The challenge of motion planning for humanoid robots playing soccer

Stefano Carpin
School of Engineering and Science
International University Bremen
Germany
carpin@ieee.org

Enrico Pagello
Department of Information Engineering
University of Padova
Italy
epv@dei.unipd.it

Abstract— **Motion planning for humanoids suffers from the high dimensionality of the configuration space. Moreover, the need to satisfy dynamic and static constraints increases the overall difficulty. While the above challenges hold for any humanoid robot, the soccer scenario adds difficulties narrowly addressed in humanoids motion planning research. Dynamic environments with active opponents, the requirement to perform short and long term plans to perform soccer relevant actions, and the necessity to plan also movements purposely terminating with a collision with the ball open a completely new scene for researchers. This paper surveys state of the art research in motion planning for humanoids robots playing soccer, outlining connections, differences, and identifying the key aspects that ought to be addressed in order to develop effective soccer players.**

I. Introduction

The essence of the Robocup vision can be paraphrased as the development of humanoid robots with human-like physical and cognitive capabilities. The chosen deadline is the year 2050. Soccer playing requires team strategy, online learning, and real time sensor processing, to say the least. On top of that, by its very own nature, soccer involves strong physical abilities. If the ambitious vision is to ever be achieved, humanoid robots with skills far beyond any currently used robot need to be developed. While to the outsider these goals may seem unreachable in the given time frame, those who witnessed the tremendous developments in first ten years of Robocup competitions have probably different expectations. Starting from a stage where teams composed by differential drive robots played almost static games in a wall surrounded arena, omnidirectional vehicles meanwhile participate in highly competitive games on playing fields that resemble more and more a real soccer pitch. In parallel, competitions with humanoid robots have been started. However, even the novice shall note that while certain technologies or algorithms can be seamlessly moved or adapted from wheeled robots to humanoids, mobility requires to enter a completely different realm.

This paper focusses on this latter aspect. We will provide a survey of some results available for humanoid motion planning, with a specific emphasis on problems, or advantages, distinguishing the *soccer* scenario. The following aspects are dominant for soccer playing humanoids:

- Soccer is a fast game. It follows that motion plans have to strive for the generation of speedy gaits. Methods exploiting the assumption of slowly evolving statically stable postures are doomed to be on the losing side.
- Soccer is played on a plain field where the only obstacles are other robots (opposing or friends), the ball, and the goals. Elaborate approaches considering whether it is more rewarding to overcome or pass an obstacle on the side are therefore almost pointless in this scenario.
- Soccer specific tasks, like defending from an adversarial shot may require a motion purposefully terminating with a fall on the ground. Robots acting as goalies need to quickly recover from these situations in order to continue their games. The majority of humanoids motion planning methods ignore the problem of getting back on two feet.
- Robots need to kick the ball. Most of motion planning research strives for the generation of motions where no impulsive interaction between the robot and the the environment occurs. Kicking a ball, obviously violates this assumption.

The paper is organized as follows. Section II formally defines the motion planning problem, and it furthermore details about the notable concepts of static and dynamic

Proceedings of the Workshop on Humanoid Soccer Robots
of the 2006 IEEE-RAS International Conference on Humanoid Robots
Genoa, Italy, December 4, 2006
pp. 71-77
ISBN: 88-900426-2-1

balance. Next, section III outlines some interesting approaches to humanoid motion planning. Section IV surveys currently used methods in the Robocup competition. Finally, conclusions are offered in section V.

II. Motion planning for humanoids: problem formulation

Early results on robot motion planning outlined that the problem suffers from the so-called *curse of dimensionality*, i.e. under the widely accepted conjecture that $P \neq NP$, time complexity is exponential in the number of degrees of freedom (d.o.f.) [1][2]. Even if robots participating in Robocup competitions are often simpler than sophisticated robots like the famous Qrio or Asimo, their number of d.o.f. is nevertheless high. As frame of reference, during the last Robocup competition (Bremen 2006), the number of d.o.f. for the robots involved in the Humanoid kid size league varied from 17 [3] to 28 [4]. Figure 1 illustrates a typical setup for the joints controlling the legs of a humanoid robot, with 6 d.o.f. per leg.

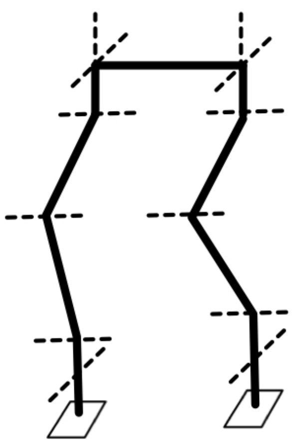

Fig. 1. A typical mechanical setup for the legs of a humanoid robot used in the Robocup competition. Six degrees of freedom are used to move each leg: three in the hip, one on the knee and two for the ankle (dashed lines in the figure show the rotational axis of the various joints).

The elevated number of d.o.f. calls for the use of randomized algorithms [5], since combinatorial techniques are too slow. However, the morphology of humanoid robots, specifically the fact that they are equipped with two legs, poses additional challenges to the already arduous motion planning problem. We shortly remind that in its basic form the problem can be formulated as follows. Let $\mathcal{C}$ be the configuration space induced by the set of d.o.f., and let $\mathcal{C}_{free}$ and $\mathcal{C}_{obs}$ be a partition of $\mathcal{C}$ into the spaces of free and obstacle configurations. Given $q_s, q_g \in \mathcal{C}_{free}$, determine a continuous function $f : [0, T] \rightarrow \mathcal{C}_{free}$ such that $q_s = f(0)$ and $q_g = f(T)$. A distinguishing aspect of humanoid motion planning is the fact that when the robot stands on two feet it forms closed chain with the ground, while when it balances on a single foot it does not form a chain anymore (but rather a tree). From a theoretical point of view this means that different manifolds characterize these two situations. In traditional motion planning, configurations in $\mathcal{C}_{obs}$ arise due to the collision between the robot and some obstacle. Humanoid robots have to move according to plans that not only avoid collisions, but keep away also from *unstable* configurations. More precisely, two types of stability have to be considered.

A. Static stability

Static stability has to be considered when no torque[1] is provided by robot's actuators. According to well known principles coming from mechanics, equilibrium is obtained when the sum of acting forces and moments is zero. In case of static stability, the only acting force is gravity. It is well known that a robot posture is statically stable if the projection on the ground of the position of the center of mass falls inside the convex hull of the supporting points. Figure 2 shows a (simulated) humanoid in a statically stable position.

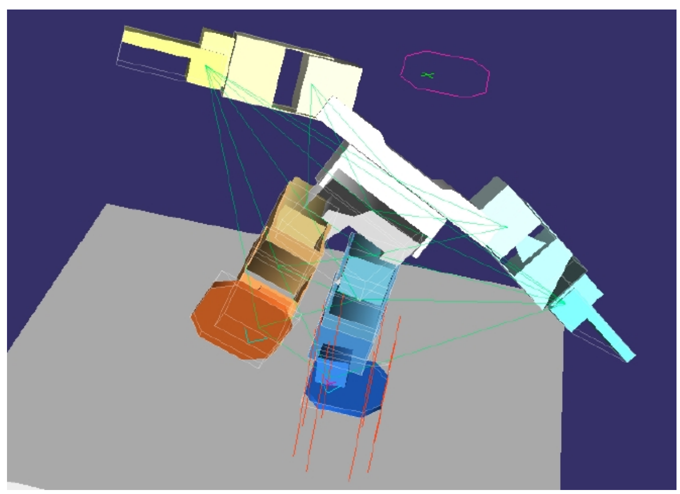

Fig. 2. A statically stable position for a simulated humanoid robot

The necessity to avoid configurations that are not statically stable introduces further constraints on the

[1]it is assumed that all actuators are associated with revolute joints, therefore they provide torques rather than forces. This assumption is consistent with state of the art humanoid platforms used in Robocup

space of configurations to be searched by the motion planner. More specifically, according to the notation used by Kuffner et al. [6], static stable paths are searched in the set $\mathcal{C}_{stable} \subset \mathcal{C}$, where $\mathcal{C}_{stable}$ is the set of statically stable configurations.

B. *Dynamic stability*

When robot actuators deliver torques, dynamic stability instead of static stability has to be considered. This concept builds upon the concept of *zero moment point* (ZMP), introduced by Vukobratović more more than thirtyfive years ago (see [7] for a recent synopsis of this concept and subsequent developments). Dynamic balance is particularly relevant during the stage of *single support*, i.e. when the robot stands on a single foot. As outlined by Vukobratović [7], ZMP can be defined, or interpreted, in different ways. First, it can be defined as *that point on the ground at which the net moment of the inertial forces and the gravity forces has no components along the horizontal axes*. When ZMP is outside the support polygon (i.e. the convex hull of the points in contact with the ground), the robot tilts over by rotating around one edge of the supporting feet[2]. The ZMP condition is necessary and sufficient for dynamic stability. An alternative, but equivalent, interpretation was given by Arakawa and Fukuda [8]. They define ZMP as the point p where $T_x = 0$ and $T_z = 0$ where T_x and T_z represent the moments around the x and y axis generated by the reaction force R and reaction torque M, respectively. They then state that *when p exists within the domain of the support surface, the contact between the ground and the support is stable*. Figure 3 illustrates this interpretation of ZMP.

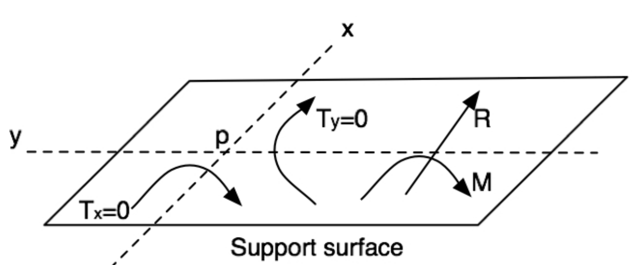

Fig. 3. A graphical interpretation of the ZMP criterion. If the point p is inside the support surface, dynamic balance is obtained.

[2]to be more precise, ZMP definition is sound only when it falls inside the support polygon. So, technically, ZMP *outside the support polygon* means ZMP undefined.

Figure 3 also hints that the closer the ZMP (point p) to the center of the support surface, the better. A ZMP close to the boundary in fact implies very little robustness to the unavoidable errors arising while moving.

III. Motion planning for humanoids: solving approaches

In this section we discuss different solutions for the motion planning problem applied to humanoid robots. They differ for the underlying hypothesis, or for the solution philosophy. In general, two major strategies are followed. The first consists in generating motion plans online, i.e. to repeatedly join two given configurations. Although more general and appealing, this method is usually slow. The second builds upon the observation that *walking* is a periodic activity, and that complex gaits can be decomposed into simpler motion primitives that are precomputed offline (either by a program or by a human operator). Motion planning in this case mease searching a sequence of primitives that achieve the given task. This approach can be more efficient, but is less general and typically ignores obstacles or dynamic situations.

A. *RRT based motion planning*

An interesting solution belonging to the first category was proposed by Kuffner et al. [6]. Their approach builds upon the Rapidly-Exploring Random Trees (RRT) algorithm [9][10], and reportedly has been the first successful attempt to provide a general motion planner for humanoid robots whose performance was confirmed on a physical robot. As every RRT based method, geometrical representations of the environment and of the robot are needed. The algorithm computes a trajectory between two given configurations that are both static stable. The authors introduce an additional subset of $\mathcal{C}_{free}$, namely $\mathcal{C}_{valid} = \mathcal{C}_{free} \cap \mathcal{C}_{stable}$. This is needed because a static stable configuration is not necessarily collision free, and the path has to satisfy both constraints. Algorithms 1 and 2 show the pseudocode for the RRT algorithm adapted to search paths in $\mathcal{C}_{valid}$

A critical point for this algorithm is the necessity to determine the *distance* between two configurations (in order to find out the *nearest neighbor*). Although any metric in the form

$$\rho(q, q') = \sum_{i=1}^{n} c_i ||q_i - q_i'||$$

with $c_i \geq 0$ is valid, the choice of the coefficients c_i has great impact. The distance between two configurations should indicate the *cost-to-go* or the effort to move the

```
BUILD_RRT(q_init, K)
INPUT a starting configuration q_init and the number of iterations K
OUTPUT a RRT τ
τ.init(q_init)
for k = 1 to K do
    q_rand ← RANDOM_CONFIG
    EXTEND(τ, q_rand)
RETURN τ
```

Algorithm 1: Construction of the RRT

```
EXTEND(τ, q)
INPUT a tree τ and a random configuration q
RETURN Trapped or Reached or Advanced
q_near ← NEAREST_NEIGHBOR(q, τ)
if NEW_CONFIG(q, q_near, q_new) then
    τ.add_vertex(q_new)
    τ.add_edge(q_near, q_new)
    if q_new = q then
        RETURN Reached
    else
        RETURN Advanced
RETURN Trapped
```

Algorithm 2: EXTENSION of the RRT

humanoid from q to q'. At the moment no general criteria are known, and iterated heuristic attempts are rather used. As acknowledged by the authors, this approach still needs improvements in order to become a viable alternative for robot soccer. Computation times ranging from 30 to 600 seconds (on state of the art machines in 2002) also indicate that special care has to be taken when applying it. A way to reduce the time is to decrease the dimension of the search space by omitting, for example, the torso and the arms. This strategy needs however to include the inertial effects of the excluded parts.

B. Walking patterns

An alternative to online motion planning is the generation of so-called *offline patterns*, i.e. walking primitives that can be used as building blocks in order to elaborate complex moves. The method proposed by Huang et al. [11] has the desirable property of generating gaits such that the hip motion is optimized to keep the ZMP in the center of the support region. This approach aims to maximize the *gait robustness*, because errors arising during the execution will be harmless as long as the ZMP stays inside the convex region. The method analyzes the periodic nature of walking and striving for motion speed it relaxes some cumbersome assumption, like the constraint that the foot stays always parallel to the soil, even while advancing. The mathematical model, not complicated but too long to be reported here, first computes the foot trajectory using a third-order spline interpolation. Next, the hip trajectory is also computed using a third-order spline. The use of third-order splines implies the desirable fact that second order derivatives are continuous, and then amenable of being appropriately tracked by the servos.

C. Gait optimization

Learning optimized gaits, a form of offline walking patters, has been investigated by Hu et al. in [12]. They distinguish two phases in the gait pattern: a *swing phase*, when the robot stands on a single foot, and *double support phase*, when both feet are on the ground. Constraints and performance criteria are then introduced in order to cast the gait search as a constrained optimization problem. Constraints are due to the geometry of the robot, limits on the forces and velocities, and dynamic stability. The performance (cost) function to be minimized is the sum of two terms, one measuring energy consumption, and the other the ZMP displacement. The evolutionary distributed algorithm (EDA) is used to search the minimum. EDA is similar to genetic algorithm (GA) but instead of generating new solutions through mutation and recombination, a sampling process over the set of promising solutions is used. In practice, after having generated a set of initial solutions, new ones emerge by altering the best ones according to a multivariate Gaussian distribution. The authors show experimental results supporting the goodness of the EDA optimization step, that therefore appears to be an interesting possibility to generate offline optimized walking gaits. These gaits will then be selected to compose more complicated sequences.

D. Omnidirectional walking

The introduction of omnidirectional wheeled platforms has been a great breakthrough in the Robocup middle and small size leagues. Starting from this standpoint Behnke has developed a procedure able to generate omnidirectional walking for bipedal robots [13]. Notably, this approach was introduced having the Robocup competition in mind and was validated on the physical robots used during Robocup competitions [14]. The algorithm accepts as input a vector consisting of three components, v_x,v_y and v_θ, that represent the desired lateral, forward and rotational speeds, respectively. The motion is broke down in the phases *shifting*, *shortening* and *moving*. In each stage simple trigonometric calculations yield the desired values for the 6 degrees of freedom controlling each leg. A valuable aspect of this research is indeed its simplicity, so that it can be implemented on pocket PC like devices. However, also according to the authors, the gait generation is optimized for stability and generality, rather than for speed. Further investigations are also needed in order to take obstacles into account while planning these omnidirectional moves – an aspect neglected at the moment.

E. Getting up again

Falling down is typically seen as a catastrophic event in humanoid robotics. In humanoid soccer, instead, certain tasks, like defending from an adversarial shot, hinge on the ability to promptly dive in the appropriate direction. As a consequence, the ability to quickly recover and get up again is also needed, because the game does not stop. Stückler et al. investigated this specific task, an aspect usually ignored in related literature [15]. Most teams are able to identify the event *tipped over* robot thanks to the onboard orientation sensors. The technique presented in [15], starts from the assumption that the robot can lie on the ground only in the supine or prone posture, due to its specific mechanical arrangement. The authors propose two distinct preprogrammed standing up procedures. The procedure to apply is easily determined upon inspection of the sensor indicating the sagittal tilt. In both cases a sequence of four motions manages to bring the robot back to the upright posture. The effectiveness of the two approaches was empirically verified during the Robocup competition.

F. Kicking the ball

Kicking the ball is unique feature of Robot soccer (see figure 4). In fact in main stream humanoid research, contacts between the robot and the environment, besides walking, happen typically only for grasping. There appears to be few results available about *planning a good kick*.

Many teams define a *kicking behavior* that is often programmed offline and then scheduled during the game when needed. An aspect that at the moment appears completely ignored is the impulsive collision with the ball. This is justified by the fact that the ball currently weights only a few grams and therefore the impact effects can be safely disregarded. However, on the way to develop robots that play soccer in conditions more and more resembling human soccer, this aspect needs to be addressed. A real soccer ball weights between 400 and 450 grams, and can reach remarkable speeds.

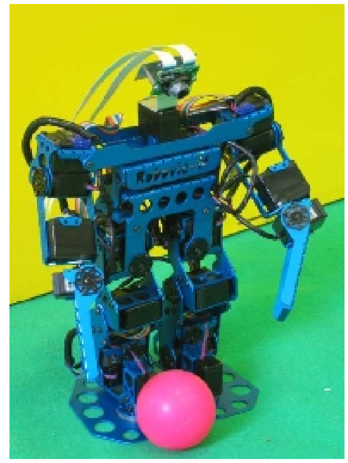

Fig. 4. A robot approaching the ball in order to kick it to the goal.

IV. CURRENT STATE OF THE ART IN ROBOCUP

The last Robocup competition attracted 20 participant teams. Each of them exhibited very different characteristics with respect to the different components needed to defeat opponents. This variety is evident also in the different ways the motion planning problem has been attacked.

Online trajectory generation based on RRT, in a way similar to the algorithm formerly illustrated was adopted by the Artisti team [16]. Another randomized approach, although different, has been used by the Robo-Erectus team in order to learn walking gaits offline [17]. This team uses the EDA technique formerly described. The winning team, TeamOSAKA, uses a dual approach to robot motion [18]. The robot can choose between predefined motion patterns or real time computed trajectories. Motion patterns are used to implement certain specific tasks, like shooting the ball, and are programmed *by hand* off line. Real-time trajectories are instead obtained by solving the inverse kinematics equations with respect to the desired goal position. These two ways to solve the problem are shared by other teams as well. The AkDong team (Univ. of Manitoba) also uses gaits preprogrammed by hand, while teams like NimbRo [14] or the Humboldt [19] team solve the inverse kinematic equations.

V. CONCLUSIONS

In this paper we have outlined challenges, results and currently used approaches to humanoid robots motion planning in the context of robot soccer. Extrapolating what happened in the small and middle size leagues, it is envisioned that the development of effective algorithms to move humanoid robots is one of the key aspect to develop winning robots. An analysis of the team description papers produced for the Robocup 2006 competition outlines that many teams still rely on many off-line programmed motions. This is understandable if one looks at the humanoid platforms used. Most of them rely on computational devices with modest computational power. On top of that, most of this power is typically devoted to vision sensor processing. With the unavoidable advent of faster and cheaper processing units to equip custom built or general purpose humanoid robots, the execution of online algorithms could become feasible. However, it is also evident that currently proposed approaches need to be nevertheless improved in order to match the unusual characteristics distinguishing humanoid robot soccer. An aspect currently largely neglected, for example, is the generation of energy efficient gaits.

In the authors' opinion, humanoid soccer represents one of the most challenging fields for researchers in motion planning, and it still offers plenty of exciting unsolved issues.

REFERENCES

[1] J. Reif, "Complexity of the mover's problem and generalization," in *Proceedings of the 20th IEEE Symposium on Foundations of Computer Science*, 1979, pp. 421–427.

[2] J. Canny, *The Complexity of Robot Motion Planning*. MIT Press, 1988.

[3] J. Baltes, S. McCann, and J. Anderson, "Humanoid robots: Abarendou and daodan," in *Team description paper for Robocup 2006*.

[4] N. Mayer, M. Ogino, R. da Silva Guerra, J. Broedecker, S. Fuke, H. Toyama, A. Watanabe, K. Masui, and M. Asada, "Jeap team description," in *Team description paper for Robocup 2006*.

[5] S. Carpin, "Randomized motion planning – a tutorial," *International Journal of Robotics and Automation*, vol. 21, no. 3, pp. 184–196, 2006.

[6] J. Kuffner, S. Kagaami, K. Nishiwaki, M. Inaba, and H. Inoue, "Dynamically-stable motion planning for humanoid robots," *Autonomous Robots*, vol. 12, no. 1, pp. 105–118, 2002.

[7] M. Vukobratović and B. Borovac, "Zero-moment point – thirty five years of its life," *International Journal of Humanoid Robotics*, vol. 1, no. 1, pp. 157–173, 2004.

[8] T. Arakawa and T. Fukuda, "Natural motion of biped locomotion using hierarchical trajectory generation method consisting of ga, ep, layers," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1997, pp. 211–216.

[9] S. LaValle and J. Kufner, "Randomized kinodynamic planning," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[10] S. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1999, pp. 473–479.

[11] Q. Huang, K. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi, and K. Tanie, "Planning walking patterns for a biped robot," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 280–289, 2001.

[12] L. Hu, C. Zhou, and Z. Sun, "Biped gait optimization using estimation of distribution algorithm," in *Proceedings of the IEEE-RAS Interntaional Conference on Humanoid Roots*, 2005, pp. 283–288.

[13] S. Behnke, "Online trajectory generation for omnidirectional biped walking," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006, pp. 1597–1603.

[14] S. Behnke, M. Schreiber, J. Stückler, H. Strasdat, and M. Bennewitz, "Nimbro kidsize 2006 team description," in *Team description paper for Robocup 2006*.

[15] J. Stückler, J. Schwenk, and S. Behnke, "Getting back on two feet: reliable standing-up routines for a humanoid robot," in *Proceedings of the 9th International Conference on Intelligent Autonomous Systems*, 2006.

[16] T. Guseo, E. Menegatti, S. Antonello, A. Maggi, F. Meli, D. Pippa, A. Pretto, M. Sergotti, and E. Pagello, "Artisti humnaoid team for robocup 2006," in *Team description paper for Robocup 2006*.

[17] P. Yue, C. Zhou, K. Peh, L. Hu, D. Tah, T. Lam, Y. Liu, and G. Hendra, "Robo-erectus: Kidsize and teensize soccer-playing humanoid robots," in *Team description paper for Robocup 2006*.

[18] H. Takayama, R. Matsumura, N. Shibatani, T. Imagawa, T. Maeda, T. Miyashita, T. Takahashi, Y. Akazawa, N. Yamato, and H. Ishiguro, "Teamosaka a (kid size) team description paper," in *Team description paper for Robocup 2006*.

[19] M. Hild, M.Jüngel, and S. Spranger, "Humanoid team humboldt team description 2006," in *Team description paper for Robocup 2006*.