

Multiple-point kinematic control of a humanoid robot

Agostino De Santis¹, Giuseppe Di Gironimo², Luigi Pelliccia², Bruno Siciliano¹ and Andrea Tarallo²

¹ *PRISMA Lab, Dipartimento di Informatica e Sistemistica*

² *VRTEST Lab, Dipartimento di Meccanica ed Energetica*

Università degli Studi di Napoli Federico II, Italy

e-mail: siciliano@unina.it

Abstract. Robots designed to operate in everyday domains have to move in environments designed for the humans. Therefore, they will often have a humanoid kinematic structure. Simple and efficient kinematic models are needed for motion control of this class of robots. An algorithm is presented to solve the inverse kinematics problem in the presence of a number of control points arbitrarily located on the whole robot body, using an augmented Jacobian approach and including posture control. Simulation experiments are reported, showing the effectiveness of the proposed approach.

Key words: Multiple-point control, inverse kinematics, humanoid robot.

1 Introduction

Modelling and control issues for a humanoid robot with a high degree of redundancy are addressed in this paper, considering a multiple-control point inverse kinematics algorithm. The increasing attention of the robotics community towards *humanoid robotics* [1], [2], [3] is not simply related to the ancestral ambition of *building something that looks like a human*, but has also an immediate and objective reason. In particular, it arises from the apparently obvious fact that for all actions performed daily by humans, the objects that they manipulate and the environment where they live have been built or structured “on a human scale”. For instance, all the objects that we manipulate have been conceived based on the shape of our hands.

If we really want to build machines able to cooperate with human beings, we need to design robots that not only can move through environments designed for humans, but can also handle objects particularly suited to our physical structure and our behavior. For instance, bipedal robots could potentially move in the same space where people work, such as an industrial plant with stairs and handrails specifically designed for human use. In this way, these robots could cooperate with humans and even collaborate with one another using already working ordinary tools or machinery. Further considerations can be made even under the aspect of human-robot and robot-robot communication channels [4]; for instance humanoid robots could even

be used in the therapy of some forms of mental disorders [5]. It is therefore necessary to develop efficient models that allow us to accurately control the motion of humanoid robots. On the other hand, it is also necessary to develop simulation tools to study robots behavior in *unstructured* environments, considering the safety issues arising from the interaction with humans.

The model of humanoid robot that will be described in the following sections is also suitable for simulating the behavior of human beings in a virtual environment. This can be very useful for ergonomics analyses or even for manual processes simulations.

Manufacturing companies, indeed, have now taken the concept of “man adaptability” as a basic parameter of quality for their products, and thus they are giving an increasing attention to ergonomic analyses, even from the early stages of design, [6], [7], [8], [9]. The so-called *virtual manikins*, provided by many process simulation software, are essentially virtual kinematic chains consisting of several segments and joints. The length of their segments are derived from anthropometric databases, which can be queried with respect to different percentiles in the population. However, these software tools are generally too much complicated to be handled and the so-called “process simulation” often becomes a very time-consuming task, mainly because of the difficulty in controlling the kinematic chain of the virtual humanoid.

Therefore, developing efficient algorithms aimed at controlling a high-articulated chain, such as the human mechanical structure, is very interesting even for fields apparently unrelated to robotics.

2 Kinematic modelling

The basic idea in the described approach is to control the highly redundant kinematic structure of a humanoid robot, by means of only few control points which can move on the structure [10], [11], [12]. If one considers a serial manipulator and its direct kinematics equation, changing the value of its Denavit–Hartenberg (DH) parameters results in the kinematics equations of another manipulator, whose end-effector is located before the real one: that is equivalent to moving the control point of the structure. If the DH values are described in a symbolic form, they are such to identify an arbitrary point as a virtual end-effector of a smaller manipulator considered for the control. An arbitrary number of such control points can be considered. It is then possible to consider these control points as fixed or moving. Related to a humanoid, different kinematic chains will be considered. In addition, during the carrying of a certain task, the postures taken by our kinematic structure largely depend on balancing and mechanics issues, that are only partially related to the considered task. This implies the need for an additional posture control.

For this purpose, the goal has been to develop an inverse kinematics algorithm that allows concentrating only on a limited number of task-related control points, without the need of specifying the Degrees of Freedom (DoFs) of the chain for

posture control. The position of the Center of Mass (CoM) of the humanoid has then been taken into account: it is calculated on-line and always kept consistent with the balancing issues of the mechanical structure, by identifying the time-varying CoM as an additional moving control point.

It is worth noticing that the selected control points on the humanoid can be also selected automatically depending on the task and the environment, giving a very powerful tool for simulation.

2.1 Hierarchical model of humanoid robot

Firstly, in order to take advantage of the systematic approaches typical of serial robots, the humanoid has been modeled as the combination of four kinematic chains, which share the same starting point, called *root*. The resulting model is the hierarchical structure shown in Figure 1. Starting from this graph, it is possible to build up the DH model of the whole kinematic chain (Figure 2).

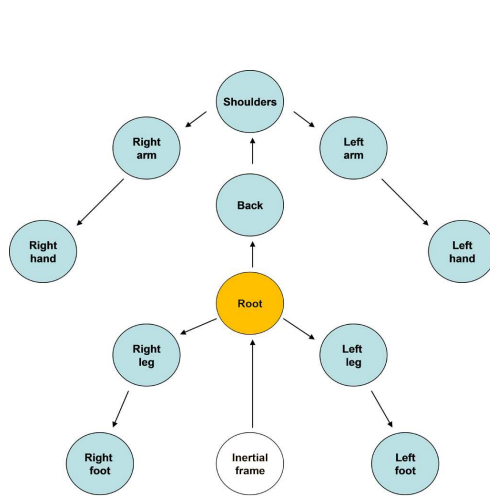


Fig. 1 Hierarchical model of humanoid robot

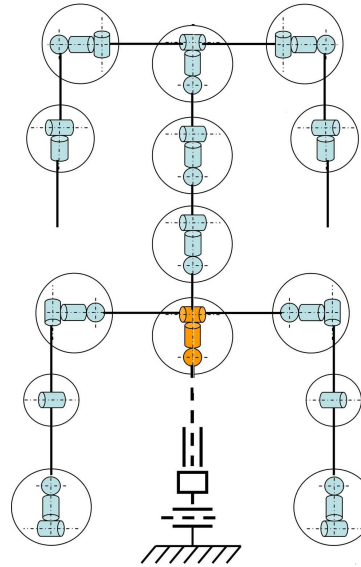


Fig. 2 DH model of humanoid robot

In particular, we can define a number of direct kinematics equations, with respect to the *root* reference frame. One or more control points on the provided chains can be selected, by considering the proper set of DH parameters that specify such points.

As we can see, the position and orientation of the *root* node with respect to the reference frame is specified by introducing 6 *virtual joints* (see Section 2.2). Thus, the considered kinematic structure has 39 DoFs in all.

This kind of modeling has the advantage of simplicity, but generally it may cause a physical consistency problem, since some links (as well as all the virtual links) are shared among different kinematic chains. For instance the “back” of the virtual humanoid is shared between its right and left arms. This issue and its solution are discussed in Section 2.5.

2.2 Virtual joints

Now, we must describe the position and orientation of the multi-legged kinematic chain with respect to an inertial frame. For industrial robots identifying such a frame is intuitive, because they have a fixed base. A humanoid robot, instead, is bound to the ground by a one-way constraint, that is the current support plane, for instance one foot.

However, this reference periodically changes during the walk, thus we apparently cannot identify a fixed base starting from which the DH method can be applied (Figure 3).

Moreover, the presence of multiple end-effectors (two hands and two feet) implies the need to describe the position and orientation of many frames, differently from industrial robots, in which the kinematic chain has only one end-effector. This problem has been overcome using the *virtual joints* approach [13]. Namely, the humanoid robot has been conceived as connected to the ground plane through a virtual manipulator consisting of three prismatic and three revolute joints, which characterize its position and orientation. The attaching point has been called *root* (Figure 4).

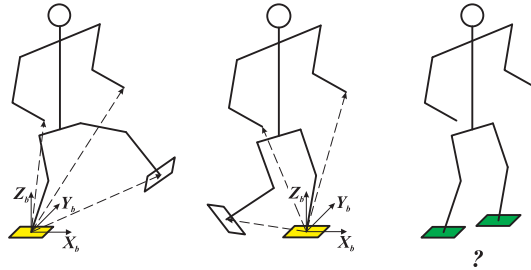


Fig. 3 Which reference frame?

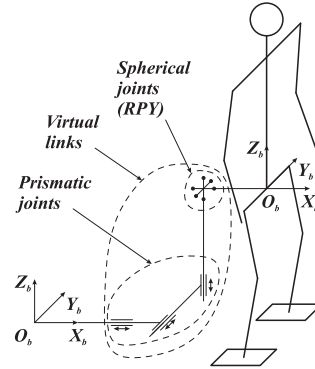


Fig. 4 Virtual joints approach

With this approach, the hands and the feet (and even any other control point) are simply end-effectors that can be controlled with velocity references. In other words, the posture of the virtual humanoid is completely specified by the following

parameters vector:

$$\mathbf{q} = [\mathbf{q}_r^T \ q_1 \ q_2 \ \dots \ q_n]^T$$

where $\mathbf{q}_r = [\mathbf{p}_r^{oT} \ \omega_r^{oT}]^T$ identifies the *root* frame.

Moreover, virtual joints technique makes unnecessary the management of closed kinematic chains during the phase of double support. Indeed, this condition becomes merely equivalent, from a kinematic point of view, to imposing a null velocity reference to the feet.

2.3 Augmented Jacobian

Each chain has its own direct kinematic function, therefore a Jacobian matrix can be computed for a generic control point of the structure. Generally, considering n control points we can define the following set of equations:

$$\begin{aligned} \mathbf{v}_1 &= \mathbf{J}_1 \dot{\mathbf{q}} \\ \mathbf{v}_2 &= \mathbf{J}_2 \dot{\mathbf{q}} \\ &\vdots \\ \mathbf{v}_n &= \mathbf{J}_n \dot{\mathbf{q}} \end{aligned} \tag{1}$$

where the generic element $\mathbf{J}_i$ is the Jacobian matrix related to a specific control point.

It is understood that, if the generic joint variable q_j does not affect $\mathbf{v}_n$, it is $(J_n)_{ij} = 0$. This set of equations can be summarized as

$$\mathbf{v} = \mathbf{J}_{AU} \dot{\mathbf{q}} \tag{2}$$

where $\mathbf{J}_{AU}$ is the so-called *Augmented Jacobian*. On one hand, this approach allows us to solve the inverse kinematic problem with only one Closed-Loop Inverse Kinematics (CLIK) algorithm [14]. On the other hand, the trajectories defined for the control points will be all treated as *primary tasks*, unlike other solution methods do, such as null-space based approaches [15], [16].

In particular, in order to define the structure of $\mathbf{J}_{AU}$, the vector $\dot{\mathbf{q}}$ must be properly sorted. Since the humanoid structure is composed by four kinematic chains, we can write four different vectors of unknowns:

$$\begin{aligned} \dot{\mathbf{q}}_1 &= [\dot{\mathbf{q}}_r^T \ \dot{\mathbf{q}}_{rl}^T]^T && \text{right leg} \\ \dot{\mathbf{q}}_2 &= [\dot{\mathbf{q}}_r^T \ \dot{\mathbf{q}}_{ll}^T]^T && \text{left leg} \\ \dot{\mathbf{q}}_3 &= [\dot{\mathbf{q}}_r^T \ \dot{\mathbf{q}}_b^T \ \dot{\mathbf{q}}_{ra}^T]^T && \text{right arm} \\ \dot{\mathbf{q}}_4 &= [\dot{\mathbf{q}}_r^T \ \dot{\mathbf{q}}_b^T \ \dot{\mathbf{q}}_{la}^T]^T && \text{left arm} \end{aligned}$$

where $\dot{\mathbf{q}}_r$ are the velocities of the virtual joints that are shared among four kinematic chains. These vectors can be summarized in only one vector of unknowns

$$\begin{aligned}\dot{\mathbf{q}} &= [\dot{\mathbf{q}}_r^T \ \dot{\mathbf{q}}_{rl}^T \ \dot{\mathbf{q}}_{ll}^T \ \dot{\mathbf{q}}_b^T \ \dot{\mathbf{q}}_{ra}^T \ \dot{\mathbf{q}}_{la}^T]^T \\ &= [\dot{q}_1 \ \dot{q}_2 \ \dots \ \dot{q}_{39}]^T.\end{aligned}\tag{3}$$

With this choice, the Augmented Jacobian takes on the following form

$$\mathbf{J}_{AU} = \begin{bmatrix} \mathbf{J}_{ri} & \mathbf{J}_{rli} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{J}_{ri} & \mathbf{0} & \mathbf{J}_{lli} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{J}_{ri} & \mathbf{0} & \mathbf{0} & \mathbf{J}_{bi} & \mathbf{0} & \mathbf{0} \\ \mathbf{J}_{ri} & \mathbf{0} & \mathbf{0} & \mathbf{J}_{bi} & \mathbf{J}_{rai} & \mathbf{0} \\ \mathbf{J}_{ri} & \mathbf{0} & \mathbf{0} & \mathbf{J}_{bi} & \mathbf{0} & \mathbf{J}_{lai} \end{bmatrix}.\tag{4}$$

The matrix $\mathbf{J}_{AU}$, with the proposed humanoid model, has 39 columns, while the number of its rows depends on the number of control points considered.

2.4 Center-of-Mass Jacobian

Unlike industrial manipulators and, more generally, non-ambulatory robots, bipedal robots must concern about their balance while performing any task. If this does not happen, obviously, the robot would lean over and fall. Moreover, humanoid robots are inherently hyper-redundant, having a much higher number of joints than traditional industrial robots. Consequently, there are many postures that achieve the same position for its body terminals, corresponding to control points. Also, taking into account the balancing issues allows the humanoid to attain more natural posture, similar to those of human beings.

For this, the Virtual End-Effectors (VEEs) technique [10] has been implemented also with respect to the center of mass (CoM) of the digital humanoid, which becomes a further control point for the kinematic chain. In particular, the trajectory of the CoM can be defined in such a way that its vertical projection on the current support plane (namely, the Center of Pressure, CoP) belongs to the stability polygon formed by the feet (Figure 5). It is worth noticing that the constraint about the CoP will be treated as a primary task, as well as the other tasks.

The basic idea is to obtain a differential relationship like

$$\mathbf{v}_G = \mathbf{J}_G \dot{\mathbf{q}},\tag{5}$$

where $\mathbf{J}_G$ is a $3 \times n$ matrix, called Center-of-Mass Jacobian. Then, Equation (5) will be inserted in Equation (1) as a further control point.

For this purpose, we can define the CoM of a kinematic chain composed of n links as

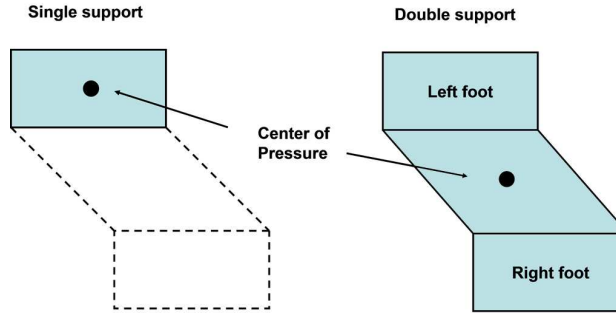


Fig. 5 Center of Pressure and support plane

$$\mathbf{p}_G = \frac{\sum_{i=1}^n m_i \mathbf{p}_{G_i}}{\sum_{i=1}^n m_i} = \frac{1}{m} \sum_{i=1}^n m_i \mathbf{p}_{G_i}. \quad (6)$$

Equation (6) can be derived with respect to time

$$\mathbf{v}_G = \frac{1}{m} \sum_{i=1}^n m_i \mathbf{v}_{G_i}. \quad (7)$$

Since the center of mass of each link can be considered as a *Virtual End-Effector* (VEE), it is always possible to write the differential relationship

$$\mathbf{v}_{G_i} = \mathbf{J}_{G_i} \dot{\mathbf{q}}$$

where

$$\mathbf{J}_{G_i} = \begin{bmatrix} \gamma_{x,1} & \cdots & \gamma_{x,i} & 0 & \cdots & 0 \\ \gamma_{y,1} & \cdots & \gamma_{y,i} & 0 & \cdots & 0 \\ \gamma_{z,1} & \cdots & \gamma_{z,i} & 0 & \cdots & 0 \end{bmatrix}. \quad (8)$$

Indeed, if the vector $\dot{\mathbf{q}}$ has been properly sorted, $\mathbf{v}_{G_i}$ can be affected at most by the first i links of the chain. Now, Equation (7) can be written as

$$\mathbf{v}_G = \left[\frac{1}{m} \sum_{i=1}^n m_i \mathbf{J}_{G_i} \right] \dot{\mathbf{q}}. \quad (9)$$

By comparing Equations (5) and (9), we can finally assume

$$\mathbf{J}_G = \frac{1}{m} \sum_{i=1}^n m_i \mathbf{J}_{G_i}. \quad (10)$$

Given $\mathbf{J}_G$, the velocity of CoM $\mathbf{v}_G$ becomes a further control point for the kinematic chain. Thus, we can insert the kinematic relation (5) in the equations set (1). As a result, we will have an Augmented Jacobian matrix with two more rows, that are related to the components of $\mathbf{v}_G$ projected on the current support plane. As mentioned above, the implemented inversion algorithm assures that a constraint on CoM

velocity becomes a high-priority task to be achieved, without using null-space projection.

Finally, it is worth emphasizing that the expression of $\mathbf{J}_G$ suggests also the possibility to use the *kinetostatic duality* [14] to compute the joint torques corresponding to forces applied to the structure.

2.5 Conflicting tasks

As mentioned above, some tracts of the humanoid structure are shared among apparently different kinematics chains. For instance, the right and left arms of the humanoid share a common tract, namely the back. But, if actually the left and right arms were modelled as *independent* chains, they could perform different or even conflicting tasks.

For this, inverse kinematics algorithms for multi-legged robots generally provide two different solutions for the left and right arm. In particular, for the back it will be

$$\dot{\mathbf{q}}_{bl} \neq \dot{\mathbf{q}}_{br}, \quad (11)$$

where $\dot{\mathbf{q}}_{br}$ and $\dot{\mathbf{q}}_{bl}$ are different solutions obtained considering the back belonging respectively to the right and to the left arm. However, generally this issue is commonly solved with the following choice for the joints of the back:

$$\dot{\mathbf{q}}_b = \frac{1}{2} (\dot{\mathbf{q}}_{br} + \dot{\mathbf{q}}_{bl}). \quad (12)$$

This guarantees a physical consistent solution, but in general none of the conflicting tasks will be actually achieved.

The CLIK algorithm based on the Augmented Jacobian cleverly resolves also this issue. Indeed, the vector of solution $\dot{\mathbf{q}}$ has been sorted in such a way that its elements appear just once, thus the inversion algorithm provides only one solution that is consistent with all the physical constraints.

On the other hand, the main problem related to the application of the Augmented Jacobian method is the matrix inversion, due to its dimensions ($\mathbf{J}_{AU}$ has 39 columns) and consequently to the detection of its singularities.

3 Simulations in VR

In order to test the proposed inversion model, several VR simulations have been carried out. First of all, a geometric model for the virtual humanoid has been built up with a typical hierarchical approach. The result has been a VRML model used for the simulations.

After that, the kinematic model of the virtual humanoid and its CLIK algorithm have been implemented with *MathWorks MATLAB–Simulink* software. Since a weighted pseudo-inverse has been adopted to compute the inverse kinematics, a proper choice of weights and of some optimization criteria have granted quite natural and fluid movements for the virtual humanoid. As a result, despite the ease of planning the movements of the digital humanoid, we can simulate quite complex tasks, by planning the trajectory for only a limited number of control points. For instance, the virtual humanoid can walk or even climb a ladder, as will be shown in the following sections.

3.1 *Standing-up from a sitting position*

In Figure 6 different frames of a standing-up simulation are shown. This task has been achieved just by imposing a null velocity to the feet of the virtual humanoid and by giving a vertical velocity reference to its pelvis. As a further constraint, the CoP must always belong to the support plane (*balance control*). As shown, the virtual humanoid performs the assigned movement always keeping itself in balance.

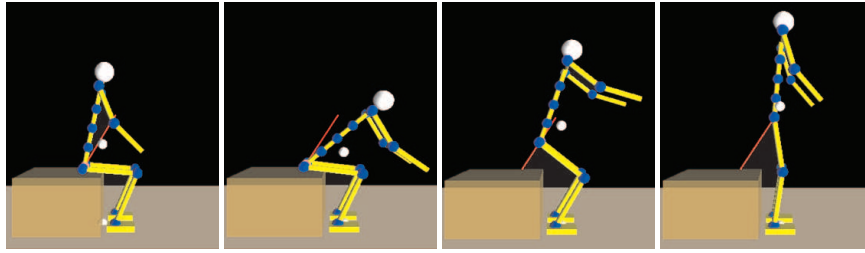


Fig. 6 Standing up from a sitting position

In a similar way, it is possible to simulate the virtual humanoid sitting down from a standing position.

3.2 *Collision avoidance*

The approach can be used to take into account also possible obstacles in the humanoid workspace. Figure 7 shows again the simulation of a standing up, but this time there is a table. This task has been achieved by assigning to the control points velocity references coming from repulsive potential fields.

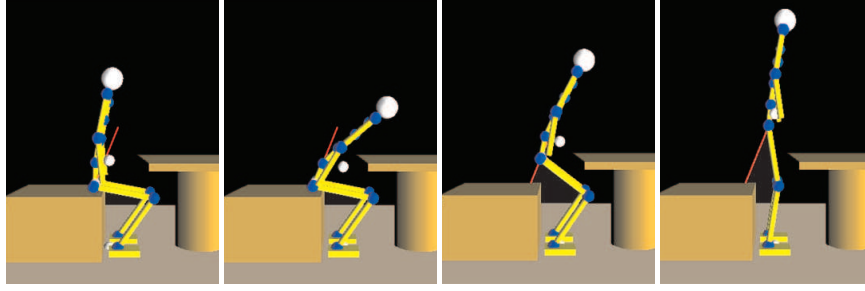


Fig. 7 Standing up from sitting position near a table

3.3 More complex tasks

In this section the results of the simulation of quite complex tasks are reported. The CLIK algorithm has always taken into account the constraints about the CoP, as mentioned above.

The results shown in the following figures can be quite interesting in the field of both computer graphics and robotics.

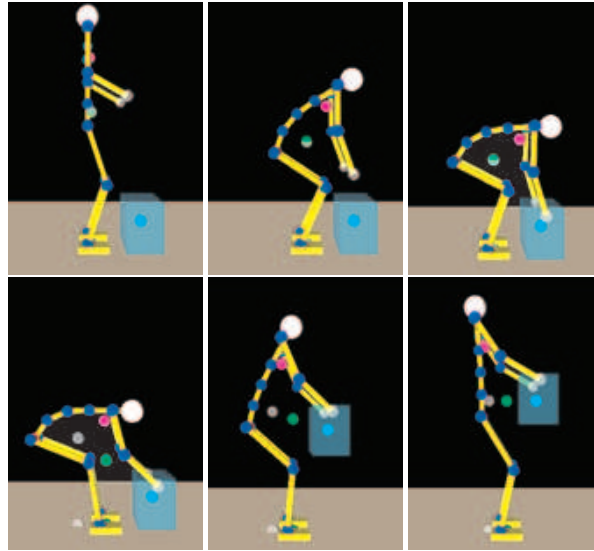


Fig. 8 The humanoid lifts a weight

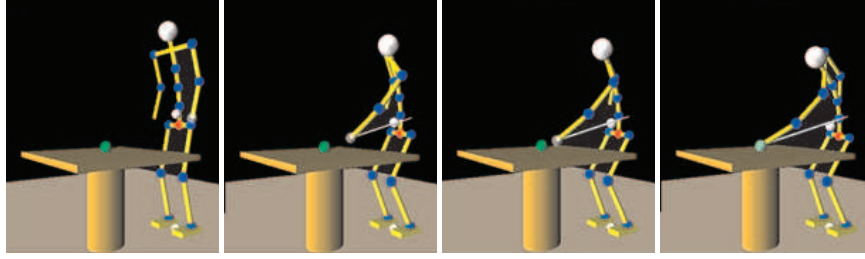


Fig. 9 The virtual humanoid grabbing an object on a table

4 Conclusions and future work

The main contribution of the proposed approach is in the computation of an Augmented Jacobian matrix to specify trajectories for different control points, including the control of the center of mass of the kinematic structure. In fact, the definition of the movements of the center of pressure as a primary task has granted natural movements to the virtual humanoid, in spite of the limited number of considered control points. Notice that these points can move on specified sections of the humanoid, giving the possibility of controlling nominally every point on the kinematic structure. The symbolic implementation leads to a very fast response of the algorithm with respect to complex simulations in the humanoid configuration space.

Moreover, the developed model lends itself to a very different set of applications, even not strictly robotic. Firstly, it can be used for digital animation of virtual humanoids in the field of ergonomics and process analysis. In fact, despite complexity and cost of already existing software tools dedicated to this type of analysis, generally their simulation algorithms are still tied to “key-frame” animation techniques. The developed model instead makes it possible to observe the resulting joint motion simply by planning the trajectory of a limited number of control points.

Another field of application could be marker-based motion capture [17], where the algorithm can be used in order to limit the number of markers needed to capture human movements.

Finally, although the described model is advanced in terms of quality of analysis, it is also computationally efficient. Specifically, a symbolic representation for the kinematics of the digital humanoid has been derived. In this way, it is possible to change in real-time several characteristic parameters of the chain, such as the applied loads, without further computational overload.

Acknowledgements The research leading to these results has been partially supported by the DEXMART Large-scale integrating project, which has received funding from the European Community’s Seventh Framework Programme (FP7/2007-2013) under grant agreement ICT-216239. The authors are solely responsible for its content. It does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of the information contained therein.

References

1. Sakagami, Y., Watanabe, R., Aoyama, C., Matsunaga, S., Higaki, N., Fujimura, K.: The intelligent ASIMO: system overview and integration. 2002 IEEE/RSJ International Conference on Intelligent Robots and System, Lausanne, CH (2002).
2. Kaneko, K., Kanehiro, F., Kajita, S., Yokoyama, K., Akachi, K., Kawasaki, T., Ota, S., Isozumi, T.: Design of prototype humanoid robotics platform for HRP. 2002 IEEE/RSJ International Conference on Intelligent Robots and System, Lausanne, CH (2002).
3. Iwata, H., Sugano, S.: Design of human symbiotic robot TWENDY-ONE. 2009 IEEE International Conference on Robotics and Automation, Kobe, J (2009).
4. Siciliano, B., Khatib, O. (Eds.): Springer Handbook of Robotics. Springer, Heidelberg, D (2008).
5. Robins, B., Dautenhahn, K., Boekhorst, T., Billard, A.: Robotic assistants in therapy and education of children with autism: Can a small humanoid robot help encourage social interaction skills? *Univ. Access Inf. Soc.*, **4**(2):105–120 (2005).
6. Di Gironimo, G., Lanzotti, A.: Designing in VR. *International Journal on Interactive Design and Manufacturing*, **3**(2):51–53 (2009).
7. Caputo, F., Di Gironimo, G., Marzano, A.: Ergonomic optimization of a manufacturing system work cell in a virtual environment. *Acta Polytechnica*, **46**(5):21–27 (2006).
8. Caputo, F., Di Gironimo, G., Papa, S.: A Virtual reality system for ergonomics and usability validation of equipment controls. *Anales de Ingeniería Gráfica*, **18**:47–64 (2006).
9. Di Gironimo, G., Patalano, S.: Re-design of a railway locomotive in virtual environment for ergonomic requirements. *International Journal on Interactive Design and Manufacturing*, **2**(1):47–57 (2008).
10. De Santis, A., Pierro, P., Siciliano B.: The virtual end-effectors approach for human-robot interaction. 10th International Symposium on Advances in Robot Kinematics, Ljubljana, SLO (2006).
11. De Santis, A., Albu-Schaeffer, A., Ott, C., Siciliano, B., Hirzinger, G.: The skeleton algorithm for self-collision avoidance of a humanoid manipulators. 2007 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Zürich, CH (2007).
12. De Santis, A., Siciliano B.: Inverse kinematics of robot manipulators with multiple moving control points. 11th International Symposium on Advances in Robot Kinematics, Batz-sur-Mer, F (2008).
13. Yamane, K.: *Simulating and Generating Motions of Human Figures*. Springer, Heidelberg, D (2004).
14. Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G.: *Robotics: Modelling, Planning and Control*. Springer, London, UK (2009).
15. Nakamura, Y.: *Advanced Robotics: Redundancy and Optimization*. Addison-Wesley, Reading, MA (1991).
16. Siciliano, B., Slotine, J.-J. E.: A general framework for managing multiple tasks in highly redundant robotic systems. 5th International Conference on Advanced Robotics, Pisa, I (1991).
17. Magnenat-Thalmann, N., Thalmann, D.: *Modelling and Motion Capture Techniques for Virtual Environments*. Springer Verlag, Geneva, CH (1998).