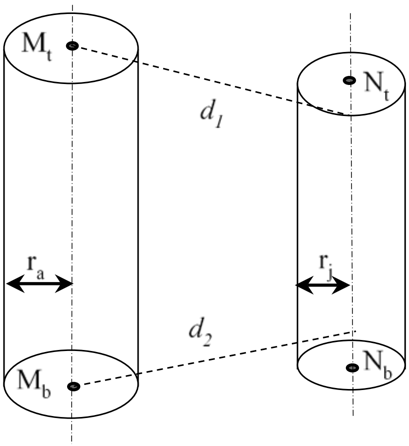


Figure 5. Collison checks between two capsules (two axes are parallel).

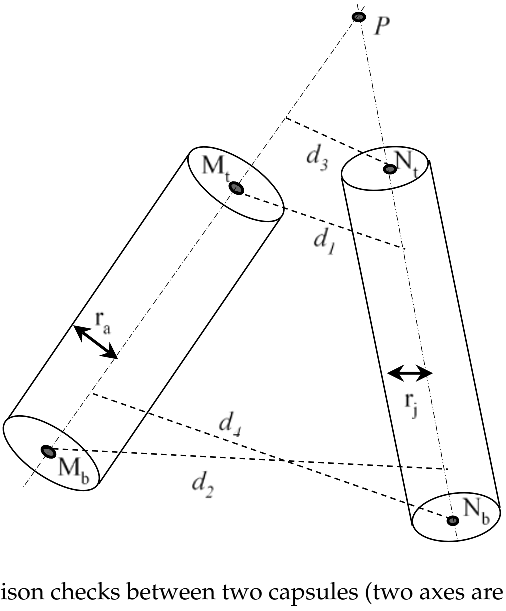


Figure 6. Collison checks between two capsules (two axes are intersecting).

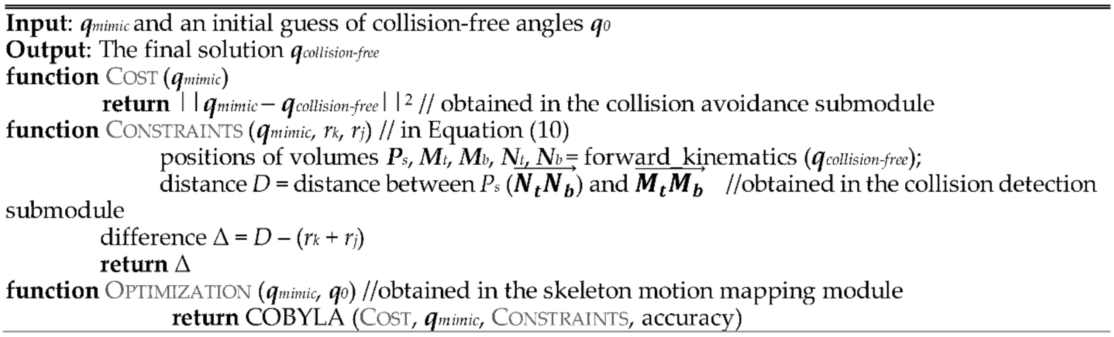
2.2.2. Self-Collision Avoidance

Given a self-collision has been detected, the self-collision avoidance submodule determines robot joint angles that best mimic a human demonstrator's pose. Namely, we define self-collision avoidance as an optimization problem which solves for the set of robot joint angles which are collision-free, $q_{collision\text{-}free}$, and has the minimum joint angle difference with a demonstrated pose q_{mimic}. The objective function, e, for the optimization is defined as:

$$\begin{array}{ll} \text{minimize} & e = \sum_{i=1}^{u} \left(\theta_i - \theta_i'\right)^2 \\ \text{subject to} & D_{kj} - \left(r_k + r_j\right) > 0,\ k = 1, \ldots, n;\ j = 1, \ldots, n \end{array} \tag{10}$$

where i is the robot joint angle being optimized, u is the total number of robot joints, θ_i and θ_i' are, respectively, the demonstrated joint angle and collision-free joint angle of joint i. The constraint for the optimization is that all the robot body parts (i.e., head, arms, upper torso, and lower torso) must be collision free. D_{kj} is the shortest distance between two modeled robot body parts, body parts k and j. r_k and r_j are the radii of robot body parts k and j, respectively. n is the total number of robot body parts.

In this work, we used a constrained optimization by linear approximation (COBYLA) algorithm [19] to solve the optimization problem. COBYLA is a derivation-free optimization method for constrained problems [20]. The specific algorithms we have designed for this optimization are presented in Figure 7.



```
Input: $q_{mimic}$ and an initial guess of collision-free angles $q_0$
Output: The final solution $q_{collision\text{-}free}$
function COST ($q_{mimic}$)
        return $||q_{mimic} - q_{collision\text{-}free}||^2$ // obtained in the collision avoidance submodule
function CONSTRAINTS ($q_{mimic}$, $r_k$, $r_j$) // in Equation (10)
                positions of volumes $P_s$, $M_t$, $M_b$, $N_t$, $N_b$ = forward_kinematics ($q_{collision\text{-}free}$);
                distance $D$ = distance between $P_s$ ($\overrightarrow{N_t N_b}$) and $\overrightarrow{M_t M_b}$   //obtained in the collision detection
submodule
        difference $\Delta = D - (r_k + r_j)$
        return $\Delta$
function OPTIMIZATION ($q_{mimic}$, $q_0$) //obtained in the skeleton motion mapping module
                return COBYLA (COST, $q_{mimic}$, CONSTRAINTS, accuracy)
```

Figure 7. Pseudo-code for searching collision-free angles.

3. Imitation Learning with the Tangy Robot

We implemented our gesture imitation learning system on the humanoid robot, Tangy, shown in Figure 8. The robot consists of two six-DOF arms, an upper torso with a tablet screen, a lower torso connected to a mobile base. The configuration and parameters of the robot are detailed in Appendix A. Following the bounding-volume-based approach presented in Section 2.2.1, the robot's upper torso is modeled as five capsules (see Figure 8a), with one capsule located at the bottom of the screen and two spheres located at the two bottom corners of the screen to incorporate the USB cables, the circles shown in Figure 8b. The lower torso, the upper arms, and the forearms are also modeled as capsules, and the head is modeled as a sphere, as shown in Figure 8a. We used the ASUS Xtion 3D sensor, Figure 9, to track the position of the demonstrator's joints using the OpenNI skeleton tracker [21]. In our real-world implementation of the gesture imitation learning system, we include angular velocity joint constraints to filter noisy sensor data which does not correspond to actual human demonstrator joint movements. Namely, an empirically identified threshold of 1 rad/s was used with our gesture imitation learning system. We have also included speed constraints to prevent the robot manipulator from exceeding 250 mm/s to comply with the current ISO 15066 for collaborative robots to ensure safety when a robot is working in close proximity to a human [22].

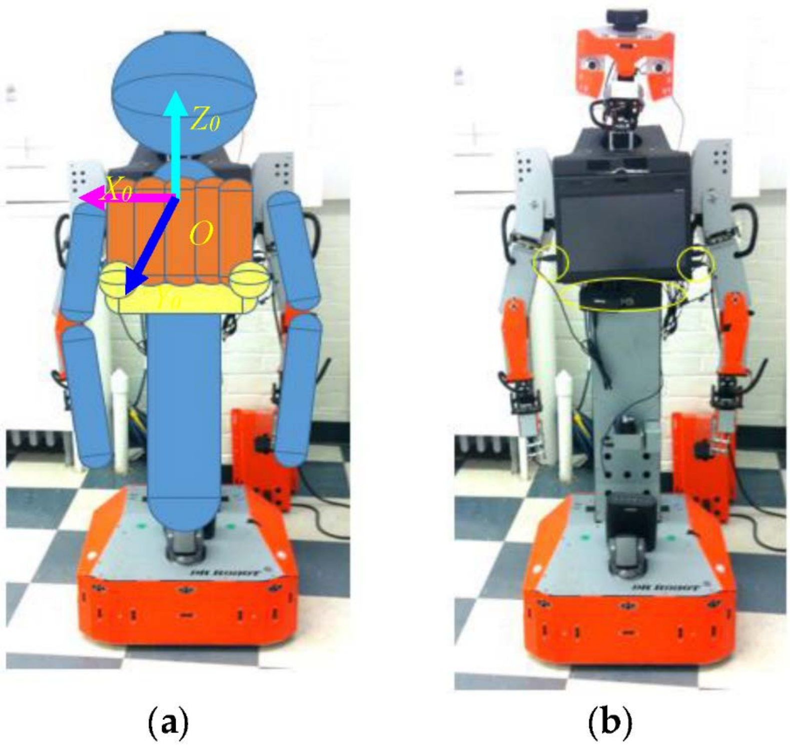


Figure 8. (**a**) The Tangy robot; (**b**) modeled with three spheres and eleven capsules.

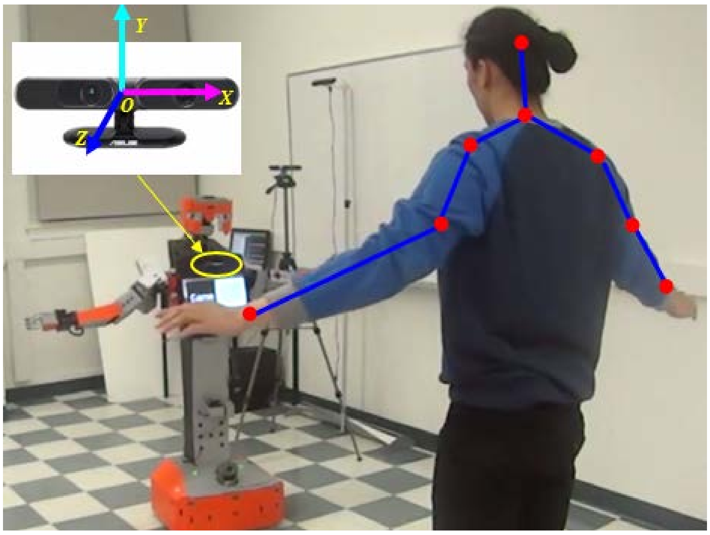


Figure 9. The Tangy robot using the ASUS Xtion 3D sensor to learn human demonstrator motions.

4. Experimental Section

We evaluate the performance of our proposed gesture imitation learning system both in simulation and in the real-world on the Tangy robot. Namely, the experiments focused on evaluating the imitation learning system to investigate the performance of the system in: (1) avoiding self-collisions during a human demonstration; and (2) generating collision-free robot arm trajectories that are similar to the demonstrator's motions.

Human demonstrators presented five separate motions to the robot, motions I to V, Figure 10. Motion I has the demonstrator extend his/her arms forward horizontally from the vertical position (from pose I_1 to pose I_2) and then back to the vertical position (from pose I_2 to pose II_1). motion II has the demonstrator extend his/her arms horizontally sideways from the vertical position (from pose II_1 to pose II_2) and then back to the vertical position (from pose II_2 to pose III_1). Motion III has the demonstrator extend his/her arms forward horizontally from the vertical position (from pose III_1 to pose III_2), then sideways (from pose III_2 to pose III_3), and finally back to the vertical position (from pose III_3 to pose IV_1). Motion IV has the demonstrator fold his/her elbow (from pose IV_1 to pose IV_2) and then back to the vertical position (from pose IV_2 to pose V_1). Motion V has the demonstrator fold his/her arms towards the chest (from pose V_1 to pose V_2) and then back to the vertical position (from pose V_2 to pose V_3). These motions were selected as they evaluate the entire range of motion for the robot's shoulder and elbow joints. Similar motions were also evaluated in [18]. Demonstrators were asked to repeat each motion twice.

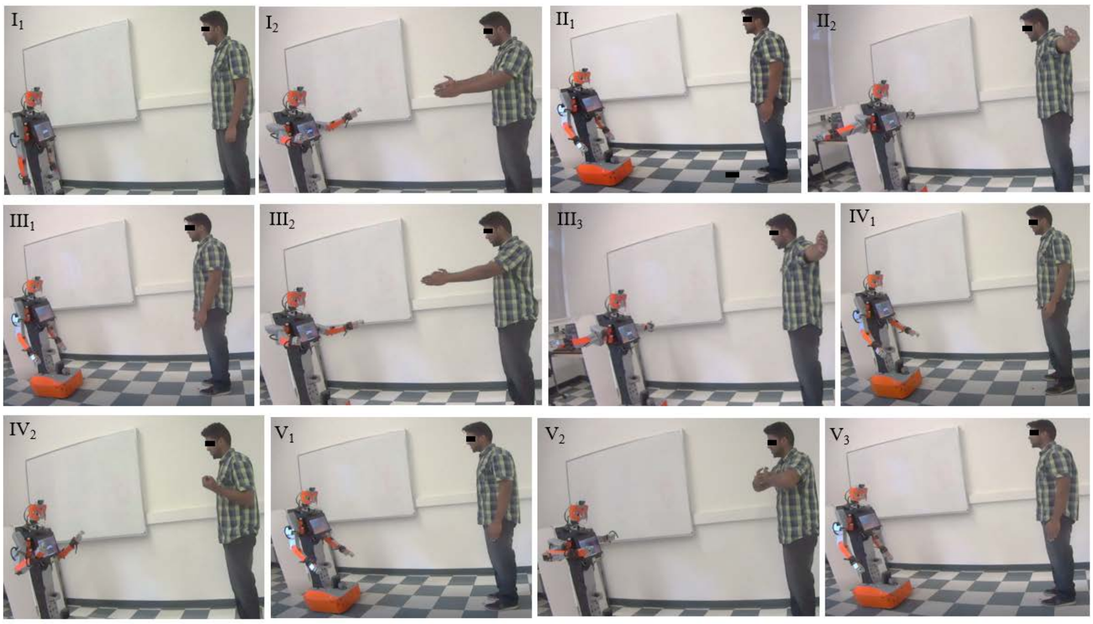
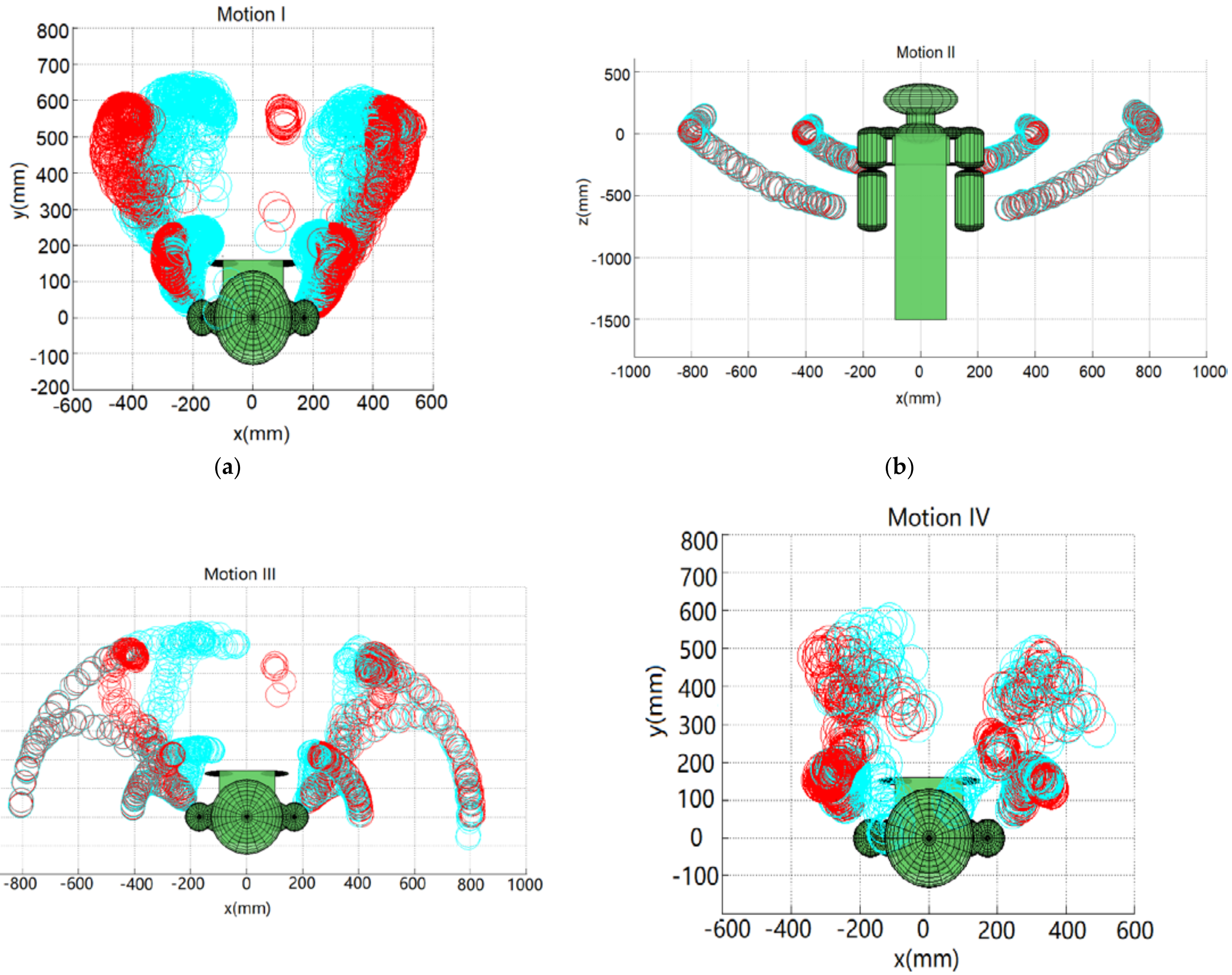


Figure 10. Video frame sequences when imitating human demonstrator motion using the imitation learning system. Five sets of motions include motion I (**$\mathbf{I_1}$–$\mathbf{II_1}$**), motion II (**$\mathbf{II_1}$–$\mathbf{III_1}$**), motion III (**$\mathbf{III_1}$–$\mathbf{IV_1}$**), motion IV (**$\mathbf{IV_1}$–$\mathbf{V_1}$**), and motion V (**$\mathbf{V_1}$–$\mathbf{V_3}$**).

4.1. Simulation Results

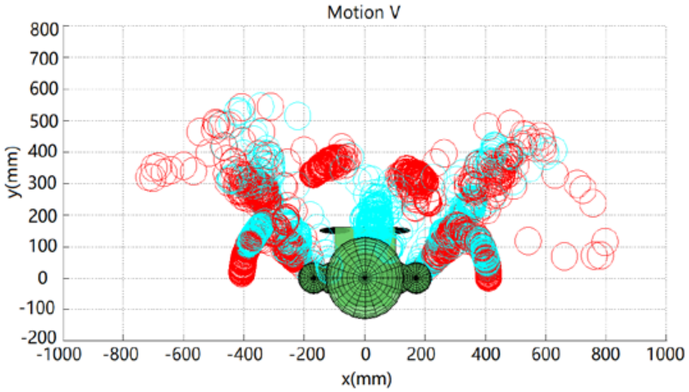
The demonstrator's five set of motions are captured using the ASUS Xtion 3D sensor. Following the geometric-based approach presented in Section 2.1, the demonstrator's poses are mapped to the robot. Figure 11 shows the position of the robot's end-effector and the elbow for the aforementioned five set of motions. The blue circles represent the workspace of the robot's end-effector and elbow when the joint angles of the human demonstrator are directly mapped to the robot, $\boldsymbol{q}_{mimic}$, without any self-collision avoidance. The red circles represent the robot's collision-free poses determined with self-collision avoidance, $\boldsymbol{q}_{collision\text{-}free}$. It can be observed from Figure 11 that some of the blue circles collide with the robot, while no red circles collide with the robot. It is noted that there is collision if the upper arm (the segment of shoulder and elbow) or the forearm (the segment of elbow and wrist) contacts any of the robot body parts. For example, without self-collision avoidance, the robot arms collide with the tablet screen in motions I, III, IV and V, and the two wrists collide with each other in motion V.



(**a**) (**b**)

(**c**) (**d**)

Figure 11. *Cont*.



(**e**)

Figure 11. The positions of the robot's arms for Motions I, II, III, IV and V shown in (**a**–**e**), respectively. Blue circles and red circles represent the positions determined by the directly mapped and collision-free trajectories, respectively.

Figure 12a,b show the $\boldsymbol{q}_{mimic}$ and $\boldsymbol{q}_{collision\text{-}free}$ trajectories for each motion, respectively. It is observed that motion I (extending arms forward) is mainly actuated by the shoulder roll angle θ_1; motions II (extending arms horizontally) and III (extending arms forward and then horizontally) are actuated by all the shoulder joints, θ_1, θ_2, and θ_3. Different from motions I, II, and III, motion IV (folding elbows) is mainly actuated by the elbow joint θ_4. Motion V (a random movement) is actuated by all the joints. Figure 12c shows which joints are actuated to avoid the potential collisions. Δ is used to represent the difference between $\boldsymbol{q}_{mimic}$ and $\boldsymbol{q}_{collision\text{-}free}$, i.e., $\Delta = |\boldsymbol{q}_{mimic} - \boldsymbol{q}_{collision\text{-}free}|$. For example, for motion I and motion III, the robot arm collide with the side of the robot's tablet screen without self-collision avoidance, thus the collision-free trajectory stretches the arms out to move the arms away from the side of the screen (Figure 11a,c), which is achieved by changing the shoulder pitch angle θ_2, as can be denoted by $\Delta\theta_2$ for motion I and III in Figure 12c. For motions IV and V, there are collisions with the screen surface without self-collision avoidance, and furthermore, in motion V between the two wrists, Figure 11d,e. To avoid these collisions, the collision-free trajectory lifts up the robot arms (via changing the shoulder yaw angle θ_3), straightens the arms (via changing the elbow roll angle θ_4), and stretches the arms out (via changing the shoulder pitch angle θ_2), as can be seen by $\Delta\theta_3$, $\Delta\theta_4$ and $\Delta\theta_2$ for motions IV and V in Figure 12c. As there were no self-collisions for motion II (Figure 11b), the difference between $\boldsymbol{q}_{mimic}$ and $\boldsymbol{q}_{collision\text{-}free}$ was 0, as can be seen in Figure 12c.

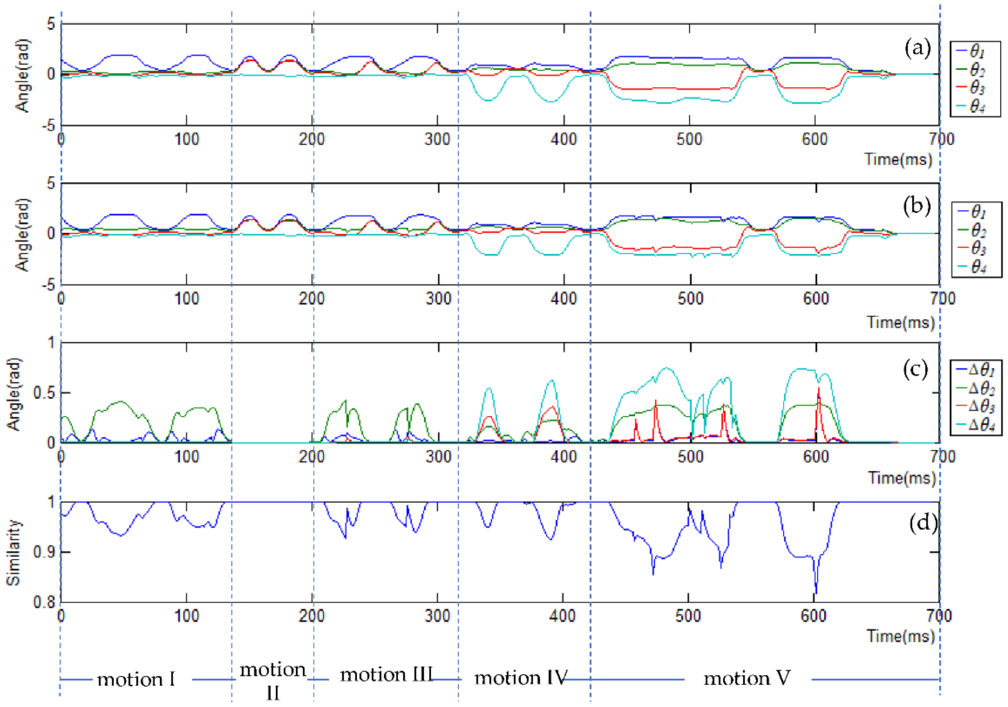


Figure 12. The trajectories of the left arm joint: (**a**) the trajectories of the four angles of the arm without self-collision avoidance; (**b**) collision-free trajectories of the arm; (**c**) the absolute angle difference between the trajectories in (**a**,**b**); (**d**) the similarity between (**a**) and (**b**).

Figure 12d shows the similarity between $\boldsymbol{q}_{mimic}$ and $\boldsymbol{q}_{collision\text{-}free}$. We define similarity as follows [15]:

$$W(\theta_i, \theta_i') = \frac{1}{1 + \sum_{i=1}^{u} \left(\frac{\theta_i - \theta_i'}{\theta_{i\_max} - \theta_{i\_min}} \right)^2}, 0 < W \leq 1 \tag{11}$$

where θ_i and θ_i' are, respectively, the demonstrated joint angle and collision-free joint angle of joint i, u is the total number of robot joints, and θ_{i_max} and θ_{i_min} are the maximum and the minimum angle of joint i. A larger W represents a higher similarity between $\boldsymbol{q}_{collision\text{-}free}$ to $\boldsymbol{q}_{mimic.}$ Namely, if $W = 1$, the collision-free pose is the same as the demonstrated pose. From Figure 12d, it can be seen that the similarity for all the motions are close to 1, that is, the collision-free trajectory is close to the trajectory that is directly mimicked.

4.2. Real-World Results with Tangy

The Tangy robot's poses during human demonstration using the imitation learning system are also shown in Figure 10 for all five motions. In particular, the figure shows the robot's arm poses during one entire demonstration session. In general, all potential collisions were avoided in real-time for all five demonstrators.

5. Conclusions

In this paper, we developed a gesture imitation learning system with self-collision avoidance using a 3D sensor. The performance of our imitation learning system with self-collision avoidance was evaluated in both simulation and on the physical robot Tangy. System performance results comparing trajectories generated by the imitation learning system to directly mapping demonstrator trajectories without self-collision showed that our proposed imitation learning system was capable of avoiding all self-collisions and generated collision-free arm trajectories that were similar to the human demonstrator motions. Furthermore, the physical robot was able to mimic the arm motions of several demonstrators in real-time when using the proposed imitation learning system.

Our current imitation learning system does not mimic the demonstrated joint angular accelerations of a gesture. As a future step, we will consider learning from the human demonstrations these accelerations because a study has shown that the level of acceleration exhibited by a robot is related to how a human interprets the affect of the robot [23].

Author Contributions: Conceptualization, T.Z. and G.N.; Methodology, T.Z.; Validation, T.Z. and W.-Y.L.; Writing-Original Draft Preparation, T.Z.; Writing-Review & Editing, G.N., W.-Y.L., and B.B.; Supervision, G.N. and B.B.; Funding Acquisition, G.N.

Funding: This research was funded in part by the AGE-WELL NCE Inc., the Natural Sciences and Engineering Research Council of Canada (NSERC), Dr. Robot Inc., and the Canada Research Chairs program (CRC).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Kinematics Model of the Tangy Robot Arm

The kinematic description of Tangy robot arms is modeled by assigning a coordinate system to each joint with its z-axis along the axis of rotation, following the rules of the Denavit-Hartenberg convention, as shown in Figure A1. Frame {0}, frame {1}, frame {2} and frame {3} are the frames assigned for shoulder roll, shoulder pitch, shoulder yaw and elbow roll, respectively. Frame {4} and frame {m} are fixed frames assigned in the end-effector and shoulder, respectively. Frame {w} is the world frame assigned in the center of the torso, as shown in Figure A1b.

The transformation from link coordinate frame {i−1} to frame {i} is represented by a 4 × 4 homogeneous matrix, as shown in Equation (A1), where $i = 0, \ldots, 4$. θ_i, a_i, d_i and α_i are the joint angle, link length, offset and twist angle, respectively:

$$ {}_{i}^{i-1}T = \begin{bmatrix} cos(\theta_i) & -sin(\theta_i) * cos(\alpha_i) & sin(\theta_i) * sin(\alpha_i) & a_i * cos(\theta_i) \\ sin(\theta_i) & cos(\theta_i) * cos(\alpha_i) & -cos(\theta_i) * sin(\alpha_i) & a_i * sin(\theta_i) \\ 0 & sin(\alpha_i) & cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A1} $$

The first three elements in the last column represent the origin position of frame {i−1} with respect to frame {i}.

The Denavit-Hartenberg (D-H) parameters of the Tangy robot arms are described in Tables A1 and A2. It is noted that d_1 = 239 mm; d_2 = 410 mm; t_1 = 170 mm. d_1 and d_2 are the length of upper arm and forearm, respectively. t_1 is the distance between frame $\{m\}$ and frame $\{w\}$. The first two sets of D-H parameters for frame $\{m\}$, and frame {0} in Tables A1 and A2 are the description of frame $\{m\}$ in frame $\{w\}$, and the description of frame {0} in frame $\{m\}$.

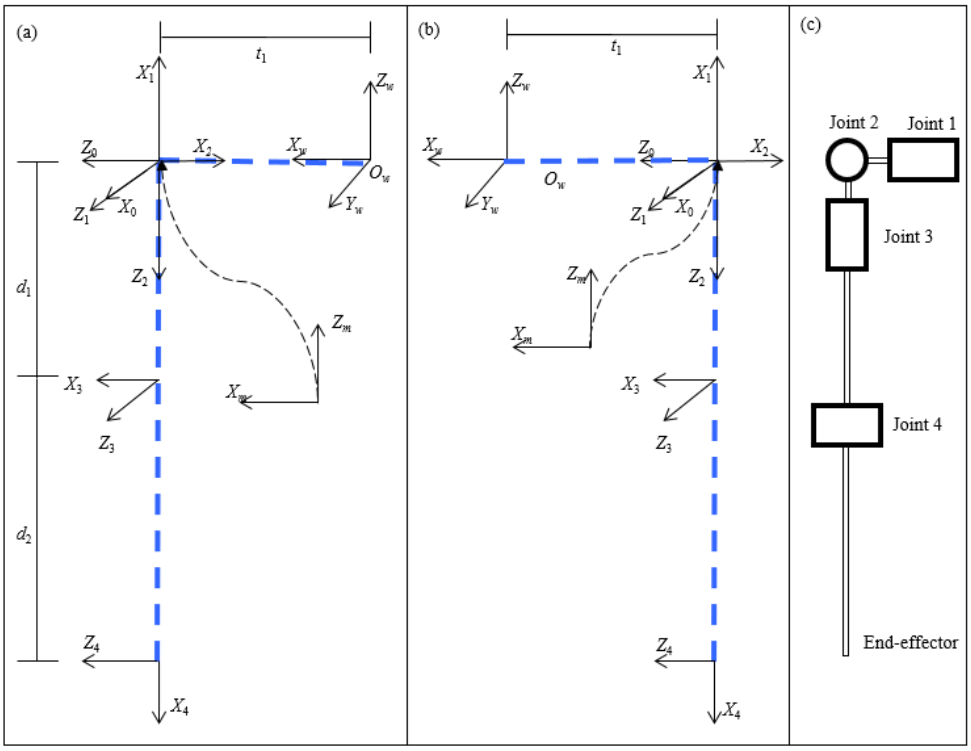


Figure A1. The D-H convention of the Tangy robot: (**a**) right arm; (**b**) left arm; (**c**) the joints and linkages diagram of the Tangy robot arm.

The homogeneous transformation relating frame {3} (elbow frame) and frame {4} (end-effector frame) to frame $\{w\}$ is denoted by Equations (A2) and (A3), respectively:

$$ {}^{w}_{3}T = {}^{w}_{m}T \cdot {}^{m}_{0}T \cdot {}^{0}_{1}T \cdot {}^{1}_{2}T \cdot {}^{2}_{3}T \tag{A2}$$

$$ {}^{w}_{4}T = {}^{w}_{3}T \cdot {}^{3}_{4}T \tag{A3}$$

Based on the D-H parameters in Tables A1 and A2 and the homogeneous transformations, ${}^{w}_{3}T$ and ${}^{w}_{4}T$ are derived. The x-y-z positions of the elbow and shoulder (the first three elements in the last column of ${}^{w}_{3}T$ and ${}^{w}_{4}T$) are derived. The positions of the left elbow and left end-effector are represented as follows:

$$x\_elbow\_left = -\, t_1 - d_1 \times cos\,(\theta_2 - \pi)$$

$$y\_elbow\_left = -d_1 \times cos\,(\theta_1 - \pi) \times cos\,(\theta_2 - \pi)$$

$$z\_elbow\_left = -d_1 \times cos\,(\theta_1 - \pi) \times cos\,(\theta_2 - \pi)$$

$$\begin{aligned} x\_end\_effector\_left = {} & d_2 \times cos\,(\theta_2 - \pi) \times cos\,(\theta_4 - \pi) - d_1 \times cos\,(\theta_2 - \pi) - t_1 \\ & - d_2 \times cos\,(\theta_3 - \pi) \times cos\,(\theta_4 - \pi) \times cos\,(\theta_2 - \pi) \end{aligned}$$

$$\begin{aligned} y\_end\_effector\_left = {} & d_2 \times cos\,(\theta_1 - \pi) \times cos\,(\theta_2 - \pi) \times cos\,(\theta_4 - \pi) - d_2 \times cos\,(\theta_4 - \pi) \\ & \times (cos\,(\theta_1 - \pi) \times cos\,(\theta_3 - \pi) - cos\,(\theta_1 - \pi) \times cos\,(\theta_2 - \pi) \\ & \times cos\,(\theta_3 - \pi)) - d_1 \times cos\,(\theta_1 - \pi) \times cos\,(\theta_2 - \pi) \end{aligned}$$

$$z\_end\_effector\_left = d_2 \times cos\,(\theta_4 - \pi) \times (cos\,(\theta_1 - \pi) \times cos\,(\theta_3 - \pi) + cos\,(\theta_2 - \pi) \times cos\,(\theta_3 - \pi) \times cos\,(\theta_1 - \pi)) - d_1 \times cos\,(\theta_1 - \pi) \times cos\,(\theta_2 - \pi) + d_2 \times cos\,(\theta_1 - \pi) \times cos\,(\theta_2 - \pi) \times cos\,(\theta_4 - \pi)$$

The positions of the right elbow and the right end-effector are represented as follows:

$$x\_elbow\_right = t_1 - d_1 \times cos\,(\theta_6 + \pi)$$

$$y\_elbow\_right = d_1 \times cos\,(\theta_6 + \pi) \times cos\,(\theta_5 - \pi)$$

$$z\_elbow\_right = d_1 \times cos\,(\theta_6 + \pi) \times cos\,(\theta_5 - \pi)$$

$$x\_end\_effector\_right = t_1 - d_1 \times cos\,(\theta_6 + \pi) + d_2 \times cos\,(\theta_6 + \pi) \times cos\,(\theta_8 - \pi) + d_2 \times cos\,(\theta_6 + \pi) \times cos\,(\theta_7 - \pi) \times cos\,(\theta_8 - \pi)$$

$$y\_end\_effector\_right = d_1 \times cos\,(\theta_6 + \pi) \times cos\,(\theta_5 - \pi) - d_2 \times cos\,(\theta_8 - \pi) \times (cos\,(\theta_5 - \pi) \times cos\,(\theta_7 - \pi) - cos\,(\theta_6 + \pi) \times cos\,(\theta_5 - \pi) \times cos\,(\theta_7 - \pi)) - d_2 \times cos\,(\theta_6 + \pi) \times cos\,(\theta_5 - \pi) \times cos\,(\theta_8 - \pi)$$

$$z\_end\_effector\_right = d_1 \times cos\,(\theta_6 + \pi) \times cos\,(\theta_5 - \pi) + d_2 \times cos\,(\theta_8 - \pi) \times (cos\,(\theta_5 - \pi) \times cos\,(\theta_7 - \pi) + cos\,(\theta_6 + \pi) \times cos\,(\theta_7 - \pi) \times cos\,(\theta_5 - \pi)) - d_2 \times cos\,(\theta_6 + \pi) \times cos\,(\theta_5 - \pi) \times cos\,(\theta_8 - \pi)$$

Table A1. The D-H Parameters for the Left Arm.

Frames	θ_i (rad)	d_i (mm)	a_i (mm)	α_i (rad)
Frame $\{m\}$	0	0	$-t_1$	0
Frame $\{0\}$	π	0	0	π
Frame $\{1\}$	$\theta_1 - \pi$	0	0	π
Frame $\{2\}$	$-\theta_2 + \pi$	0	0	π
Frame $\{3\}$	$-\theta_3 + \pi$	d_1	0	$-\pi$
Frame $\{4\}$	$\theta_4 - \pi$	0	d_2	π

Table A2. The D-H Parameters for the Right Arm.

Frames	θ_i (rad)	d_i (mm)	a_i (mm)	α_i (rad)
Frame $\{m\}$	0	0	t_1	0
Frame $\{0\}$	π	0	0	π
Frame $\{1\}$	$\theta_5 - \pi$	0	0	π
Frame $\{2\}$	$\theta_6 + \pi$	0	0	π
Frame $\{3\}$	$-\theta_7 + \pi$	d_1	0	$-\pi$
Frame $\{4\}$	$\theta_8 - \pi$	0	d_2	π

With the positions of the elbow and hand, the distance between two modeled robot body parts can be derived based on Equation (9).

References

1. Gonzalez-Pacheco, V.; Malfaz, M.; Fernandez, F.; Salichs, M.A. Teaching Human Poses Interactively to a Social Robot. *Sensors* **2013**, *13*, 12406–12430. [CrossRef] [PubMed]
2. Louie, W.G.; Nejat, G. A Learning from Demonstration System Architecture for Social Group Recreational Activities. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 808–814.
3. Louie, W.G.; Li, J.; Vaquero, T.; Nejat, G. A Focus Group Study on the Design Considerations and Impressions of a Socially Assistive Robot for Long-Term Care. In Proceedings of the 23rd IEEE International Symposium on Robot and Human Interactive Communication, Edinburgh, UK, 25–29 August 2014; pp. 237–242.

4. Clotet, E.; Martínez, D.; Moreno, J.; Tresanchez, M.; Palacín, J. Assistant Personal Robot (APR): Conception and Application of a Tele-Operated Assisted Living Robot. *Sensors* **2016**, *16*, 610. [CrossRef] [PubMed]
5. Park, G.; Ra, S.; Kim, C.; Song, J. Imitation Learning of Robot Movement Using Evolutionary Algorithm. In Proceedings of the 17th World Congress the International Federation of Automatic Control, Seoul, Korea, 6–11 July 2008; Volume 41, pp. 730–735.
6. Steil, J.J.; Röthling, F.; Haschke, R.; Ritter, H. Situated Robot Learning for Multi-modal Instruction and Imitation of Grasping. *Robot. Auton. Syst.* **2004**, *47*, 129–141. [CrossRef]
7. Billard, A.; Calinon, S.; Dillmann, R.; Schaal, S. Robot Programming by Demonstration. In *Springer Handbook of Robotics*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 1371–1394.
8. Calinon, S.; Sauser, E.L.; Billard, A.G.; Caldwell, D.G. Evaluation of a Probabilistic Approach to Learn and Reproduce Gestures by Imitation. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 2671–2676.
9. Ibrahim A, R.; Adiprawita, W. Analytical Upper Body Human Motion Transfer to NAO Humanoid Robot. *Int. J. Electr. Eng. Inform.* **2012**, *4*, 563–574. [CrossRef]
10. Mukherjee, S.; Paramkusam, D.; Dwivedy, S.K. Inverse Kinematics of a NAO humanoid Robot Using Kinect to Track and Imitate Human Motion. In Proceedings of the 2015 International Conference on Robotics, Automation, Control and Embedded Systems (RACE), Chennai, India, 18–20 February 2015; pp. 1–7.
11. Al-faiz, M.Z.; Shanta, A.F. Kinect-Based Humanoid Robotic Manipulator for Human Upper Limbs Movements Tracking. *Intell. Control. Autom.* **2015**, *6*, 29–37. [CrossRef]
12. Paulin, S.; Botterill, T.; Chen, X.; Green, R. A Specialised Collision Detector for Grape Vines. In Proceedings of the Australasian Conference on Robotics and Automation, Canberra, Australia, 2–4 December 2015; pp. 1–5.
13. Dube, C.; Tapson, J. A Model of the Humanoid Body for Self-Collision Detection Based on Elliptical Capsules. In Proceedings of the 2011 IEEE International Conference on Robotics and Biomimetics (ROBIO), Karon Beach, Thailand, 7–11 December 2011; pp. 2397–2402.
14. Zlajpah, L. Smooth Transition between Tasks on a Kinematic Control Level: Application to Self-Collision Avoidance for Two Kuka LWR Robots. In Proceedings of the 2011 IEEE International Conference on Robotics and Biomimetics (ROBIO), Karon Beach, Thailand, 7–11 December 2011; pp. 162–167.
15. Zhao, X.; Huang, Q.; Peng, Z.; Li, K. Kinematics Mapping and Similarity Evaluation of Humanoid Motion Based on Human Motion Capture. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan, 28 September–2 October 2004; pp. 840–845.
16. Luo, R.C.; Shih, B.H.; Lin, T.W. Real Time Human Motion Imitation of Anthropomorphic Dual Arm Robot Based on Cartesian Impedance Control. In Proceedings of the 2013 IEEE International Symposium Robotic and Sensors Environments (ROSE), Washington, DC, USA, 21 October 2013; pp. 25–30.
17. Ou, Y.; Hu, J.; Wang, Z.; Fu, Y.; Wu, X.; Li, X. A Real-Time Human Imitation System Using Kinect. *Int. J. Soc. Robot.* **2015**, *7*, 587–600. [CrossRef]
18. Alibeigi, M.; Rabiee, S.; Ahmadabadi, M.N. Inverse Kinematics Based Human Mimicking System Using Skeletal Tracking Technology. *J. Intell. Robot. Syst.* **2017**, *85*, 27–45. [CrossRef]
19. Johnson, S.G. The NLopt Nonlinear-Optimization Package. 2007. Available online: http://ab-initio.mit.edu/nlopt (accessed on 2 June 2016).
20. Powell, M.J. A View of Algorithms for Optimization without Derivatives. *Math. Today Bull. Inst. Math. Appl.* **2007**, *43*, 170–174.
21. OpenNI. 2010. Available online: http://www.openni.org (accessed on 20 March 2016).
22. International Organization for Standardization. *ISO/TS 15066:2016(E): Robots and Robotic Devices—Collaborative Robots*; ISO: Geneva, Switzerland, 2016.
23. Saerbeck, M.; Bartneck, C. Perception of affect elicited by robot motion. In Proceedings of the 2010 ACM/IEEE International Conference on Human-Robot Interaction (HRI), Nara, Japan, 2–5 March 2010; pp. 53–60.