Statically Stable Legged Locomotion with Leg Redundancy

D. Prattichizzo A. Bicchi G. Casalino

D.S.E.A and Centro "E.Piaggio", Università di Pisa
via Diotisalvi 2, 56126 Pisa, Italy

Abstract

This paper deals with the problem of planning and controlling the trajectories of the legs of a mobile robot moving through statically stable configurations. Legs are supposed to possess significant weight compared to the chassis, and a degree of redundancy that can be used to maximize the robustness of equilibrium. We use optimal control techniques for planning joint trajectories that comply with given feet trajectories, equilibrium and workspace constraints. This method implies iterative solutions and is used in a preliminary off-line phase. A real time control law is then derived, that can be applied to on-line control of the above trajectories while accommodating for unexpected external disturbances.

1 Introduction

This work focus its attention on statically stable walking machines. Their peculiarity is to be in equilibrium even when the power or the controller are turned off or when unexpected failures occur. A wide literature is devoted to the analysis of walking vehicle gaits. Choi and Song studied fully automated gaits which can be used to cross over standard types of obstacles [3]. In the work of Song and Waldron [10], an analytical approach of the gait study was developed. A generalized method producing standard foot trajectories for a walking vehicle was studied in [5].

The control hierarchy for walking vehicle motion envisioned by most contributions comprises a high-level gait planner that generates feet trajectories for moving the entire vehicle along given trajectories. Equilibrium of the vehicle is guaranteed provided that the weight of the legs is negligible compared to that of the body and that no external disturbances act on the vehicle. Given feet trajectories, if the walking robot possesses a degree of redundancy, it can be exploited to realize a more robust equilibrium of locomotion.

We propose an algorithm which accepts feet trajectories as input from a higher-level gait planner and, by means of an optimizing inverse kinematics algorithm, generates joint trajectories. Its peculiarity consists in exploiting redundancy to maximize the equilibrium stability while ensuring other characteristics of the walk, such as to maintain an average heigth and an orientation of the chassis. The algorithm can be employed on-line for continuously solving reference feet trajectories into joint trajectories. By means of the equilibrium stability optimization, a reflexive-type behaviour is exhibited by the robot vehicle. Its reaction to disturbances, e.g. to unexpected changes of the center of mass position, is to recover an equilibrium configuration by balancing its weight with the exploitation of redundancy.

2 Optimal trajectory planning algorithm and control

In general, the trajectory planning problem for redundant robots is comprised of: a specified task-space trajectory to be followed exactly, $\hat{\mathbf{x}}(t) = \mathbf{f}(\mathbf{q}(t))$, being $\mathbf{f}(\mathbf{q})$ the direct kinematic relationship of the redundant robot whose inverse may not be available in closed form; a set of inequality constraints on joint variables $\mathbf{s}(\mathbf{q}) < 0$, arising from a variety of limitations such allowable range of joints, tilt of the chassis, degree of static stability; and of a cost function of joint velocities $\mathbf{c}(\dot{\mathbf{q}})$, arising from limitations on centrifugal and Coriolis forces, to be minimized. The differential kinematic relationship, $\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$, can be used to locally linearize and iteratively invert kinematics. For instance, by using a generalized right inverse $\mathbf{J}^R$ and a basis $\mathbf{N}$ of the kernel of $\mathbf{J}$, the local inverse mapping can be written as

$$\dot{\mathbf{q}} = \mathbf{J}^R\dot{\hat{\mathbf{x}}} + \mathbf{N}\mathbf{u}. \qquad (1)$$

Eq.(1) has been used to arrange a closed-loop algorithm for inverse kinematics [11] [4] [9] [2].

The optimal trajectory planning problem from time 0 to T, with mixed constraints, can be stated as follows:

$$\begin{array}{ll} \text{Minimize} & P = \int_0^T 0.5\dot{\mathbf{q}}^T\mathbf{W}\dot{\mathbf{q}}dt \\ \text{subject to:} & \dot{\mathbf{q}} = \dot{\mathbf{q}} \\ & \mathbf{f}(\mathbf{q}(t)) - \hat{\mathbf{x}}(t) = 0, \forall t \\ & \mathbf{s}(\mathbf{q}(t)) < 0 \end{array} \qquad (2)$$

This problem is difficult to deal with because of the presence of inequality constraints. In the following a penalty function approach to solve the constrained optimization problem is described. Consider a particular choice of penalty functions, namely superquadrics. The superquadric penalty associated with a scalar inequality constraint $s_i(\mathbf{q}) < 0$ is

$$V_i = \begin{cases} \frac{1}{2\, s_i^{2q}} & s_i(\mathbf{q}) \leq -\kappa \\ a\, s_i^2 + b\, s_i + c & s_i(\mathbf{q}) > -\kappa \end{cases} \tag{3}$$

where κ is a small positive number, and the constants a, b, and c can be chosen to make V_i at least twice continuously differentiable with respect to s_i. As the superquadric exponent q is increased, a better approximation of a step function in s_i is obtained, so that little changes in V_i correspond to different values $s_i < 0$, while V_i is very steep outside the κ boundary layer. Inequality constraints on joint velocities are dealt with analogously.

Using (1) and replacing inequality constraints with the summation of penalty functions $V = \sum V_i$, the problem (2) can be restated as

$$\begin{array}{ll} \text{Minimize} & P = \int_0^T V(\mathbf{q}(t)) + 0.5\dot{\mathbf{q}}^T \mathbf{W} \dot{\mathbf{q}} dt \\ \text{subject to:} & \dot{\mathbf{q}} = \mathbf{J}^R \dot{\hat{\mathbf{x}}} + \mathbf{N}\mathbf{u} \\ & \mathbf{q}(0) = \mathbf{q}_0, \mathbf{f}(\mathbf{q}_0) = \hat{\mathbf{x}}_0 \end{array} \tag{4}$$

Upon derivation of the Hamiltonian associated to this optimum control problem,

$$H = -V(\mathbf{q}) - 0.5\dot{\mathbf{q}}^T \mathbf{W} \dot{\mathbf{q}} + \lambda^T (\mathbf{J}^R \dot{\hat{\mathbf{x}}} + \mathbf{N}\mathbf{u}),$$

from the maximum condition of the Hamiltonian, the optimal control $\hat{\mathbf{u}}$ is obtained:

$$\hat{\mathbf{u}} = (\mathbf{N}^T \mathbf{W} \mathbf{N})^{-1} \mathbf{N}^T (-\mathbf{W} \mathbf{J}^R \dot{\hat{\mathbf{x}}} + \lambda) \tag{5}$$

while the adjoint variable dynamics are

$$-\dot{\lambda}^T = -\frac{\partial V(\mathbf{q})}{\partial \mathbf{q}} \quad \text{with} \quad \lambda^T(T) = 0. \tag{6}$$

It is well known that the solution of such a non-linear optimum problem can be determined only with the use of forward-backward iterative procedures, such as, for instance, the so called gradient method [8], [1]. For a discussion of the application of optimal control tecniques to trajectory planning for robots, see also [6] [7]. The iterative nature of the solutions to this problem only allows their use in an off-line preliminary planning phase.

On the other hand, in order to face with the real time implementation of trajectory control, we can simplify the optimal control approach by reducing the time horizon to a very short value in order to eliminate backward recursion on the adjoint system differential equations. So, considering only very short time horizons,

$$\lambda(T-\tau) = \lambda(T) - \dot{\lambda}(T)\tau + O(\tau^2) \approx -\left.\frac{\partial V}{\partial \mathbf{q}}\right|_T \tau,$$

whence the optimal trajectory

$$\hat{\mathbf{u}} = (\mathbf{N}^T \mathbf{W} \mathbf{N})^{-1} \mathbf{N}^T (-\mathbf{W} \mathbf{J}^R \dot{\hat{\mathbf{x}}} - \frac{\partial V}{\partial \mathbf{q}} \tau). \tag{7}$$

Joint velocities result

$$\dot{\mathbf{q}} = (\mathbf{I} - \mathbf{N}\mathbf{N}_W^+) \mathbf{J}^R \dot{\hat{\mathbf{x}}} - \mathbf{N}\mathbf{N}_W^+ \mathbf{W}^{-1} \frac{\partial V}{\partial \mathbf{q}} \tau, \tag{8}$$

where

$$\mathbf{N}_W^+ = (\mathbf{N}^T \mathbf{W} \mathbf{N})^{-1} \mathbf{N}^T \mathbf{W}. \tag{9}$$

Including τ in a multiplicative coefficient ζ,

$$\dot{\mathbf{q}} = (\mathbf{I} - \mathbf{N}\mathbf{N}_W^+) \mathbf{J}^R \dot{\hat{\mathbf{x}}} - \zeta \mathbf{N}\mathbf{N}_W^+ \mathbf{W}^{-1} \frac{\partial V}{\partial \mathbf{q}} \tag{10}$$

Law (10) represents the velocities of the q-variables solving, in very short time horizon, the problem (2) with penalty functions. Observe that, being $\hat{\mathbf{q}}$ the solution of the off-line optimum problem, in the on-line problem formulation, a suitable cost function, weighting the distance of $\mathbf{q}$ from $\hat{\mathbf{q}}$, can be added, with the aim of keeping the actual trajectory as close as possible to the absolute optimum.

The first term of the right hand size of the continuous time law (10) corresponds to an open loop trajectory control that, when implemented on a digital computer in a discretized version, produce trajectory drifts due to numerical errors. To avoid this problem a closed loop law for the trajectory control is used,

$$\dot{\mathbf{q}} = (\mathbf{I} - \mathbf{N}\mathbf{N}_W^+) \mathbf{J}^R (\hat{\mathbf{x}} - \mathbf{f}(\mathbf{q})) - \zeta \mathbf{N}\mathbf{N}_W^+ \mathbf{W}^{-1} \frac{\partial V}{\partial \mathbf{q}}. \tag{11}$$

It can be proved that, for slowly varying trajectory, the error $\mathbf{e} = \hat{\mathbf{x}} - \mathbf{f}(\mathbf{q})$ converges in module to zero and that the cost function reaches a minimum value [4] [2].

3 Case Study

The model of the vehicle we refer to, for simulations, is a planar biped, schematically depicted in (1), fig.1. Two legs are joined to the chassis, each composed of three links, of length l, connected by revolute joints. Let C_0 and C_c be the world frame and the frame fixed on the chassis, respectively. The robot configuration is completely positioned by the generalized coordinates of the structure. In details, let the 3-vectors $\mathbf{q}_1$ and $\mathbf{q}_2$ be the joint variables of leg 1 and leg 2 w.r.t. the chassis frame; and the 3-vector $\mathbf{p} = (x_0, y_0, \theta)^T$ be the position and the orientation of the chassis frame w.r.t. the world frame. The chassis is supposed four times heavier than each link of the leg. The case that the weight of the leg is not negligible compared to that of the chassis often occurs in practice, but it is not always considered in literature. Neglecting the weight of the legs may cause an assumedly statically-stable gait to fail in guaranteeing equilibrium. To define the motion planning problem with mixed constraints (2) for the biped locomotion, suppose that a feet trajectory ${}^d\mathbf{x}_f$ to be followed has been generated. The equality constraint corresponding to the assigned trajectory results ${}^d\mathbf{x}_f = \mathbf{f}_f(\mathbf{q})$,, where $\mathbf{q}^T = (\mathbf{p}^T, \mathbf{q}_1^T, \mathbf{q}_2^T)^T$ and $\mathbf{f}_f(\mathbf{q})$ is the direct kinematics of the feet.

The first constraint considered is related to the bounds on the location of the center of mass projection (M.C.P.). The objective of the associated penalty function is to constrain the biped M.C.P. within the stable region. Let x_G and x_S be the x-coordinate, in the world frame, of the center of mass and of the middle point of the supporting region. This is the foot base (of length a) if only one foot is supporting the robot and the entire region between the two feet if both feet are supporting the biped. If the robot is supported by only one foot, inequality constraints on the biped M.C.P. can be written as

$$\begin{cases} S_{G_M}(\mathbf{q}) &= x_S(\mathbf{q}) + \frac{a}{2} - x_G(\mathbf{q}) &\leq 0 \\ S_{G_m}(\mathbf{q}) &= x_G(\mathbf{q}) - x_S(\mathbf{q}) + \frac{a}{2} &\leq 0. \end{cases} \quad (12)$$

Penalty functions V_{G_M} and V_{G_m}, associated with the higher and the lower bound respectively, can be designed according to (3). Other inequality constraints concern the chassis tilt angle, height and joint range limits. The chassis orientation θ is bounded to lie within the interval $[\theta_m, \theta_M] = [-0.1, 0.1]rad$ and the heigth h within the interval $[h_m, h_M] = [1.5l, 3l]$. In order to satisfy mechanical constraints of the joints of the legs, penalty functions $V_{q_{1i}}$ and $V_{q_{2i}}$ bound joint angles to be within sectors of $2\frac{\pi}{3}$. Also, a constraint on the angles of incidence of the legs with the horizontal axis is imposed with a cost function. A total cost function results

$$V_t = V_G + V_\theta + V_h + V_{s1} + V_{s2} + \sum_{i=1}^{3} V_{q_{1i}} + \sum_{i=1}^{3} V_{q_{2i}} \quad (13)$$

where the single terms are the sum of the penalty functions associated with the higher and the lower bounds.

4 Simulations

Simulations have been executed showing the capabilities of the real time trajectory control algorithm. To implement the on-line closed loop control law (11) on a digital computer, a discrete-time version of the algorithm is employed,

$$\begin{aligned} \mathbf{q}_{k+1} - \mathbf{q}_k &= (\mathbf{I} - \mathbf{N}\mathbf{N}_W^+)\mathbf{J}^R({}^d\mathbf{x}_f^{i+1} - \mathbf{f}_f(\mathbf{q}_k)) + \\ &\quad -\xi \mathbf{N}\mathbf{N}_W^+ \mathbf{W}^{-1} \frac{\partial V_t}{\partial \mathbf{q}}(\mathbf{q}_k), \end{aligned} \quad (14)$$

where ${}^d\mathbf{x}_f^{i+1}$ is the $(i+1)$-th point of the discretized feet trajectory $({}^d\mathbf{x}_f^1, {}^d\mathbf{x}_f^2, \ldots)$. Suppose the robot is in a configuration with joint vector $\mathbf{q}_i$, and feet position ${}^d\mathbf{x}_f^i = \mathbf{f}_f(\mathbf{q}_i)$. When the high-level planner updates

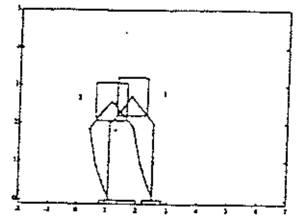

Figure 1: Initial and final configurations

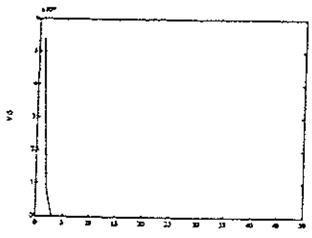

Figure 2: Penalty function on M.C.P.

${}^d\mathbf{x}_f$ to the next value ${}^d\mathbf{x}_f^{i+1}$, feet position error becomes $({}^d\mathbf{x}_f^{i+1} - \mathbf{f}_f(\mathbf{q}_i))$. The algorithm output, $\mathbf{q}_{i+1}$, solves the inverse kinematics ${}^d\mathbf{x}_f^{i+1} = \mathbf{f}_f(\mathbf{q}_{i+1})$ and minimizes the cost function guaranteeing that inequality constraints are not violated. For obtaining a good performance of the trajectory control system, feet trajectories have to be updated at a lower rate than the closed loop dynamics.

In the first simulation the performance of the algorithm (14) is described during a single reference trajectory step. The initial configuration is the (1) in fig.1. The center of mass (star), falls on the middle point of the stable region included between the two feet both in supporting phase. Suppose that the robot has to prepare to take a step moving the right foot. Being the left foot the next supporting one, the robot has to move its center of mass projection within the base of this foot. To prepare the robot for a step, the high-level planner inputs to the closed loop trajectory control (14) consist of new limits for the center of mass projection x_G, to be constrained in the left foot base, while feet reference position is fixed.

The closed loop algorithm (14) acts taking the center of mass into the new bounds. When convergence is reached and the constraints are satisfied the resulting configuration is as depicted in fig.1 (2). The center of mass falls within the foot base and the height h is better centered in the interval $(1.5l\ ,\ 3l)$. During the convergence phase the evolution of V_G, reported in fig.2, is decreasing at a minimum value. The behaviour of error feet position reaches the zero-value in about 5 steps of the inner loop. Values assumed by the orientation are reported in (fig.3): the orientation angle θ reaches the zero

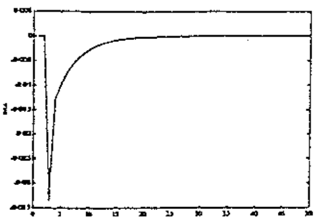

Figure 3: Chassis orientation

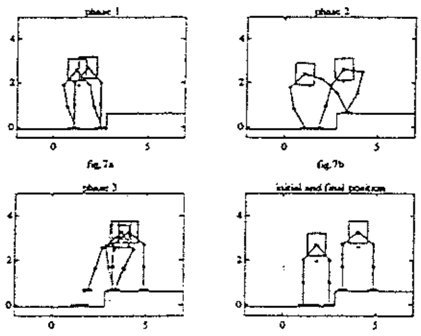

Figure 4: Climbing a step

value, corresponding to the penalty function minimum, in about 50 steps. Analogous behaviours characterize all controlled variables.

In the second simulation, the robot task is to climb a step. As before, the given feet trajectory is followed point to point using (14). The results are summarized in (fig.4), where the most significant postures are shown. In phase 1 the biped gets ready to start: the center of mass adjustment is caused by the penalty function V_G. In phase 2, the robot has placed its right foot on the step and before going up with the other leg, shifts its body forward to let the center of mass fall within the base of foot 2. Phase 3 depicts three robot configurations: in the first, the robot has raised the left foot, supporting itself on the right. Fig.4d summarizes the initial and final configuration.

5 Conclusions

The problem of planning and controlling joint trajectories of statically stable walking robots have been investigated. We have supposed that the robot legs possess a significant weight and a degree of redundancy.

An algorithm has been proposed to exploit redundancy, so that the equilibrium of the walking vehicle is optimized and workspace constraints are satisfied complying with given feet trajectories. Allowable penalty functions have been used to include inequality constraints in the optimum problem formulation. The solution of the optimum problem with a finite time horizon T can only be obtained by iterative procedures and is therefore suitable for implementation as an off-line preliminary phase. The on-line execution of the optimal planning procedure via forward-backward recursion in real time, by exploiting the intrinsic repetitive nature of the gait, will be investigated in the sequel of this research. A simplification of the optimal control approach by reducing the time horizon to a very short value has been studied. Its implementation in an on-line trajectory control provides walking robots with a reflexive-type behaviour. Simulations on a biped robot, showing the effectiveness of the on-line trajectory control algorithm, has been presented.

References

[1] A.E. Bryson, Y.-C. Ho, "Applied Optimal Control", (Blaisdell, Waltmann, Mass., 1969).

[2] P. Chiacchio, S. Chiaverini, L. Sciavicco, B. Siciliano, "Closed Loop Inverse Kinematics Schemes for Constrained Redundant Manipulators with Task Space Augmentation and Task Priority Strategy", Int. Journ. of Robotics Research vol.10, n.4, 1991, pp.410-425.

[3] B.S. Choi, S.M. Song, "Fully Automated Obstacle-Crossing Gaits for Walking Machines", IEEE Int. Conf. on Robotics and Automation, 1988.

[4] H. Das, J-J.E. Slotine, T.B. Sheridan "Inverse Kinematic Algorithms for Redundant Systems", IEEE Int. Conf. on Robotics and Automation, 1992.

[5] S. Hirose, O. Kunieda, "Generalized Standard Foot Trajectory for a Quadruped Walking Vehicle", Int. Journ. of Robotics Research vol.10, n.1, 1991, pp.3-12.

[6] Y. Nakamura, H. Hanafusa and T. Yoshikawa, "Task-Priority Based Redundancy Control of Robot Manipulators", Int. Journ. of Robotics Research vol.6, n.2, 1987, pp.3-15.

[7] Y. Nakamura, "Advanced Robotics Redundancy and Optimization" (Addison-Wesley Publishing Company, 1991).

[8] A.P. Sage, "Optimum Systems Control" (Prentice-Hall, Inc., 1968).

[9] L. Sciavicco, B. Siciliano, "A Solution Algorithm to the Inverse Kinematic Problem for Redundant Manipulators", IEEE Journ. of Robotics and Automation vol.4, n.4, 1988, pp.403-410.

[10] S.M. Song, K.J. Waldron, "An Analytical Approach for Gait Study and Its Applications on Wave Gaits", Int. Journ. of Robotics Research vol.6, n.2, 1987, pp.60-71.

[11] W.A. Wolovich, H. Elliot, "A Computational Technique for Inverse Kinematic ", Proc. 23nd Conf. on Decision and Control, Las Vegas 1984.