

A Goal driven perspective to generate humanoid motion synthesis

Joseph Salini, Vincent Padois, Aurélien Ibanez, Philippe Bidaud, Axel Buendia

▶ **To cite this version:**

Joseph Salini, Vincent Padois, Aurélien Ibanez, Philippe Bidaud, Axel Buendia. A Goal driven perspective to generate humanoid motion synthesis. Bidaud, P. and Tokhi, M. and Grand, C. and Virk, G. Field Robotics Proceedings of the 14th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, World Scientific, pp.889-897, 2011, 978-981-4374-27-9. 10.1142/9789814374286_0104 . hal-00720395

HAL Id: hal-00720395
https://hal.science/hal-00720395

Submitted on 24 Jul 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Goal Driven Perspective to Generate Complex Humanoid motion synthesis

Joseph Salini, Vincent Padois, Aurélien Ibanez, Philippe Bidaud

Université Pierre et Marie Curie,
Institut des Systèmes Intelligents et de Robotique - CNRS UMR 7222
Pyramide Tour 55, BC 173 — 4 Place Jussieu, F-75005 Paris
E-mail: {salini,padois,ibanez,bidaud}@isir.upmc.fr

Axel Buendia

SpirOps AI,
8 passage de la bonne graine, F-75011 Paris
CNAM CEDRIC,
292 rue St Martin, F-75003 Paris
E-mail: axel.buendia@spirops.com

Keywords: Humanoid Control; Dynamic; Linear Quadratic Problem; Goal-Driven Perspective.

1. Introduction

In the paper of Salini *et al.*,[1] the authors proposed a generic controller which performs the synthesis of complex humanoid whole-body behaviors. The core of this low-level controller is based on a Linear Quadratic Program (LQP) where the minimum of a quadratic cost function is computed, subject to linear constraints. The cost function represents a weighted sum of tasks to be achieved by the robot and the constraints the physical limitations of the system. Similar controllers are used in the works of Abe *et al.*[2] and Colette *et al.*[3]

Although multi-criteria (weights based) controllers cannot give a meaningful physical interpretation of the criterion to optimize, the approach is competitive in terms of tasks accomplishment compared to hierarchies based methods (see Kanoun *et al.*,[4] and Escande *et al.*[5]). It is computationally more efficient and offers greater flexibility by allowing complex task blending. However, the choice of the weights/hierarchies is left to the programmer, which can limit the missions/goals achievement.

This issue can be bypassed with a higher level controller which reactively elects tasks of interest based on contextual information. These tasks and their associated relative weights can then be accounted for at a lower, physics related, control level.

A recent work in this domain is proposed in Philippsen *et al.*[7] which introduces an architecture combining a Whole Body Control (low) level and a Reactive Symbolic Planning (high) level through what is called a Mobile Manipulation Database and whose goal is to deal with information which is relevant to all components of the system.

The contribution of the proposed work has to be interpreted from an "action selection" perspective, where no explicit plan is constructed, and where a mission is described in terms of high level goals, associated to contextual information (percepts) and rules, which allow the triggering of high level actions — blends of lower level tasks. Hence it illustrates the use of such a high level controller in combination with a powerful task-level controller where complex whole-body humanoid behaviors cannot be planned in a deterministic manner.

2. Method

2.1. *Task level control*

2.1.1. *The Constraints*

The task-level controller has been extensively described in the paper of Salini *et al.*[1] The robot is subject to many constraints, such as the equation of motion

$$M(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = J_\chi(\boldsymbol{q})^T \boldsymbol{\chi} + \boldsymbol{g}(\boldsymbol{q}). \tag{1}$$

where $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$ are respectively the generalized coordinates, velocity and acceleration vectors, $M, \boldsymbol{n}, \boldsymbol{g}$ the generalized inertia matrix, the non-linear effects vector (Coriolis and centrifugal) and the gravity vector. $J_\chi(\boldsymbol{q})^T = \left[J_{c,1}(\boldsymbol{q})^T \dots J_{c,n_c}(\boldsymbol{q})^T \; S^T \right]$ is the generalized external wrench jacobian composed of the jacobians of contacts $J_{c,i}(\boldsymbol{q})$ and of the actuation matrix S. $\boldsymbol{\chi} = \left[\boldsymbol{w}_{c,1}^T \dots \boldsymbol{w}_{c,n_c}^T \; \boldsymbol{\tau}^T \right]^T$ is called the **action variable** of the system, with $\boldsymbol{w}_{c,i}$ the external wrenches of contact and $\boldsymbol{\tau}$ the input torque vector.

The robot has also several limitations, which can be interpreted as internal or external constraints. The internal ones are typically the joint actuation limits, as the torque (Eq. 2) and its derivative (Eq. 3), the position (Eq. 4), the velocity (Eq. 5) and the acceleration (Eq. 6), expressed as follows:

$$\boldsymbol{\tau}_{min} \leq \tau \leq \boldsymbol{\tau}_{max} \tag{2}$$

$$\dot{\boldsymbol{\tau}}_{min} \leq (\tau - \tau_{prec})/dt \leq \dot{\boldsymbol{\tau}}_{max} \tag{3}$$

$$\boldsymbol{q}_{min} \leq \boldsymbol{q} + h_2\dot{\boldsymbol{q}} + \frac{(h_2)^2}{2}\ddot{\boldsymbol{q}} \leq \boldsymbol{q}_{max}, \tag{4}$$

$$\dot{\boldsymbol{q}}_{min} \leq \dot{\boldsymbol{q}} + h_1\ddot{\boldsymbol{q}} \leq \dot{\boldsymbol{q}}_{max} \tag{5}$$

$$\ddot{\boldsymbol{q}}_{min} \leq \ddot{\boldsymbol{q}} \leq \ddot{\boldsymbol{q}}_{max} \tag{6}$$

where τ_{prec} is the previous torque vector and $h_1,\ h_2$ are anticipation coefficients set to predict the future value of the state $(\boldsymbol{q}, \dot{\boldsymbol{q}})$ given the generalized acceleration $\ddot{\boldsymbol{q}}$. The external constraints typically describe the interactions of the robot with its environment. For example frictional contacts are characterized by two constraints, one which equals the acceleration at contact point to 0 (Eq. 7) and the other which bounds the force into the linearized friction cone $C_{f,i}$ (Eq. 8).

$$J_{c,i}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{J}_{c,i}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \mathbf{0} \tag{7}$$

$$C_{f,i}\boldsymbol{w}_{c,i} \leq \mathbf{0} \tag{8}$$

Kinematic loops, which are also considered as external constraints, link two frames of the tree-structure system, and result in the following equation $\boldsymbol{t}_i - \boldsymbol{t}_j = \mathbf{0}$ where $\boldsymbol{t}_i$ and $\boldsymbol{t}_j$ are respectively the twists of frames i and j. Expressed in terms of task space acceleration, this equation becomes

$$\left(J_i(\boldsymbol{q}) - J_j(\boldsymbol{q})\right)\ddot{\boldsymbol{q}} + \left(\dot{J}_i\left(\boldsymbol{q}, \dot{\boldsymbol{q}}\right) - \dot{J}_j\left(\boldsymbol{q}, \dot{\boldsymbol{q}}\right)\right)\dot{\boldsymbol{q}} = \mathbf{0}. \tag{9}$$

2.1.2. *The Tasks*

Given the **variable** of the system $\boldsymbol{\chi}' = \left[\ \ddot{\boldsymbol{q}}^T\ \boldsymbol{\chi}^T\ \right]^T$, a task can be defined in a general manner as the minimization

$$\min_{(\boldsymbol{\chi}')}\ \ T_j(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\chi}') = \left\| E_j(\boldsymbol{q})\boldsymbol{\chi}' + \boldsymbol{f}_j(\boldsymbol{q}, \dot{\boldsymbol{q}}) \right\|. \tag{10}$$

In the case of a desired task space acceleration $\dot{\boldsymbol{t}}_j^{des}$, the task is written $\left\| J_j(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{J}_j(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \dot{\boldsymbol{t}}_j^{des} \right\|$. With D_a defined such that $\ddot{\boldsymbol{q}} = D_a\boldsymbol{\chi}'$, the formulation of a task space acceleration leads to $E_j(\boldsymbol{q}) = J_j(\boldsymbol{q})D_a$ and $\boldsymbol{f}_j(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \dot{J}_j(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \dot{\boldsymbol{t}}_j^{des}$.

In the case of a desired task space wrench $\boldsymbol{w}_j^{des}$, the task is written $\left\| \boldsymbol{w}_j - \boldsymbol{w}_j^{des} \right\|$. With $D_{w,j}$ defined such that $\boldsymbol{w}_j = D_{w,j}\boldsymbol{\chi}'$, the formulation of a task space wrench leads to $E_j(\boldsymbol{q}) = D_{w,j}$ and $\boldsymbol{f}_j(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{w}_j^{des}$.

These generic tasks allow to control many parts of the robot, as frames and contact forces, but also its joints in acceleration and torque, its center of mass (CoM), etc.

The desired values of the tasks $\dot{\boldsymbol{t}}_j^{des}$ and $\boldsymbol{w}_j^{des}$ are computed separately with any controller of the literature, for example a simple proportional derivative controller $\dot{\boldsymbol{t}}^{des} = \dot{\boldsymbol{t}}^{goal} + K_p\boldsymbol{\epsilon}_p + K_d$ where K_p, K_d are respectively the coefficient of stiffness and damping, and more complex one like a preview controller to control the CoM of the robot for walking tasks.[10]

$$\dot{\boldsymbol{\epsilon}}_p \tag{11}$$

2.1.3. *The Resolution: Weighting and Hierarchy*

Two strategies exist to solve a multi-tasks problem, the weighting or the hierarchy. Given a set of n_t tasks, in the first case each task is related to a weight, $(T_i(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\chi}'), \omega_i)$ $i \in [1..n_t]$, and the minimization problem is defined as:

$$\begin{aligned} &\min_{(\boldsymbol{\chi})} \ \tfrac{1}{2}\left(\textstyle\sum_{i=1}^{n_t}((\omega_i.T_i(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\chi}'))^2) + (\omega_0.T_0(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\chi}'))^2\right) \\ &s.t.: \qquad\qquad G\boldsymbol{\chi} \preceq \boldsymbol{h} \\ &\qquad\qquad\quad A\boldsymbol{\chi} = \boldsymbol{b} \end{aligned}$$

where $\omega_0 \ll \omega_i \ \forall i$ and G, A, $\boldsymbol{h}$ and $\boldsymbol{b}$ are the concatenated representations of the constraints.

In the second case, each task is related to a level, $(T_i(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\chi}'), i)$ $i \in [1..n_t]$, and the minimization problem is solved recursively as following:

`for i=1..`n_t`:`

$$\begin{aligned} &\min_{(\boldsymbol{\chi})} \ \frac{1}{2}\left((T_i(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\chi}'))^2 + (w_0.T_0(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\chi}'))^2\right) \\ &s.t.: \ G\boldsymbol{\chi} \preceq \boldsymbol{h} \\ &\qquad A_i\boldsymbol{\chi} = \boldsymbol{b}_i \\ \\ &\qquad A_{i+1} \leftarrow A_i \cup E_i \\ &\qquad \boldsymbol{b}_{i+1} \leftarrow \boldsymbol{b}_i \cup (E_i\boldsymbol{\chi}_i^*) \end{aligned}$$

As the general purpose is to synthesize the humanoid motion, transitions may occurs in the set of importances and constraints. These transitions may involve sharp evolutions of the actuation vector, so in order to avoid this

problem the authors prefer to use the weighing strategy, because weights "(floating values) can be modified in a continuous way, whereas in a hierarchy strategy levels (discrete values) change in a discontinuous way.

2.2. *Goal driven control*

To blend the different tasks and perform good behavior synthesis, one needs to change the tasks weights efficiently. This can be done manually, as shown in the paper of Salini *et al.*[1] for simple missions, but it has its limitation when unknown dynamic external events occur. The robot must be controlled with goals in a high-level description language as "grab the red box" or "go to this location", but also with the contextual information which describes the robot perceptions, i.e. "box is there", "location is reachable", "goal has been performed", etc.

The lowest level includes a set of n_t **tasks** T "(defined above) that the robot can perform. It is composed of posture tasks, i.e. sitting, crouching, and standing, and frame tasks, i.e. hands, feet, head, pelvis, bottom and CoM. The blending of all these tasks is done by the LQP controller and the weighting strategy.

The intermediate level includes a set of n_a **actions** $\mathbb{A}$ which defines predefined tasks combination using **blending factors** β_{ij} between the action $\mathbb{A}_j$ and the task T_i.

Finally, at the highest level, a set of n_g **goals** G is defined in accordance to the objective of the mission. Each goal G_k refers to an action $\mathbb{A}_j$ and compute its **Desire** of completion $\delta_{jk} = \alpha_{jk}.\theta_{jk}$ according to its **adequacy** $\alpha_{jk} \in [0, 1]$ ("I want"), and its **opportunity** $\theta_{jk} \in [0, 1]$ ("I can"). Notice that one action can be related to many goals, for example "running" related to "attack" and ""flee".

The decision making engine selects for each action the goal with the highest desire δ_j^*, which leads to the weight ω_i of task T_i

$$\begin{aligned} \delta_j^* &= \max_k(\delta_{jk}) \\ \omega_i &= \sum_{j=1}^{n_a} \beta_{ij}\delta_j^*. \end{aligned} \tag{12}$$

This high level goal driven controller is performed with Spirops,[11] which provides an open environment to easily and incrementally design autonomous decision processes, thanks to a graphical editor which edits independent goals. Each of them can be seen as extended 0+ fuzzy rules,[12] and the decision making engine computes the interest and the parameters

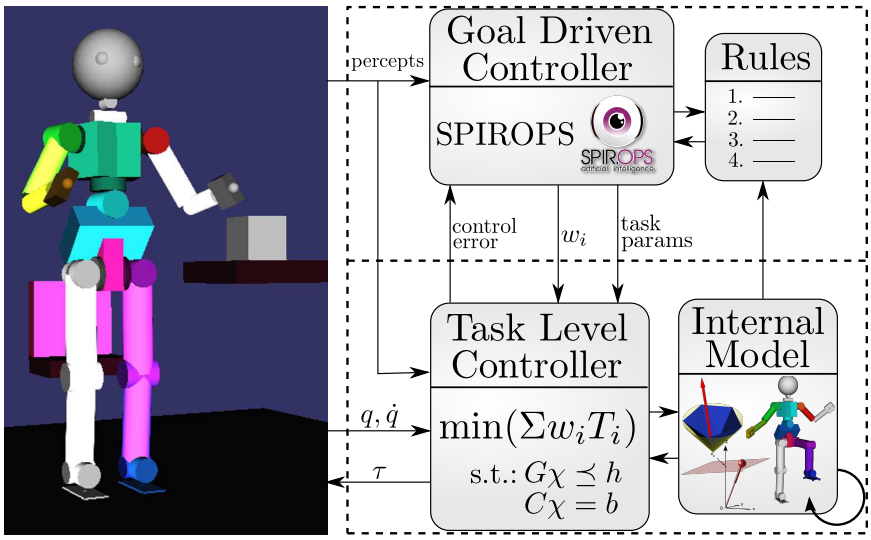


Fig. 1. Overall control architecture

for each action to produce. The general architecture of the whole controller is shown in Fig. 1.

3. Results

This architecture has been used in several simulated experimental results where a humanoid robot iCub moves in an unknown environment *a priori* and has to perform some high level goals, for example "to grab a box" and "to drop it".

3.1. *A changing World*

The first experiment shows the iCub robot in some situations with different initial conditions. It may sit or stand, and its objective is to grab a box which can be in its range or not. The situation is evaluated by the high level decision making engine which decides what set of actions it has to perform, and leads to the control of low level tasks. In Fig. 2, the pictures on the left side show that the scenario starts with 2 different initial conditions, And the robot successfully grabs the box. On the right of the figure, the graph shows the evolution of the tasks weights according to its perceptions.

In the second experiment, the robot opens a door many times in the same sequence. It pushes the door at the maximum, and when the task

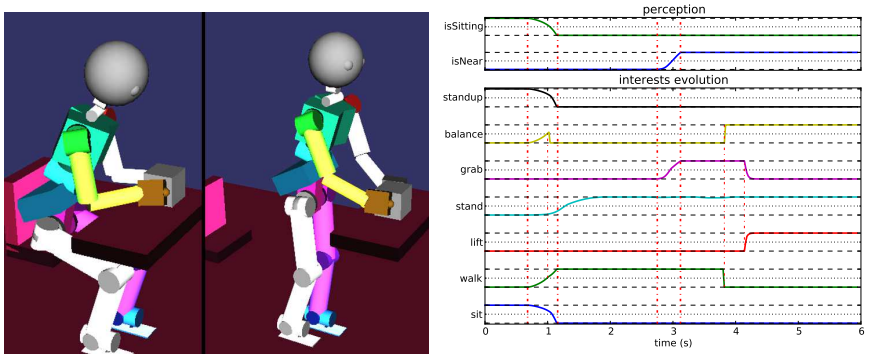


Fig. 2. Experiment 1 – Left: Two scenarios with different initial configurations and one objective, to grab the box – Right: Evolution of the actions interests according to perceptions. Bold dashed red lines show the correlation with perception, the thin ones highlight the relations between actions.

cannot be performed correctly, the robot adapts its position and posture, uses the closest hand to grab the knob and pushes the door again. The Fig. 3 illustrates this experiment.

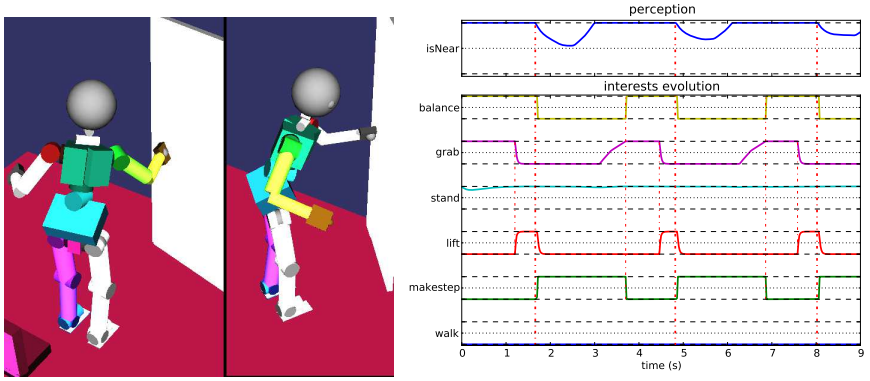


Fig. 3. Experiment 2 – Left: The robot opens a door many times. It uses its left or right hand, depending on the distance between the hands and the knob. – Right: Evolution of the actions interests according to perceptions. Bold dashed red lines show the correlation with perception, the thin ones highlight the relations between actions.

3.2. *Dynamic Adaptation*

The last experiment describes the adaptation of the robot subject to dynamic disturbances. The objective is to displace a box from a location to another. This box can be light or heavy. During the displacement, the center of mass of the system "robot + box" get closer to the base of the support defined by the convex hull of the feet, and if it reaches a threshold, the robot drops the box and moves to a closer location to achieve its objective. Fig. 4 gives the configuration of the robot displacing a heavy box. On the right of the figure, the evolution of the center of mass, the box and the action interests are plotted. Notice the peaks in the evolution of the center of mass, due to the actions of grabbing and dropping.

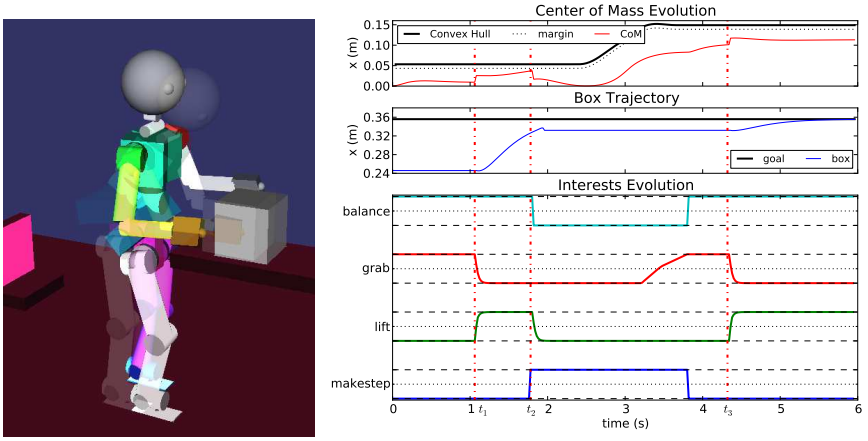


Fig. 4. Experiment 3 – Left: The robot has to displace a $2Kg$ box from one side of the table to another – Right: Evolution of the position of the center of mass and the box, and the interest of the actions. The robot needs to drop the box once and make a step to achieve its goal.

4. Conclusion

This paper shows the combination of a low level controller based on weighted tasks mixed with a LQP, and a high level controller based on goals and actions. This association is useful for non deterministic scenarios, and it is illustrated trough three simulations.

In the future, the authors will focus on the consistence of the tasks to get a physically meaningful interpretation of the weighted sum of tasks in

the LQP. Furthermore, a new formalism of the problem will be explored to reduce the computational time of the control loop.

References

1. J. Salini, V. Padois and P. Bidaud, Synthesis of complex humanoid whole-body behavior: a focus on sequencing and tasks transitions, in *Proc. of the IEEE ICRA*, (Shanghai, China, 2011).
2. Y. Abe, M. da Silva and J. Popovic, Multiobjective control with frictional contacts, in *Proc. of the Symposium on Computer Animation*, 2007.
3. C. Collette, A. Micaelli, P. Lemerle and C. Andriot, Robust balance optimization control of humanoid robots with multiple non coplanar grasps and frictional contacts, in *Proc. of the IEEE ICRA*, (Pasadena, USA, 2008).
4. O. Kanoun, F. Lamiraux, P.-B. Wieber, F. Kanehiro, E. Yoshida and J.-P. Laumond, Prioritizing linear equality and inequality systems: Application to local motion planning for redundant robots, in *Proc. of the IEEE ICRA*, (Kobe, Japan, 2009).
5. A. Escande, N. Mansard and P.-B. Wieber, Fast resolution of hierarchized inverse kinematics with inequality constraints, in *Proc. of the IEEE ICRA*, (Anchorage, USA, 2010).
6. W. Li, Optimal control for biological movement systems, PhD thesis, University of California, (San Diego, USA, 2006).
7. R. Philippsen, N. Nejati and L. Sentis, Bridging the gap between semantic planning and continuous control for mobile manipulation using a graph-based world representation, in *1st Int. Work. on Hybrid Control of Autonomous Systems held in conj. with the Int. Joint Conf. on AI*, (Pasadena, USA, 2009).
8. E. Yoshida, I. Belousov, C. Esteves and J.-P. Laumond, Humanoid motion planning for dynamic tasks, in *Proc. of the IEEE/RAS Int. Conf. on Humanoid Robots*, (Tsukuba, Japan, 2005).
9. S. Cambon, R. Alami and F. Gravot, *The Int. Jour. of Robotics Research* **28**, 104(January 2009).
10. S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi and H. Hirukawa, Biped walking pattern generation by using preview control of zero-moment point, in *Proc. of the IEEE ICRA*, (Taipei, Taiwan, 2003).
11. A. Buendia, Spirops AI, a behavior production tool.
12. P. D. Magnus, *forallx An Introduction to Formal Logic* (University at Albany, State University of New York, 2005).