Motion Planning for a Vigilant Humanoid Robot

Jean-Bernard Hayet1 Claudia Esteves2 Gustavo Arechavala3 Eiichi Yoshida4

Abstract—This paper presents a motion planner for a vigilant humanoid robot. In this context of surveillance, the robot task is to keep a distinctive point in the environment in sight during all of its motion. The method we propose consists of three main ingredients: (1) A motion planner for an appropriate simplified model of the walking robot, adapted to the particular needs of humanoid robots, that outputs an admissible path with local optimality properties. This path is guaranteed to satisfy the visibility constraints resulting both from the landmark and from the angular limits of the mechanical system; (2) A generic walking pattern generator that produces stable walking motions; (3) A generalized inverse-kinematics module to satisfy the remaining collisions and posture constraints, in particular the gaze direction. The effectiveness of this method is shown with several examples on the humanoid robot platform HRP-2.

I. INTRODUCTION

Along the recent years, the robotics community has seen the rise of humanoid robotics and the multiplication of their applications in human-made environments. Although these developments are mainly contained to scientific experiments, it becomes plausible to imagine humanoid robots accompanying us in our daily life, a few decades from now.

However impressive these robot capabilities are, there is still a lot to do with respect to the autonomy of their tasks. In this work, we place ourselves in the context of surveillance where a humanoid robot is in charge of visually monitoring a location provided by some specialized algorithm or by a human operator, inside a man-made environment. We define the monitoring task as the action of walking from a point to another one while keeping a point in sight all the time during the trajectory execution. For solving this task, we consider two levels of constraints when planning the motions of the humanoid robot: first, we search to satisfy the global constraints, i.e. visibility, mechanical limitations for the gaze, stability and global collision constraints; then, we tackle local constraints such as local collision avoidance and secondary tasks satisfaction. To satisfy the global constraints we present a two-step approach involving (1) a complete motion planner satisfying the visibility constraints for a reduced model of the humanoid robot, which we treat as a differential-drive system and (2) a walking pattern generator. Finally, to solve the local constraints along the trajectory, we use a generalized inverse-kinematics algorithm with task priorities.

Globally, the proposed approach constitutes an important alternative to the more classical insight that handles visibility constraints as a serie of local tasks defined in a prioritized-task framework. Here, the idea, which is the main contribution of this work, is to handle them at the global level, i.e. right inside the planning algorithm. We reached this goal by designing a complete motion planner for a reduced model of the system, that satisfies both visibility and global collision avoidance constraints. The planner is derived from our previous work on different differential drive robots (DDR) [1], [2], and includes several new features proper to humanoids. In particular, the trajectories resulting from this planner satisfy, in the absence of obstacles, a minimal distance criteria with a term penalizing backward-walking portions of the trajectory. We believe that these backward-walking motions are natural and unavoidable when enforcing visibility constraints (e.g., imagine a camera-man having to walk while recording a target). As a result, we get a series of footprints which are given as input to a walking pattern generator for a humanoid robot and the appropriate head configuration to enforce visibility constraints. Figure 1 gives an example of the generated footprints used to feed the pattern generator.

This article is organized as follows: In Section II we discuss the related work, in particular those that handle
visibility constraints as *tasks* with assigned priority inside a generalized inverse-kinematics approach. In Section III we present and justify the reduced model used for planning the walking motions. Section IV, the core of this work, describes the motion planner that enforces visibility constraints for the reduced model introduced in Section III. In Section V we describe how the whole-body motions of the humanoid robot are generated from the planner and in Section VI we present some simulated results with the humanoid platform HRP-2. Finally, in Section VII we conclude with a brief discussion of our approach and we present future research on this topic.

II. RELATED WORK

Most of the research related to this work lies in the area of visual servo-control. Among the works in this vast area are those that handle the visual constraints inside a prioritized stack of tasks framework. These stacks usually include, as tasks with a high priority, collision avoidance and maintaining visibility of a certain feature in the environment. In order to solve as many tasks in the stack as possible different approaches have been proposed, some enabling or disabling tasks and changing their priorities (e.g. [3],[4]) or redefining some of the subtasks when needed to ensure convergence (e.g. [5]).

A common observation throughout all these works is that a global strategy is needed beforehand to control the local motion provided by solving the stack of tasks. A global motion planner, such as the one we propose, can be set as layout to ensure convergence of environmental and visibility constraints. Such integrated approaches with a global supervisor have already proven to be effective when handling multiple tasks as for example in [6] and [7]. The latter works illustrate well the need to ensure visibility constraints to increase the autonomy of a humanoid robot.

Several motion planners for humanoid robots have also been proposed in the literature (e.g. [8], [9]. Our work is done in the same spirit as [9] where a reduced model of the humanoid robot is used to ensure environmental constraints and then a stack of tasks is used to carry a dumbbell around the environment. We propose a similar approach but ensuring in the first stage not only environmental constraints (collision-avoidance) but also visibility constraints using a more adapted simplified model of the system described hereafter.

III. PLANNING MODEL

Planning *global* locomotion patterns for humanoid robots, such as walking, represents a challenging problem. Some of the main difficulties are related to the inherent redundancy of this type of mechanism, its inevitable dynamic constraints and its underactuated nature. The approach followed by some of the motion planners reported in the literature (e.g. [9], [10]) is based on a two-stage strategy, the first stage being the search for a *global path* (if exists) for a simplified model. In much of the cases this model considers some of the underactuated parameters of the humanoid robot. For instance, two parameters \((x, y)\) for locating the body position on the plane and \(\theta\) its direction. If the planner finds a collision-free *global path*, then it solves the motion coordination of the whole-body in the second stage.

In this work, we consider a simplified model encoding as much as possible the geometric form of the feasible human-like trajectories for joining a pair of configurations in the free configuration space.

Some studies on the steering of human locomotion at the trajectory planning level suggest that, for a pair of configurations \((x, y, \theta)\) and in the absence of obstacle, humans choose a common walking pattern among an infinite number of solutions [11], [12]. Moreover, it has been validated experimentally [12] that a differential system satisfying the well-known nonholonomic constraint arising in several wheeled mobile robot is a good approximation of human walking.

The nonholonomic behavior of human walking is activated when the trace of the body direction corresponds to the tangent direction of the path. The differential coupling between the body position and direction is given by

\[
y \cos \theta - \dot{x} \sin \theta = 0.
\]

In [13], the authors proposed a nonholonomic model together with a cost function to approximate the geometric shapes of walking trajectories composed by pieces of clothoid arcs. However, other human locomotion strategies are also valid, e.g., *backward* and *sideward* steps. The switching decision between locomotion strategies may not be based only on the distance from the initial to final configurations but also for obstacle avoidance. Indeed, in [14] the authors unified in a single model together with an optimality criterion the *forward* and *sideward* human locomotion strategies. Such a model has been validated in the HRP-2 platform.

The proposed model here differs from the previous one in three senses. First, we consider a differential drive robot rather than the unicycle or its dynamic extension. This means that we do not locate the body position as the midpoint between the shoulders. In our case, each wheel defines the position of each shoulder. However, the body direction remains the same for both models. The second difference is that we incorporate *backward* steps in our strategy. Finally, we consider additional constraints on the robot sensor (e.g. vision): landmark visibility and mechanical constraints on the sensor angle and range. It appears that optimal paths for the differential drive system together with a minimum-distance cost function and these sensor constraints, are made of pieces of logarithmic spirals, straight lines and in-site rotation (see next section), which are primitives not in contradiction with respect to the human walking strategies.

IV. MOTION PLANNER

The motion planning algorithm for the reduced system, i.e. the DDR, is based on [1], that we expand here for the case of humanoids. For most of the following notations, refer to Fig. 2: the landmark \(L\) – or more generally the location of interest to monitor – is located at \((0, 0)\), and the robot position is \((x, y)\) in Cartesian coordinates, \((r, \alpha)\) in
polar coordinates. The robot orientation is $\theta$, while its gaze direction is $\phi$. The constraints for this system are three-fold:

1) Kinematic non-holonomy constraints resulting from the robot mechanism, i.e. Eq. (1).

2) Visibility constraint resulting from restricting the landmark to be at the center of the robot gaze,

$$\theta = \alpha - \phi + (2k + 1)\pi, \ k \in \mathbb{Z};$$

(2)

3) Physical constraints on the sensor angle and range,

$$\phi_1 \leq \phi \leq \phi_2,$$

$$d_{\text{min}} \leq r \leq d_{\text{max}}.$$  

(3)

(4)

The planner from [1] has the advantage of handling all of these constraints, and of being complete, i.e. it will find a trajectory solution if it exists. By construction, it also gives optimal paths in terms of Euclidean distance in the plane in the absence of obstacles, since it relies on the optimal path synthesis for systems satisfying Eqs. 1, 2 and 3 (Eq. 4 is handled at the upper, recursive level of the algorithm).

In the aforementioned article, the optimality criterion was the total Euclidean distance corresponding to the trajectories connecting the initial and final configurations, $P_i$ and $P_f$ in $SE(2)$. Here, to incorporate the fact that, for the humanoid robots, backward motion should be minimized, we modified the criterion $C$ to penalize backward motion over the set of possible trajectories $P : [0, T_P] \rightarrow SE(2)$, with $P(0) = P_i$ and $P(T_P) = P_f$, as follows,

$$C(P) = \int_0^{T_P} q(t) \sqrt{(x'(t))^2 + (y'(t))^2} dt,$$

$$P^* = \min_P C(P),$$

(5)

where $P(t) = (x(t), y(t), \theta(t))^T$, and

$$q(t) = \begin{cases} 1 & \text{if } \dot{x}(t) \cos \theta(t) + \dot{y}(t) \sin \theta(t) > 0, \\ Q & \text{if } \dot{x}(t) \cos \theta(t) + \dot{y}(t) \sin \theta(t) < 0, \end{cases}$$

where $Q > 1$ is a constant penalizing term for backward motion that constitutes a parameter of the algorithm. In the following, we describe the nature and distribution of the shortest paths according to $C$ when no obstacle is present, and then describe the planning algorithm among obstacles.

A. Shortest paths in the absence of obstacles

In the absence of obstacles, we rely on our previous work [2] to get the full synthesis of shortest paths for a DDR to connect any two configurations of $SE(2)$, while keeping a landmark in sight. A nice local property of these optimal trajectories is that they are either line segments, in-site rotations or logarithmic spirals. The spirals correspond to trajectories saturating $\phi$ at $\phi_1$ or $\phi_2$, i.e. satisfying

$$r(t) = r_p e^{-\frac{\alpha(t)}{Q}}, \text{ where } i = 1, 2,$$

and where $P = (r_p, \alpha_P)$ is any point of the spiral. Let us denote in the following logarithmic spirals trajectory parts as “$S_1$” or “$S_2$” (according to at which angle the sensor is saturated, $\phi_1$ or $\phi_2$), line segments trajectory parts as “$L$”, in-site rotations trajectory parts as “$R$”. From this local characterization and by using simple geometric arguments [2], one can prove that the resulting trajectories are to be found among the following types: $L$, $L - S_2$, $S_2 - L$, $S_2 - R - S_1$, $L - S_2 - R - S_1$, $S_2 - R - S_1 - L$, $L - S_2 - R - S_1 - L$ and 6 other types that can be obtained by exchanging $S_1$ and $S_2$. In the same article, a plane partition according to the relative position of initial and final points $P_i$ and $P_f$ is given.

The extension to the criterion $C(P)$ of Eq. 5 instead of the Euclidean distance is straightforward, as the $Q$ factor only affects: (1) the spatial distribution of the nature of optimal curves and (2) the parameters of the curves involving both backward and forward motion. We spare the fastidious algebra to the reader, and give the resulting plane partition, for some $Q > 1$, in Fig. 3. As an example, the region including the figure upper right corner is the one where $L - S_2 - R - S_1 - L$ trajectories are the shortest. We heavily rely on this partition in the main planning algorithm below.
To illustrate the effect of \( Q \), we give in Fig. 4 two examples of optimal trajectories, without obstacle, for the same pair \((P_1, P_f)\). The upper one is the \( L - S_2 - R - S_1 - L \) trajectory obtained for \( Q = 1 \), the lower one is the trajectory of the same type obtained for \( Q > 1 \). Note that the first part of the trajectory (straight line, then spiral \( S_2 \)) is done forwards whereas the second part (spiral \( S_1 \), then straight line) is done backwards. One can observe that the effect of \( Q \) is to reduce the backward part to a simple straight line.

**B. Planning paths among obstacles**

The planning algorithm among obstacles utilizes a scheme “a la Laumond” [15], that, first, computes a path for an equivalent holonomic system and, then, recursively divides the resulting path in two whenever the starting and ending point of the sub-paths cannot be connected by the optimal primitives described in the previous sub-section. In this scheme, the landmark non-visibility notion is translated into the generation of (virtual) obstacles corresponding to the shadows generated by the obstacles with a (virtual) light located on the landmark. Another remark is that the holonomic path for the system in \( SE(2) \) can be easily deduced from paths of points in \( \mathbb{R}^2 \), as we showed it in [1]. Hence, steps 1 and 2 of the algorithm hereafter captures the connectivity of the free space \( C^g_{free} \) with a Generalized Voronoi Graph (GVG) for a circular robot without considering its orientation (i.e., in \( \mathbb{R}^2 \)). It dilates the physical obstacles and takes their union with the aforementioned shadows. The final algorithm is quite similar to [1], and now includes an optimization phase (step 5) and a generation of footprints for the humanoid robot (step 6):

1. Build a representation of \( C^g_{obst} = C^f_{free} \) as the union of the dilated obstacles with the shadows induced by the landmark visibility;
2. Build the GVG on \( C^g_{free} \); \( C^g_{free} \) being made of parts of lines or circles, the GVG is made of parts of lines, parabolas or hyperbolas;
3. Given a starting and a goal configurations, compute a path \( \tilde{s} \) for the holonomic system associated to the robot by connecting these locations to the GVG; if not possible, no non-holonomic path can be found as well;
4. Recursively connect the starting and ending points with the optimal primitives of IV-A; whenever the sub-paths induced by these primitives are in collision, use the point at middle-path in \( \tilde{s} \) as a sub-goal and re-apply the recursive procedure to the two resulting sub-paths;
5. Optimize the final path: (i) generate \( n_s \) randomly possible shortcuts between the constitutive primitives of the trajectory and select the one that improves \( C(\mathcal{P}) \), and (ii) iterate \( n_s \) times the same procedure;
6. Transform the resulting path into a set of footprints for the humanoid robot, i.e., into sets of left and right feet positions, given the mechanical robot properties.

We illustrate various of these steps of the previous algorithm in Fig. 1: \( C^g_{free} \) is depicted in yellow, the obstacles (step 1 of the algorithm) in red (or dashed red for the dilated ones), and the GVG built onto \( C^g_{free} \) (step 2) in blue. The resulting trajectory (step 5) is depicted in pink, and the resulting left and right footprints (step 6) appear in dark blue. Note that the resulting trajectory is free of collision for the dilated obstacles, which allows the footprints to be free of collision for the real obstacles. Also note that our implementation assigns weights to the GVG ponderated both by distance from a node to another and by the minimal clearance along the edge. It is also important to stress that this recursive algorithm necessarily converges, because of the local controllability properties of our reduced system [1].

**V. MOTION GENERATION**

In the next step of our algorithm, the footprints that were computed through the motion planner, described in the previous section, are used as input for a dynamic pattern generator for humanoid robots [16]. The latter transforms the computed footprints into a dynamically executable motion. This pattern generator is based on the preview control of the Zero Moment Point (ZMP) of an invert pendulum model, as in [17]. The resulting walking pattern is dynamically stable, as it maintains the ZMP inside the support polygon formed by the feet (foot in single-support phase).

At this point of the algorithm, we have a dynamic walking pattern following the path specified by the motion planner. The remaining task is to ensure that the robot head is directed to the landmark. We are sure, because of the nature of the planned path, that visibility constraints can be met with the robot sensing capabilities but these constraints have not been met yet. Therefore, the last stage of our algorithm is to apply a generalized inverse kinematics solver to the upper-body of the robot at each sample of the dynamic trajectory as in [9]. Here, the inverse kinematic task is only to direct the gaze of the robot towards the landmark.

We work under the assumption that the displacement achieved in the \( 5ms \) timeframe between each sample of the dynamic trajectory is small enough to relate linearly, through Eq.6 a change \( \delta \rho \) in the gaze direction \( \rho \) to a change \( \delta \gamma \) in the robot posture \( \gamma \),
where $J$ is the Jacobian matrix for the corresponding task. As we are dealing with a mechanism that is redundant with respect to the imposed gazing task, we use classically Eq. 7 to solve the inverse kinematics problem,

$$\delta \gamma = J^\dagger \delta \rho + (I_n - J^\dagger J) \mu,$$

where $J^\dagger$ is the pseudoinverse of the task Jacobian matrix, $n$ the total number of DOF of the robot, $I_n$ the n-dimensional identity matrix, and $\mu$ an arbitrary vector to optimize.

The advantage of such a IK solver for a gazing task is the remaining flexibility to perform other tasks (e.g. grasping an object, avoiding remaining collisions or adjusting posture due to an unexpected event) while the trajectory is executed. In this work we only ensure the visibility constraint.

VI. RESULTS

The experimentations presented here are simulation realized by using widespread software libraries. The CGAL library was used for the 2D planner; we benefited in particular from the implementation of Voronoi diagrams. The pattern generation, including dynamics simulation and motion controllers, was done using the Open-HRP platform [16], and the final visualization of trajectories with Matlab.

We have conducted several simulations on different scenarios. On the first row of Fig. 5 the path and trajectory for the first scenario is shown. Here, no obstacles are present and therefore the computed path is optimal in distance with a penalty term for backward motions. Notice how the robot keeps the landmark in view during the whole motion. On the second and third rows two other scenarios, this time with obstacles, are shown. Here the robot avoids the obstacles through its path. These collision avoidance is done, for a bounding box of the robot and therefore no 3D collision-avoidance has been computed. It could be included inside the stack of tasks. The 2D path image shows the footprints computed around the obstacles as well as the Voronoi diagram for each of the environments. Finally, Fig. 6 shows a close-up of some configurations near the landmark with the robot gaze directed toward it.

Movies and more snapshots can be found at: http://www.cimat.mx/~jbhayet/VDDR/

VII. CONCLUSIONS AND FUTURE WORK

The approach we presented tackles the problem of planning motion for humanoid robots with sensory constraints imposing a landmark to be visible during the whole trajectory. Instead of many other current approaches, we incorporate these constraints at the planning level, which processes a reduced model of the robot (a differential drive), that fits particularly well for trajectory planning tasks in humanoid robotics. The planner delivers footprints that feed a walking pattern generator, and inverse kinematics is finally used to properly configure the upper part of the body to satisfy the visibility constraints. This approach has been validated on simulation examples for the HRP-2 humanoid robots and we currently aim to implement it on the real HRP-2 robot.

Among ongoing and future work, we intend to generalize the approach to several landmarks, and modify the planner to incorporate lateral motions that are currently not supported.

VIII. ACKNOWLEDGMENTS

The authors acknowledge CONCYTEG for partially funding this work (project 09-02-K662-073). They warmly thank Gerardo Jarquin from CINVESTAV Saltillo for his precious help in the 3D trajectory visualization, and Kawada Industries for allowing us to use the HRP-2 VRML model.

REFERENCES

Fig. 5. Three examples of scenarios with (2nd and 3rd rows) or without (1st row) obstacles. The left column shows, in three cases, the trajectories computed by the 2D planner that allows for sensor restrictions to be met all along the trajectories. The spiral and line parts are quite distinguishable; the blue structures are the edges of the underlying GVG. The right column displays several configurations of the real trajectory by the humanoid. The landmark is represented by a yellow ball.

Fig. 6. Individual configurations of the HRP-2 robot extracted from the trajectories above. Note that the inverse kinematics tend to use many of the degrees of freedom of the body upper part to set the gaze onto the landmark, e.g. in the left frame.