A model of the humanoid body for self collision detection based on elliptical capsules

Chioniso Dube
Mobile Intelligent Autonomous Systems
Council for Scientific and Industrial Research,
South Africa
Email: cdube@csir.co.za

Mohohlo Tsoeu
Department of Electrical Engineering
University of Cape Town,
South Africa

Jonathan Tapson
Department of Electrical Engineering
University of West Sydney,
Australia

***Abstract*—This paper presents a self collision detection scheme for humanoid robots using elliptical and circular capsules as 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. This collision detection model provides a good fit to the humanoid body shape while being simple to implement. A case study of the self collision free workspace of the humanoid arm is then presented to illustrate the effectiveness of the collision detection scheme.**

I. Introduction

Humanoid robots are being developed for numerous applications such as service, household and healthcare robots. These robots have to perform a wide range of different tasks that require a large arm workspace, and well coordinated motion of the robot. Self collision 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. Self collision detection is thus important for motion planning, as well as for determining the workspace of humanoid robots. At its basis, self collision detection computes the closest distance between the various segments of the humanoid and determines whether any two segments are in contact. Segments are modeled using a number of different bounding volumes which can be split into two groups depending on how tightly they fit the humanoid form.

The first group of bounding volumes includes those that attempt to precisely model the shape of the humanoid, giving a tight fit to the humanoid form. For instance, convex hulls [1], [2], [3], [4] and swept-sphere volumes [5], [6], [7], [12], have been used in real-time collision avoidance schemes. However these methods tend to be computationally expensive as they generally involve a large number of collision checks for each segment. In addition, fitting models such as the swept-sphere volumes and convex hulls to the humanoid can be complex. Computing the distance between segments at fine level of detail given by these models is not always necessary. In fact, in [6], the collision detection is performed in two steps; a coarser model of the humanoid is used first to compute the proximity between segments. Only if two segments are close to each other is the finer model used to detect collision.

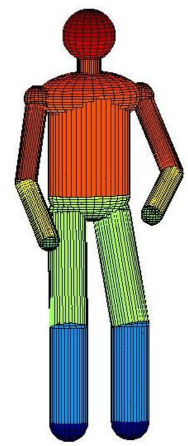

Fig. 1. Model of humanoid body using circular and elliptical capsules as bounding volumes for self-collision detection

The second group of bounding volumes use simpler representations of the humanoid shape which are generally less computationally expensive. In many applications such as human motion imitation or during the design stage of the humanoid, a simple representation of the humanoid form for self collision detection is sufficient. Common bounding volumes used in this group are circular cylinders and spheres. In [9] and in [10], circular cylinders are used to represent a robot arm. While collisions between the cylinder bodies are simple to model, the collisions of the cylinder end points are relatively complicated. Adding a spherical cap to the end points of the cylinders as done by a number of authors, simplifies end point collison detection. In [11] circular cylinders capped by spheres are used to detect self collision of a humanoid for human motion imitation. This cylinder with end caps is similar to the line swept sphere used in [12] to model the human body for collision detection. Circular cylinders and spheres are also used in [13] to model the robot body and arms. In [14], to generate self collision free motion for a humanoid from human motion, check points are placed inside the robot's body and the arm is modeled using circular cylinders. The distance between a cylinder and the check points is then used to determine collisions. While these methods may be relatively simple to implement, they do not provide a good fit for the humanoid torso shape.

In this paper a new self collision detection scheme, falling into the second category of bounding volumes, is formulated. The collision detection scheme uses a new type of bounding volume, an elliptical capsule, which is an elliptical cylinder capped at either end by ellipsoids, to model the humanoid body. The elliptical capsule gives a tighter fit than a capped circular cylinder or line swept spheres to the humanoid torso shape. It has the advantage of simplicity while proving a good fit to the humanoid form. A case study then compares collision detection using the elliptical capsule and the line swept sphere and capped cylinder as bounding volumes for the torso.

II. Representation of the humanoid body

An elliptical capsule is defined here 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.

Each capsule is made up of three sections; two end caps and a cylinder, and has the following properties:

l is the length of the cylinder
$\boldsymbol{p_0}$ and $\boldsymbol{p_1}$ are the cylinder axis end points
$\boldsymbol{u}$ is the unit vector of the cylinder axis
r is the radius of the circular cylinder
a is the width of the elliptical cylinder
b is the depth of the elliptical cylinder
c is the height of the ellipsoid

To apply the capsule models to a humanoid robot, the major dimensions of the humanoid are required. The width and depth of each torso segment of the humanoid are used for the parameters a, and b of the elliptical cylinder and the ellipsoid. The width a should also satisfy the condition:

$$a \leq \frac{L_{SG}}{2} - 10 \tag{1}$$

where L_{SG} is the shoulder girdle length in millimeters. This is to ensure that the glenohumeral joint is attached to the shoulder girdle while remaining separate from the body to enable self collision detection between the upper arm and body. The length of the elliptical cylinder and the height of the ellipsoid are given by the length L_T of the torso segment.

$$L_T = l + 2c \tag{2}$$

The height c of the ellipsoid is determined by the shape of the top and the bottom of the torso. For instance, if the torso is flat towards the top and bottom, a low value of c is used, and if the torso has a curved top and bottom, a higher value of c is used. For the arms and legs, the length and radius of each segment are used as inputs for the circular capsule parameters.

III. Self collision detection

Possible collisions between the humanoid segments are shown in Table I. Depending on the range of the humanoid joint angles, certain segment collisions are unlikely or impossible. For example, collisions between the head and legs is theoretically possible, however due to the limited range of waist and hip joint movements in most humanoids, such a collision is highly unlikely for those humanoids. For connected segments such as the upper and lower arm, the joint limit of the elbow prevents the two segments from colliding.

UA_r										
LA_r										
UA_l	●	●								
LA_l	●	●								
B	●	●	●	●						
H, N	●	●	●	●						
T_r	●	●	●	●	○	○				
L_r	●	○	●	○	○	○				
T_l	●	●	●	●	○	○	●	●		
L_l	●	○	●	○	○	○	●	●		
	UA_r	LA_r	UA_l	LA_l	B	H, N	T_r	L_r	T_l	L_l

TABLE I
Possible segment collisions.

● - Likely collision, ○ - Unlikely collision
(UA - Upper Arm, LA - Lower Arm, B - Body, H - Head, N - Neck, T - Thigh, L - Leg, $_r$ - right, $_l$ - left)

A. Collision detection algorithm

Each collision check is between two capsules representing the two test segments of the humanoid. The moving segments of the humanoid - the arms and legs - are represented by circular capsules. Collisions can thus be either between a circular capsule with an elliptical capsule (see figure 2) or a circular capsule with another circular capsule. For the simpler case of the head and the shoulder girdle, a sphere or an ellipsoid respectively are tested with the circular capsules that represent the moving humanoid segments.

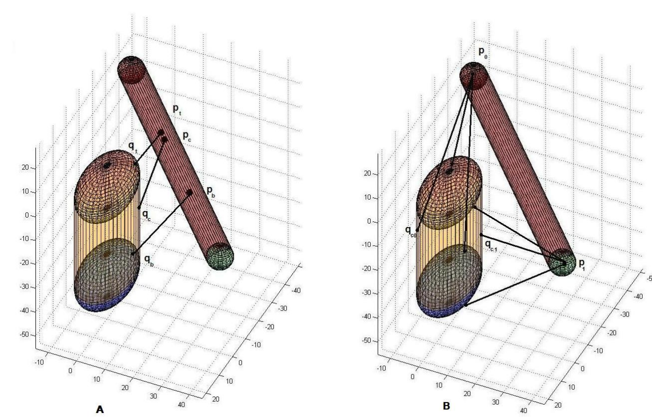

Fig. 2. Collision test points of two capsules showing the three critical closest points between capsules (A) and test points of the capsule end caps (B)

For collisions between two capsules, each collision check is then reduced to a collision check between a sphere in space and a capsule. This is done by finding the critical points that gives the closest distance between the two test capsules. There are three possible critical points representing the closest distance between the circular capsule (Capsule P) with the cylindrical section (c) and the top (t) and bottom (b) end points of the second, circular or elliptical, capsule (Capsule Q). A sphere is located at the critical points on the circular capsule P and collision checks are performed using spheres at the critical points as well as the end caps of Capsule P. Figure 2 illustrates the concept.

The algorithm for collision detection between two capsules is :

1) Find the three critical points $\boldsymbol{p_i}$ (where $i = t,\ c,\ b$) on the axis of Capsule P representing the closest points between the two capsules
2) Find distance d_i between each critical point $\boldsymbol{p_i}$ and the second test capsule, Capsule Q
3) If all of critical point distances are greater than their respective minimum distance d_m, no collision possible. i.e. $d_i > d_m$.
4) Else check if the critical points $\boldsymbol{p_i}$ lie within the Capsule P line segment. ie $|\boldsymbol{p_i} - \boldsymbol{p_0}| + |\boldsymbol{p_i} - \boldsymbol{p_1}| = l_p$
5) If any critical point lies on the Capsule P line segment and its respective distance, $d_i > d_m$, then collision occurs. And for the point $\boldsymbol{p_c}$, its corresponding closest point $\boldsymbol{q_c}$ on Capsule Q must also lie on the Capsule Q line segment for collision to occur. ie $|\boldsymbol{q_c} - \boldsymbol{q_0}| + |\boldsymbol{q_c} - \boldsymbol{q_1}| = l_q$
6) Else find distances between the two Capsule P end points with all three sections of Capsule Q
7) If each distance is greater than its minimum distance d_m then no collision occurs
8) Else collision occurs. And for collisions with the cylindrical section of Capsule Q, the closest point to the Capsule P end points must also lie on the Capsule Q line segment for collision to occur.

The minimum distances are: $d_m = r_p + r_q$ for two circular capsules, and $d_m = r_p$ for a circular capsule and elliptical capsule.

Figure 3 shows all nine possible collisions between two capsules. Collisions are between the cylindrical sections of the capsules as well as the end points of the capsules.

B. Computing the critical points

1) Critical points for circular capsule collisions: For two circular capsules, P and Q with axis having direction given by unit vectors $\boldsymbol{u_p}$ and $\boldsymbol{u_q}$, the critical points $\boldsymbol{p_c}$, $\boldsymbol{p_t}$ and $\boldsymbol{p_b}$ giving the shortest distance between the capsules are found simply using well known equations relating the closest points between two lines in space and a point and a line in space.

The critical point $\boldsymbol{p_c}$ is:

$$\boldsymbol{p_c} = \boldsymbol{p_0} + \mu_c \boldsymbol{u_p} \tag{3}$$

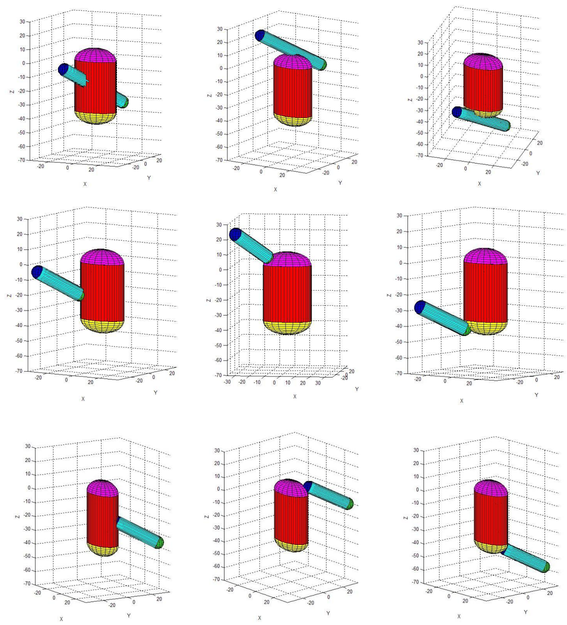

Fig. 3. Nine possible collisions between two capsules

where [10]:

$$\mu_c = \frac{((\boldsymbol{p_0} - \boldsymbol{q_0}) \times \boldsymbol{u_q} l_q) \cdot (\boldsymbol{n})}{\boldsymbol{n} \cdot \boldsymbol{n}} \tag{4}$$

and the common normal $\boldsymbol{n}$ between the two axis is:

$$\boldsymbol{n} = \frac{\boldsymbol{u_p} \times \boldsymbol{u_q}}{|\boldsymbol{u_p} \times \boldsymbol{u_q}|} \tag{5}$$

If the axis are parallel then there is no unique value for $\boldsymbol{p_c}$.

The closest points $\boldsymbol{p_t}$ and $\boldsymbol{p_b}$ on the Capsule P axis to the end points of Capsule Q are given by:

$$\boldsymbol{p_t} = \frac{(\boldsymbol{q_0} - \boldsymbol{p_0}) \cdot (\boldsymbol{p_1} - \boldsymbol{p_0})}{l_p} \boldsymbol{u_p} + \boldsymbol{p_0} \tag{6}$$

$\boldsymbol{p_b}$ is found similarly.

2) Critical points for elliptical capsule collisions: To find the critical points, the circular cylinder is first projected onto the coordinate frame given by the elliptical capsule axes (i.e the elliptical cylinder axis and the ellipse major and minor axis). The critical points $\boldsymbol{p_i}$ on the circular cylindrical capsule giving the shortest distance to the elliptical capsule are then found.

On the plane given by the elliptical cylinder axis, the parametric equations of the projected circular capsule axis are:

$$x_p = x_0 + \lambda u_x \tag{7}$$

$$y_p = y_0 + \lambda u_y \tag{8}$$

Substituting the parametric equations of the projected capsule P axis into the ellipse equation of the capsule Q cylindrical section, let:

$$F(\lambda) = \frac{(x_0 + \lambda u_x)^2}{a^2} + \frac{(y_0 + \lambda u_y)^2}{b^2} - 1 \tag{9}$$

Which is a quadratic equation in λ. The turning point λ_c of the quadratic function is given by:

$$\lambda_c = \frac{-\beta}{2\alpha} \tag{10}$$

where β is the coefficient of λ and α is the coefficient of λ^2.

The critical point $\boldsymbol{p_c}$ is:

$$\boldsymbol{p_c} = \boldsymbol{p_0} + \lambda_c \boldsymbol{u_p} \tag{11}$$

λ_c gives the critical point and has the following properties:

If $F(\lambda_c) = 0$, then $\boldsymbol{p_c}$ touches the ellipse.

If $F(\lambda_c) < 0$, then $\boldsymbol{p_c}$ lies inside the ellipse.

If $F(\lambda_c) > 0$, then $\boldsymbol{p_c}$ lies outside the ellipse and gives the closest distance to the ellipse.

If the axis are parallel then there is no unique value for $\boldsymbol{p_c}$.

For the critical point $\boldsymbol{p_t}$ between the circular capsule with the ellipsoid end point, the formulation can be extended to an ellipsoid as follows:

$$F(\lambda) = \frac{(x_0 + \lambda u_x)^2}{a^2} + \frac{(y_0 + \lambda u_y)^2}{b^2} + \frac{(z_0 + \lambda u_z)^2}{c^2} - 1 \tag{12}$$

and for the critical point $\boldsymbol{p_b}$

$$F(\lambda) = \frac{(x_0 + \lambda u_x)^2}{a^2} + \frac{(y_0 + \lambda u_y)^2}{b^2} + \frac{(z_0 + \lambda u_z + l_q)^2}{c^2} - 1 \tag{13}$$

C. Computing the shortest distance between capsules

For collision detection of Capsule P and Q, five spheres representing Capsule P critical points and end points are placed with centers $\boldsymbol{p_b}$, $\boldsymbol{p_t}$, $\boldsymbol{p_c}$, $\boldsymbol{p_0}$, $\boldsymbol{p_1}$. For the general case of a sphere with a capsule collision, let $\boldsymbol{s}$ represent the center of any sphere.

1) Sphere - circular capsule collisions: As shown in figure 4, the shortest distance between the cylinder section of capsule Q and sphere S is found using the distance of the point $\boldsymbol{s}$ to the cylinder axis of P. The distance between the ends points of the capsule and the sphere is given by the distance between two points in space.

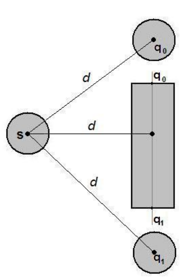

Fig. 4. Collision testing of sphere and circular capsule

The shortest distance between the center of the sphere $\boldsymbol{s}$ and the axis of Capsule Q is:

$$d = \frac{|(\boldsymbol{s} - \boldsymbol{q_0}) \times (\boldsymbol{s} - \boldsymbol{q_1})|}{l_q} \tag{14}$$

The shortest distance between the center of the sphere $\boldsymbol{s}$ and the end points of Capsule Q is:

$$d = |(\boldsymbol{s} - \boldsymbol{q_0})| \tag{15}$$

2) Sphere - elliptical capsule collisions: To find the shortest distance between the sphere and the elliptical capsule, the sphere is projected onto the cooedinate frame given by the elliptical capsule axes (i.e the elliptical cylinder axis and the ellipse major and minor axis). The closest distance between the sphere and the surface of the elliptical cylinder and ellipsoids (see figure 5) is the found. For the elliptical cylinder scetion, the shortest distance lies on the plane given by the center of the sphere and the elliptical cylinder axis. The plane contains an ellipse and a circle.

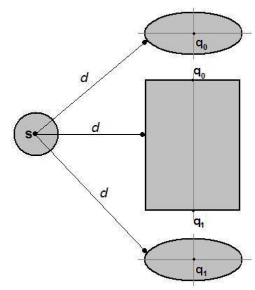

Fig. 5. Collision testing of sphere and elliptical capsule

[16] gives the equations relating the closest point on an ellipse to a given point. The closest point $k_s = (x_k, y_k)$ on the ellipse surface to the center of the circle $s = (x_s, y_s)$ occurs such that the line connecting the two points is normal to the ellipse. The ellipse normal at any point (x, y) on the ellipse is given by: [16]

$$\frac{1}{2}\nabla\left(\frac{x^2}{a^2} + \frac{y^2}{b^2} - 1\right) = \left(\frac{x}{a^2}, \frac{y}{b^2}\right) \tag{16}$$

Orthogonality condition gives [16]:

$$(x_s - x_k, y_s - y_k) = t\left(\frac{x_k}{a^2}, \frac{y_k}{b^2}\right) \tag{17}$$

Replacing in the equation of the ellipse gives [16]:

$$\left(\frac{a x_s}{t + a^2}\right)^2 + \left(\frac{b y_s}{t + b^2}\right)^2 - 1 = 0 \tag{18}$$

Expanding gives a quartic polynomial in t. The largest root t_{max} of the polynomial leads to the closest point [16]. The roots of a quartic polynomial can be obtained using a number of methods e.g. Newton-Raphson. However some programing languages such as Matlab have built in functions that can be used to easily and quickly compute the roots.

The closest point on the elliptical cylinder surface is [16]:

$$x_k = \frac{a^2 x_s}{t_{max} + a^2} \tag{19}$$

$$y_k = \frac{b^2 y_s}{t_{max} + b^2} \tag{20}$$

To check if the point lies outside of the ellipse:

$$F(k) = \frac{{x_k}^2}{a^2} + \frac{{y_k}^2}{b^2} - 1 \tag{21}$$

if $F(k) > 0$ the point is outside the ellipse.
The distance is then:

$$d = |(\boldsymbol{s} - \boldsymbol{k_s})| \tag{22}$$

To calculate the distance from the sphere to the ellipsoid end caps, the above equations can be extended as follows [15]:

$$\left(\frac{ax_s}{t + a^2}\right)^2 + \left(\frac{by_s}{t + b^2}\right)^2 + \left(\frac{cz_s}{t + c^2}\right)^2 - 1 = 0 \tag{23}$$

for the bottom end cap

$$\left(\frac{ax_s}{t + a^2}\right)^2 + \left(\frac{by_s}{t + b^2}\right)^2 + \left(\frac{c\,(z_s + l_p)}{t + c^2}\right)^2 - 1 = 0 \tag{24}$$

and the closest point $\boldsymbol{k_s} = (x_k, y_k, z_k)$ formulated as above.

IV. Case study: Self-collision free arm workspace

Using the forward kinematics [17] of a humanoid arm, the arm workspace is computed with the added collision detection constraint. To compute the workspace, each arm joint angle is cycled through its minimum to maximum range of motion and a discrete representation of the wrist and elbow position is found. Any arm configurations that violate the collision detection check are invalid and the corresponding wrist and elbow positions are removed. The workspace is computed for a simulated humanoid with the properties shown in Table II and Table III:

Segment	Length (mm)
Upper arm	365
Lower arm	270
Shoulder girdle	465
Body depth	190
Body width	445

TABLE II
Humanoid physical properties

Joint	Min	Max
Glenohumeral Adduction	-40	180
Glenohumeral Flexion	-60	180
Glenohumeral Rotation	-90	90
Elbow Flexion	0	150

TABLE III
Humanoid Joint Angles

Two cases are considered. The first case computes the self collision free workspace with the torso modeled as a circular capsule which is similar to the single line swept sphere or the capped circular cylinder used in [12] and [11] respectively. The second case computes the self collision free workspace using an elliptical capsule, as formulated in this paper, to represent the torso.

A. Results

Figure 7 shows the results of the self collision detection for the wrist workspace of a typical 7 degree of freedom humanoid arm, using an elliptical capsule to represent the torso. The workspace points are mapped to the nearest 2cm and each graph shows a cross-section with a range of of points equal to 10cm. Figure 8 shows the self collision free workspace that would be obtained using a circular capsule to represent the humanoid torso. Using the circular capsule as a bounding volume for the torso results in 2.5% less number of feasible workspace points compared to using an elliptical capsule as a bounding volume. Figure 6 shows the difference in workspace points obtained using an elliptical capsule versus a circular capsule. As shown in Figure 6, not only the workspace points close to the robot torso are affected, but the overall reach of the arm is restricted by using the circular capsule to represent the torso.

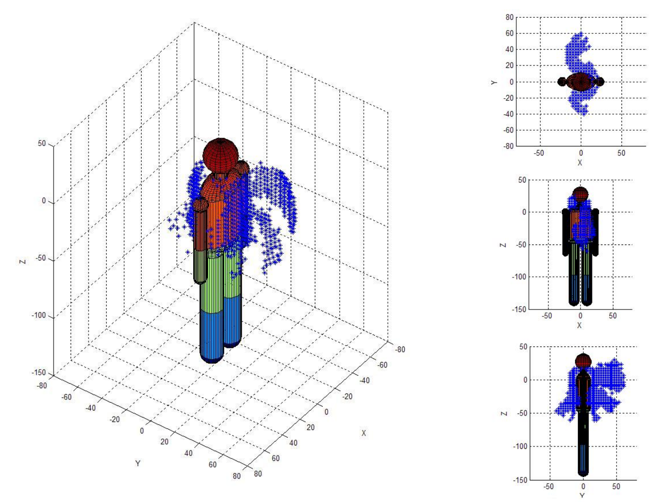

Fig. 6. Difference in workspace points obtained using elliptical capsule versus a circular capsule for the torso

V. Conclusion

The self collision detection scheme based on elliptical and circular capsules as bounding volumes provides a simple and quick way of modeling self collisions for humanoid robots. It provides a good fit to the humanoid form and can be used effectively for applications such as determining the self collision free workspace and can potentially be used for humanoid motion planning. Further work includes applying the self collison detection scheme to a real humanoid robot and using the scheme in motion planning for humanoid robots.

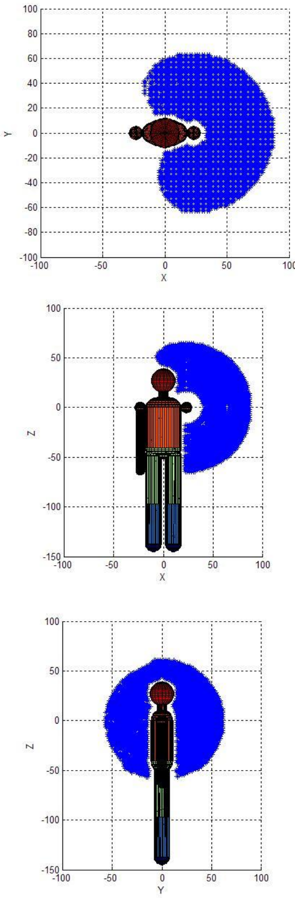

Fig. 7. Self-collision free workspace of a humanoid arm using an elliptical capsule

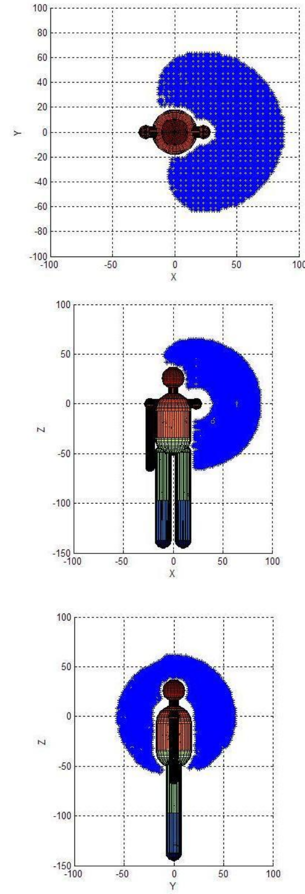

Fig. 8. Self-collision free workspace of a humanoid arm using a circular capsule

REFERENCES

[1] O. Stasse, A. Escande, N. Mansard, S. Miossec, P. Evrard, A. Kheddar, "Real-Time (Self)-Collision Avoidance Task on a HRP-2 Humanoid Robot", in *Proceedings of the IEEE International Conference on Robotics and Automation* , 2008.

[2] K. Okada, M. Inaba, "A Hybrid Approach to Practical Self Collision Detection System of Humanoid Robot", in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* , 2006.

[3] K. Okada, M. Inaba, H. Inoue, "Real-time and Precise Self Collision Detection System for Humanoid Robots", in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1060-1065, 2005.

[4] J. Kuffner, K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba, H. Inoue, "Self-collision detection and prevention for humanoid robots ", in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2265-2270, 2002.

[5] H. Sugiara, M. Gienger, H. Janssen, C. Goerick, " Real-Time Collision Avoidance with Whole Body Motion control for Humanoid Robots", in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* , 2007.

[6] H. Sugiara, M. Gienger, H. Janssen, C. Goerick, " Real-Time Collision Avoidance for Humanoid Robots by means of Nullspace Criteria and Task Intervals", in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots* , 2006.

[7] E. Larsen, S. Gottschalk, M. C. Lin, D. Manocha, "Fast Proximity Queries with Swept Sphere Volumes", Department of Computer Science, University of North Carolina, Chapel Hill, Tech. Rep., 1999.

[8] J. Tornero, J. Hamlin, R. B. Kelly, "Spherical-Object Representation and fast Computation for Robotic Applications", in *Proceedings of the IEEE International Conference on Robotics and Automation* , pp. 1602-1608 ,1991.

[9] R. V. Patel, F. Shadpey, F. Ranjbaran, J. Angles, "A Collision-Avoidance Scheme for Redundant Manipulators: Theory and Experiments", *Journal of Robotic Systems*, vol 22(12), pp. 737-757, 2005.

[10] J. Ketchel, P. Larochelle, "Collision detection of cylindrical rigid bodies for motion planning" in *Proceedings of the IEEE International Conference on Robotics and Automation* , pp. 1530-1535, 2006.

[11] A. Safonova, N.S. Pollard, J.K. Hodgins, "Optimizing Human Motion for the Control of a Humanoid Robot", in *Proceedings of the IEEE International Conference on Robotics and Automation* , 2003.

[12] J.A. Corrales, F.A. Candelas, F. Torres, "Safe humanrobot interaction based on dynamic sphere-swept line bounding volumes", *Robotics and Computer-Integrated Manufacturing*, vol 27, pp. 177185, 2011.

[13] F. Seto, K. Kosuge, Y. Hirata, "Self-collision Avoidance Motion Control for Human Robot Cooperation System using RoBE", in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 50-55, 2005.

[14] M. Ruchanurucks,S. Nakaoka, "Offline and online trajectory generation with sequential physical constraints", in *Proceedings of the IEEE International Conference on Robotics and Biomimetics* , 2008.

[15] D. Eberly, "Distance from a Point to an Ellipsoid", Geometric Tools, LLC, October 2008. URL http://www.geometrictools.com/Documentation/DistancePointToEllipsoid.pdf

[16] D. Eberly, "Distance from a Point to an Ellipse in 2D", Geometric Tools, LLC, January 2004. URL http://www.geometrictools.com/Documentation/DistancePointToEllipse2.pdf

[17] Sciavicco, L., Siciliano, B., *Modelling and Control of Robot Manipulators,* London: Springer-Verlag, 2005.