

# Self collision avoidance for humanoids using circular and elliptical capsule bounding volumes

Chioniso Dube

Mobile Intelligent Autonomous Systems  
Council for Scientific and Industrial Research,  
South Africa

**Abstract**—This paper presents a self collision avoidance scheme for humanoid robots using elliptical and circular capsules as collision bounding volumes. A capsule is defined as an elliptical or circular cylinder capped with ellipsoids or spheres respectively. The humanoid body is modeled using elliptical capsules, while the moving segments, i.e. arms and legs, of the humanoid are modeled using circular capsules. The self collision distance between two capsules is computed and used to generate self collision free motion of the humanoid. Collisions are avoided by adjusting the joint angles of the colliding segments based on the collision distance and the location of the collision points. A case study of a humanoid dance is used to test the self collision avoidance method. Self collision free motion is attained by the humanoid for the entire dance.

## I. INTRODUCTION

Humanoid robots are developed for use as assistants within human environments such as homes, hospitals and offices. These robots have to perform a wide range of different tasks in a safe manner. Humanoids not only have to avoid collisions with people and objects but also have to avoid self collisions. Self collision of a humanoid robot occurs when any segment of the humanoid collides with another segment while the robot is in motion, or statically when the desired pose of the robot would result in the intersection of two segments. Such collisions not only impede the robot's motion but can cause damage to the robot itself. The motion of the robot therefore has to be restricted to avoid these self collisions.

Dariush et al, in [1] and [2], used a self collision avoidance method in which a virtual surface was placed around one of the colliding body segments. The colliding point was then moved such that it slides along the direction which is tangent to the virtual surface. A redirected velocity vector was computed and used as an input into their closed loop inverse kinematics. To avoid discontinuities in the motion, a blending function was used. This method, while effective, assumes the use of the closed loop inverse kinematics method for generating the humanoid motion.

Stasse et al, in [3], used a collision avoidance method integrated in a low-level reactive controller. The controller was based on a proximity distance with a continuous gradient. Seto et al represented the robot's elements by 'elastic elements', in [4]. 'Control points' were placed on the robot and their positions controlled to avoid collisions. A reaction force was generated on each elastic element of the robot and was transformed into a force applied at the control point.

Sugiura et al, in [5], used a collision avoidance method using null-space optimisation criteria and task intervals. Only one joint of the colliding segments was used to avoid self collisions. A virtual force was generated between the two colliding segments. Motions were continually blended between collision avoidance and target reaching. The priority between the two was changed depending on the distance between the segments.

A self collision detection scheme for humanoids, based on elliptical capsules was developed in our previous paper [6]. (See Figure 1.) The self collision detection method was shown to be simple and quick, while providing a good fit to the humanoid form. This paper now extends the previous work by presenting a simple and effective self collision avoidance method for humanoid robots that uses the elliptical capsule self collision detection. This paper will begin with an overview of self collision detection using elliptical and circular capsules in Section II. Self collision avoidance is then formulated in Section III. A case study of human dance imitation is used to validate the self collision avoidance method in Section IV.

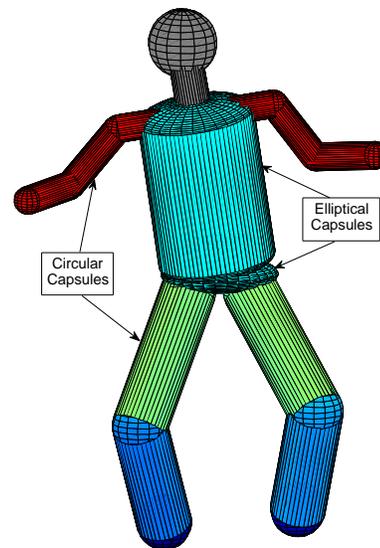


Fig. 1. Model of humanoid body using circular and elliptical capsules as bounding volumes for self-collision detection

## II. SELF COLLISION DETECTION USING CIRCULAR AND ELLIPTICAL CAPSULES

An elliptical capsule as defined in [6], is as an elliptical cylinder capped by ellipsoids. A circular capsule is defined as a circular cylinder capped by spheres. As shown in Figure 1, the arms, legs and neck of the humanoid are modeled using circular capsules. The body is modeled using elliptical capsules. If the body has a waist joint, two elliptical capsules are used. The shoulder girdle is an ellipsoid and spheres are used to represent joints.

### A. Self collision detection

The self collision detection method developed in our previous paper [6] is used to compute the collision distance between humanoid body segments and is briefly summarised here. For segments represented by two circular capsules, the critical points giving the shortest distance between the capsules axis are found using the common normal between the two axes. For two circular capsules,  $P$  and  $Q$  with axis having direction given by unit vectors  $\mathbf{u}_p$  and  $\mathbf{u}_q$ , the critical point  $\mathbf{p}_c$ , giving the shortest distance between the capsules is found as follows [6]:

$$\mathbf{p}_c = \mathbf{p}_0 - \mu_c \mathbf{u}_p l_p \quad (1)$$

where [7] [8]:

$$\mu_c = \frac{((\mathbf{p}_0 - \mathbf{q}_0) \times \mathbf{u}_q l_q) \cdot (\mathbf{n})}{\mathbf{n} \cdot \mathbf{n}} \quad (2)$$

and the common normal  $\mathbf{n}$  between the two axis is [7] [8]:

$$\mathbf{n} = \mathbf{u}_p l_p \times \mathbf{u}_q l_q \quad (3)$$

The shortest distance between axes of the capsules is then obtained by the distance between a point and a line in space as follows:

$$d = \frac{|(\mathbf{p}_c - \mathbf{q}_0) \times (\mathbf{p}_c - \mathbf{q}_1)|}{l_q} \quad (4)$$

The distance between the capsule end points is given by the distance between two points in space. The distance between a capsule end point and a capsule axis is given by the distance between a point and a line in space.

For segments represented by an elliptical capsule and circular capsule, the circular cylinder is first projected onto the coordinate frame given by the elliptical capsule axis. This gives the parametric equations of the circular capsule axis [6]:

$$x = x_0 + \lambda u_x \quad (5)$$

$$y = y_0 + \lambda u_y \quad (6)$$

where  $(x_0, y_0)$  is the capsule endpoint, and  $(u_x, u_y)$  is a unit vector in the direction of the cylinder axis. Substituting the parametric equations into the ellipse equation gives a quadratic equation in  $\lambda$  [6]:

$$F(\lambda) = \frac{(x_0 + \lambda u_x)^2}{a^2} + \frac{(y_0 + \lambda u_y)^2}{b^2} - 1 \quad (7)$$

where  $a$  is the width of the ellipse and  $b$  is the depth of the ellipse. The turning point  $\lambda_c$  of the quadratic function is given by [6]:

$$\lambda_c = \frac{-\beta}{2\alpha} \quad (8)$$

where  $\beta$  is the coefficient of  $\lambda$  and  $\alpha$  is the coefficient of  $\lambda^2$ .

The value of  $\lambda_c$  gives the critical point  $s = (x_s, y_s)$ . The closest point  $k_s = (x_k, y_k)$  on the ellipse surface to the critical point occurs such that the line connecting the two points is normal to the ellipse [9]. Replacing the ellipse normal in the equation of the ellipse gives [9]:

$$\left(\frac{ax_s}{\mu + a^2}\right)^2 + \left(\frac{by_s}{\mu + b^2}\right)^2 - 1 = 0 \quad (9)$$

Expanding the above equation gives a quartic polynomial in  $\mu$ . The largest root  $\mu_{max}$  of the polynomial leads to the closest point [9]. The closest point on the elliptical cylinder surface is then [9]:

$$x_k = \frac{a^2 x_s}{\mu_{max} + a^2} \quad (10)$$

$$y_k = \frac{b^2 y_s}{\mu_{max} + b^2} \quad (11)$$

The distance is then:

$$d = |(\mathbf{s} - \mathbf{k}_s)| \quad (12)$$

For the distance between the circular capsule end points and elliptical capsule, the distance between a point in space and an ellipse is used. For the distance between the circular and elliptical capsule end points, the equations can be adjusted to compute the distance between a point in space and an ellipsoid.

## III. SELF COLLISION AVOIDANCE

In this section the self collision avoidance method, which extends on the elliptical capsule self collision detection method, is formulated. Possible collisions that can be detected between the humanoid segments are shown in Table I. For connected segments, such as the upper and lower arm, the joint limits of the segments prevent the two segments from colliding. Collision detection and avoidance is thus only applied for unconnected segments. Due to the overall motion limits of the humanoid, certain segment collisions are unlikely or impossible. For example, collisions between the head and legs are highly unlikely due to the limited range of humanoid waist and hip joint movements. Collision avoidance is thus applied only for collisions between; an arm and the torso; an arm and head or neck; two arms; two legs; and an arm and a leg.

TABLE I  
POSSIBLE SEGMENT COLLISIONS.

$A_r$						
$A_l$	•					
$T$	•	•				
$H, N$	•	•	○			
$L_r$	•	•	○	○		
$L_l$	•	•	○	○	•	
	$A_r$	$A_l$	$T$	$H, N$	$L_r$	$L_l$

KEY: • - Likely collision, ○ - Unlikely collision  
A - Arm, T - Torso, H - Head, N - Neck, L - Leg,  $r$  - right,  $l$  - left

#### A. Collision avoidance method

To obtain collision avoidance between an arm and torso or an arm and leg, only the arm position is adjusted to avoid the collision. For collision avoidance between two arms or two legs, both limbs are moved away from the collision point.

For arm segments with a collision distance  $d$ , self collision avoidance is implemented by adjusting the glenohumeral abduction and /or glenohumeral rotation joint angles, of the humanoids. First, collision avoidance for the upper arm is obtained, followed by collision avoidance for the lower arm. For the legs, the hip abduction and /or knee flexion joint angles are adjusted. For either the arm or the leg, collision avoidance is attained, first, by adjusting the relevant abduction joint angle  $\theta_U$  by the collision avoidance angle  $\theta_{col}$ :

$$\theta_U = \theta_U - \frac{\theta_{col} \times (|d| + 1)}{d_S} \quad (13)$$

where the distance  $d_S$ , (shown in Figure 2), of the collision point from the glenohumeral joint or the hip joint is:

$$d_S = \frac{\sqrt{\mathbf{p}_c - \mathbf{p}_S}}{l} \quad (14)$$

where  $\mathbf{p}_S$  is the position of the joint and  $l$  is the length of the segment.

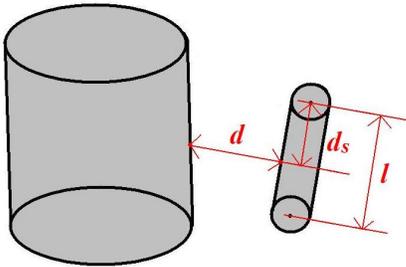


Fig. 2. Self collision avoidance parameters

Using the above equations, the collision avoidance angle  $\theta_{col}$  is first scaled by the magnitude of the collision distance to allow the colliding segments to move away from each other.  $\theta_{col}$  is then scaled by the inverse of  $d_S$ , the distance of the collision point from the joint, to ensure that the resultant posture remains close to the original posture.

The resultant joint angles are then used as input into the direct kinematics equations and the new collision distance computed. If collisions of the lower arm or lower leg still exist, collision avoidance for the lower arm or lower leg alone is conducted in a similar manner.

Collision avoidance is attained by adjusting the glenohumeral rotation joint angle or the knee flexion joint  $\theta_L$  by the collision avoidance angle  $\theta_{col}$ :

$$\theta_L = \theta_L + \frac{\theta_{col} \times (|d| + 1)}{d_S} \quad (15)$$

#### B. Joining function

The above collision avoidance method can result in discontinuities in the joint motion at the points where collision avoidance starts and ends. To remove the discontinuous motion, the following function is applied for the discontinuous joint angle  $\theta$  for collision distances within a certain threshold  $d_t$ :

$$\theta = \theta \pm \theta_{col} \times (f - |d|) \quad (16)$$

where  $f$  is a factor.

## IV. CASE STUDY - SELF COLLISION FREE DANCE

Human dance imitation of a humanoid is used as a case study to test the collision avoidance developed. The motion capture of a human dance is used as the motion input. Due to the differences between the human and humanoid forms, motion capture data of humans has to be processed to fit the humanoid structures. Self collision free motion of the humanoid is then generated. Only collisions between the arms and the torso are investigated in the case study.

#### A. Humanoid robot model

A humanoid robot is simulated for use in the case study as shown in Figure 1. The humanoid's torso has a three DOF waist and two four DOF arms comprising of a three DOF glenohumeral joint and a one DOF elbow joint. The dimensions of the humanoid are shown in Table II:

#### B. Motion capture data

The human dance motion capture data used in this study, was captured by the CSIR and the University of Johannesburg. This data set has a large range of different upper body movements which other data sets tend to lack. The motion capture data gives the  $x$ ,  $y$ , and  $z$  position coordinates of the wrist, elbow and glenohumeral joint for over 2100 frames of motion. Information for the position of the sternum is missing, thus the position is estimated using the method developed in [10].

TABLE II  
HUMANOID ROBOT MODEL DIMENSIONS.

Segment	Value (cm)
Upper arm length	25
Forearm length	25
Shoulder girdle width	46
Arm radius	4
Thigh length	43
Lower leg length	38
Leg radius	8
Upper torso length	51
Lower torso length	5
Torso depth	24
Torso width	34

### C. Motion transfer

Human motion capture data is transferred to the humanoid using inverse differential kinematics employing the Damped Least Squares (DLS) jacobian  $\mathbf{J}^*$ , [11] [12], and a joint limit weighing matrix [1]. Given the joint angles of the initial posture of the robot for the dance, the joint angles  $\mathbf{q}$  at each motion time instant  $t$ , with time interval  $\Delta t$ , in the dance can be calculated numerically using [11] [12]:

$$\mathbf{q}(t_{k+1}) = \mathbf{q}(t_k) + \dot{\mathbf{q}}(t_k)\Delta t \quad (17)$$

where  $\dot{\mathbf{q}}$  is [11] [12]:

$$\dot{\mathbf{q}} = (\mathbf{J}^*)^{-1} \mathbf{v}$$

where  $\mathbf{v}$  is the vector of link linear velocities and  $\mathbf{J}^*$  is [1]:

$$\mathbf{J}^* = \mathbf{W}^{-1} \mathbf{J}^T (\mathbf{J} \mathbf{W}^{-1} \mathbf{J}^T + \lambda^2 \mathbf{I})^{-1} \quad (18)$$

where  $\lambda$  is a damping constant and  $\mathbf{I}$  is an identity matrix and  $\mathbf{W}$  is the positive definitive matrix which allows constraints such as joint limit constraints to be added to the solution.

### D. Motion transfer and collision avoidance implementation

The humanoid robot model, the motion transfer process and the self collision avoidance were programmed and implemented using MATLAB. The process used in this case study is as follows:

- 1) Scale the human motion capture data to the humanoid robot dimensions,
- 2) Estimate the initial upper body joint angles for the first motion frame,
- 3) Compute the weighted DLS differential inverse kinematics,

- 4) Compute the resultant humanoid posture using direct forwards kinematics,
- 5) Detect self collisions using the elliptical capsule method,
- 6) Compute the resultant joint angles with added self collision avoidance,
- 7) Compute the final humanoid posture using direct forwards kinematics.

In step 1, to scale the human motion capture data to the humanoid robot dimensions, the unit vector in the direction of each human body segment is found. This is then multiplied by the length of the appropriate body segment of the humanoid.

To estimate the initial upper body joint angles of the humanoid in step 2, the desired posture of the robot is plotted graphically in MATLAB. Possible joint angles are then used as input into the forward kinematics equations of the humanoid. The resultant posture is plotted and compared to the desired posture. The joint angles are adjusted until the two figures overlap.

Differential inverse kinematics using the DLS jacobian is used to transfer human motion to the humanoid. Once the joint angles are determined in step 3, they are used as input into the forward kinematics equations in step 4 to find the resultant postures of the humanoid upper bodies.

Steps 5 and 6 then implement the self collision detection formulated previously and the self collision avoidance formulated in this study. Collision avoidance is activated at a collision distance of  $d \leq 0$  and the joining function is activated at a threshold of  $d_t < 2$ . A factor  $f = 2.5$  is used. This value is obtained empirically. The final self collision free postures of the humanoid is then computed in step 7.

## V. RESULTS

Self collision avoidance is successfully achieved for both arms of the humanoid throughout the dance. Figures 3, 4a and 4b illustrate the results of self collision avoidance. Figures 5 and 6 show the self collision joint angles with and without the joining function.

### A. Collision distance

Figure 3 shows the minimum distance between the left arm and torso for a portion of the dance. For collisions of the upper arm with the upper torso, the maximum collision distance that can be attained is 2cm. This is because the glenohumeral joint is located at a fixed distance of 2cm from the torso. Due to the length of the upper arm, no collisions occur between the upper arm and lower torso. The largest number of collisions are between the lower arm and the upper torso. In some cases, for collision distances below the threshold  $d_t$ , the joining function used causes the collision distance to change.

In Figure 4a, without self collision avoidance, the left arm intersects with the torso, while in Figure 4b the arm is moved away from the torso to avoid self collision. The arm in Figure 4b shows that collision avoidance is achieved while keeping the arm close to the desired position.

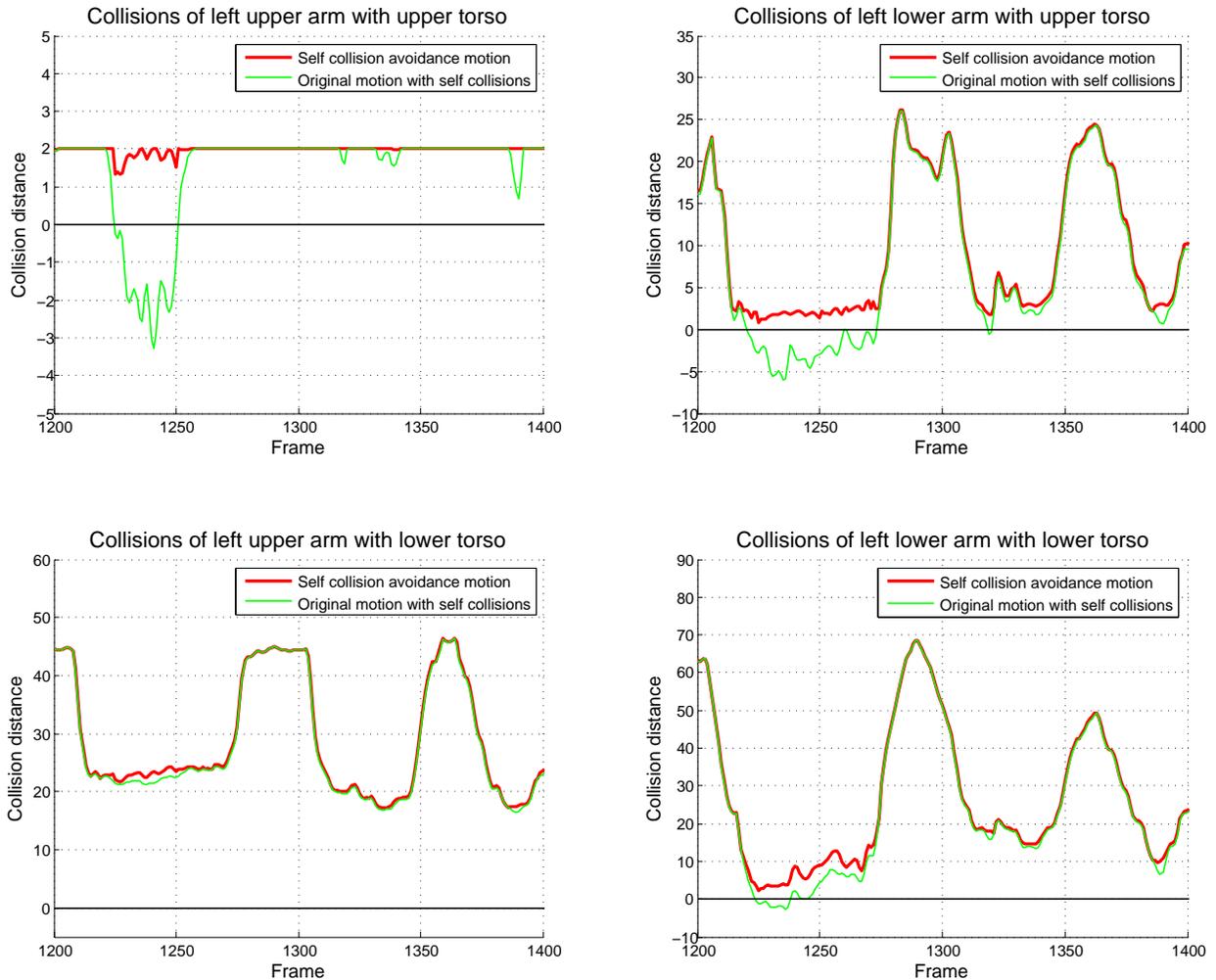


Fig. 3. Left arm and torso self collision avoidance distance (cm)

### B. Joining function

Figure 5 shows the joint angles of the humanoid with, and without self collision avoidance. The joining function is not applied for the results shown in Figure 5. As can be seen in the figure, at some points there is discontinuous and jerky motion of the joint due to collision avoidance without a joining function. Figure 6 shows the effect of the joining function. The discontinuities are removed and a smoother joint path is obtained. There is a very small offset in overall joint angles due to the joining function.

## VI. CONCLUSIONS

The self collision detection and avoidance method based on elliptical capsules and circular capsules provides a simple way of detecting and avoiding self collisions for humanoid robots. Self collision avoidance for the arms and torso was obtained successfully for a simulated humanoid robot imitating a human dance obtained from motion capture. Further work includes adapting the self collision avoidance method for real-time self

collision avoidance and applying the self collision avoidance method to a real humanoid robot.

## REFERENCES

- [1] B. Dariush, M. Gienger, A. Arumbakkam, Y. Zhu, B. Jian, K. Fujimura, and C. Goerick, "Online transfer of human motion to humanoids," *International Journal of Humanoid Robotics*, vol. 6, pp. 265–289, 2009.
- [2] B. Dariush, Y. Zhu, A. Arumbakkam, and K. Fujimura, "Constrained closed loop inverse kinematics," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 2499–2506, May 2010.
- [3] O. Stasse, A. Escande, N. Mansard, S. Miossec, P. Evrard, and A. Kheddar, "Real-time (self)-collision avoidance task on a HRP-2 humanoid robot," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 3200–3205, May 2008.
- [4] F. Seto, K. Kosuge, and Y. Hirata, "Self-collision avoidance motion control for human robot cooperation system using RoBE," in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 3143–3148, Aug. 2005.
- [5] H. Sugiura, M. Gienger, H. Janssen, and C. Goerick, "Real-time collision avoidance with whole body motion control for humanoid robots," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 2053–2058, Nov. 2007.
- [6] C. Dube, M. Tsoeu, and J. Tapson, "A model of the humanoid body for self collision detection based on elliptical capsules," in *Robotics*

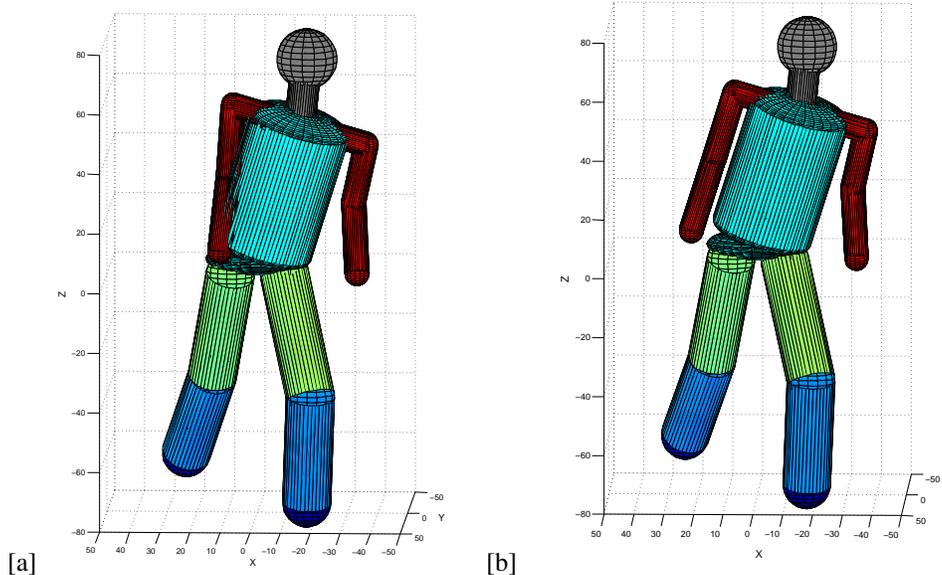


Fig. 4. [a] Humanoid without self collision avoidance. [b] Humanoid with self collision avoidance

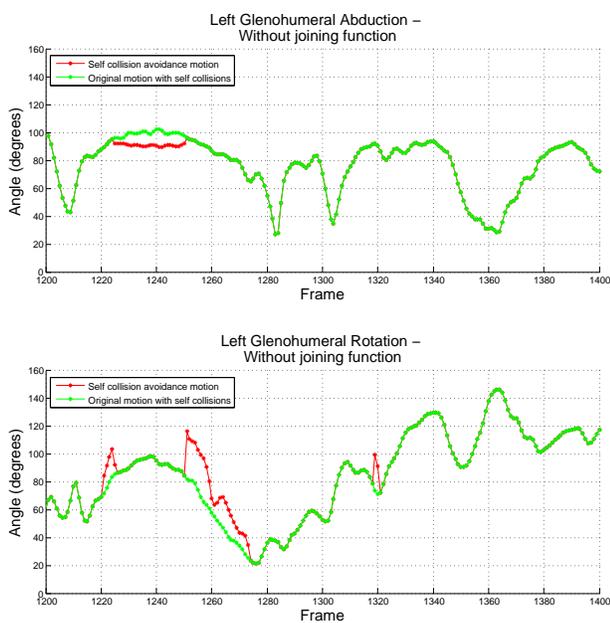


Fig. 5. Left arm joint angles without joining function, resulting in discontinuities in motion

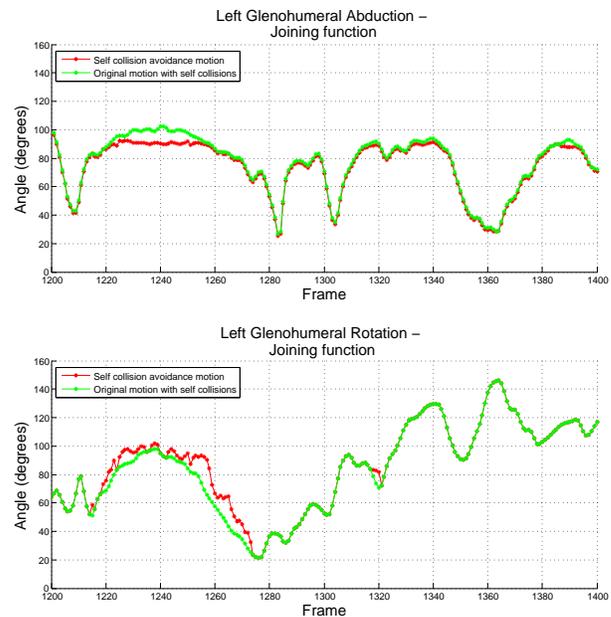


Fig. 6. Left arm joint angles with joining function, resulting in smooth motion

and Biomimetics (ROBIO), 2011 IEEE International Conference on, pp. 2397–2402, Dec. 2011.

- [7] J. Ketchel and P. Larochele, “Collision detection of cylindrical rigid bodies for motion planning,” in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 1530–1535, May 2006.
- [8] R. V. Patel, F. Shadpey, F. Ranjbaran, and J. Angeles, “A collision-avoidance scheme for redundant manipulators: Theory and experiments,” *Journal of Robotic Systems*, vol. 22, no. 12, pp. 737–757, 2005.
- [9] D. Eberly, “Distance from a point to an ellipse in 2D.” Available:

<http://www.geometrictools.com/Documentation/DistancePointToEllipse2.pdf>, Jan. 2004. [Online; last visited June 2011].

- [10] C. Dube and J. Tapson, “Kinematics design and human motion transfer for a humanoid service robot arm,” in *Robotics and Mechatronics Conference of South Africa*, Nov. 2009.
- [11] L. Sciavicco and B. Siciliano, *Modelling and Control of Robot Manipulators*. London: Springer-Verlag, 2005.
- [12] S. Chiaverini, G. Oriolo, and I. Walker, “Kinematically redundant manipulators,” in *Springer Handbook of Robotics*, pp. 245–268, Springer-Verlag, 2008.