The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Gait generation for a quadruped robot using Kalman filter as optimizer

Rafael Fontes Souto, Geovany Araújo Borges and Alexandre Ricardo Soares Romariz

Abstract— **In this paper, the kinematic model of a quadruped robot is derived. The model is equivalent to that of a parallel manipulator, in that each leg can be seen as a manipulator. However, the model is extended to consider that in one gait cycle some legs are in contact with the ground and others are not. In order to obtain the inverse kinematics model, this paper presents as contribution the use of the extended Kalman filter as optimizer in two different situations of the leg motion: unconstrained case, for the swing leg(s), and constrained case, for the leg(s) in contact with the ground. This method was evaluated for locomotion in plain and inclined surfaces. The results obtained with the kinematics model were satisfactory when implemented in a point-to-point trajectory in simulation, and also in an experiment with a four-legged platform with three degrees of freedom in each leg.**

I. INTRODUCTION

Wheeled robots are limited in the types of terrain that they can navigate. In contrast, legged robots can navigate a much wider type of terrain. However, achieving kinematic or dynamic stability for most of the gaits of these robots is a complex process that needs to address multiple Degrees Of Freedom (DOF), specially when considering multiple constraints, gait optimization and adaptation to terrain. Hence, kinematic modelling represents a particularly important issue when dealing with legged robots.

In recent years, given the difficulties of traditional designs for legged robots, biologically-inspired approaches are considered. In these approaches, robots are designed to crawl, gallop or even trot, and some of them are capable to shift between two different gaits using a hybrid motion model [1]. According to Xu *et al.* [2] and Fukuoka *et al.* [3], dynamic walking simulations on irregular terrains show that biologically-inspired control has the potential ability of autonomous adaptation.

In order to generate rhythmic movements and control the locomotion of legged robots, the biological Central Pattern Generator (CPG) approach based on oscillators is the most popular in the literature. A CPG based on Matsuoka's oscillator was developed in [4] and [5] to create rhythmic motion for a quadruped robot. Other CPGs based on the phase of the oscillators and sensory feedback were also treated in [6] and [7]. A different architecture based on a hierarchical control system is presented by Kolter *et al.* [8] and extensive experiments were made to verify whether the controller proposed is able to robustly cross a wide variety of challenging terrains. Even master-slave quadruped walking robots using radio control devices were showed to work by experimental results, although they have problems with stability and direction control [9]. An approach using sensory feedback for the observation of gravity load and stimulus-reaction mechanism provided several significant tips on the control of the quadruped walking robot, as shown in [10]. There are many more ways to control the locomotion of legged robots, *e.g*, Variable Constraint Control [11], Graph Search Method [12], [13], Fuzzy Control [14], [15], [16], [17], among others [18], [19], [20]. However, all these biological inspired robots still have many limitations related with their locomotion when compared to living creatures, in general a consequence of their lower degree of freedom.

In the particular case of four legged robots, despite the complexity of control, the stability is not difficult to be achieved in static gaits, since at all times three legs can be lowered to a stable tripod while the other leg moves. In dynamic gaits, such as gallop, it is possible to maintain the stability even when for short periods of time none of the legs touches the ground. Many researches deal with these aspects of stability in gaits [21], [22], [23] and attitude control [24], [25]. However, this is out of the scope of this work, which focuses on the motion generation in a low speed walk, that is, a static gait. Actually, we are interested in the interactive and real-time solution of the inverse kinematic model of a four-legged robot with three DOF in each leg. Such solution must consider the constraints in supporting legs.

Recently, some development was done in kinematic modelling. Al-Zaydi and Amin [26] and Gasparetto *et al.* [27] developed algorithms for solving the forward and inverse kinematics problems and simulated the locomotion of legged mobile robots. The algorithms are based on a geometric approach where the solution is obtained by trigonometric relations between the leg geometry parameters and the angles related with the joint variables. Nevertheless, the task of calculating all of the joint angles that would result in a specific position of one foot of the robot is not so simple due to the singularities involved. Thus, some authors preferred to solve the inverse kinematic problem through artificial neural networks [28], [29] or reinforcement learning [30], [31]. Pechev [32] solved it without matrix inversion, although the work by P. Lin *et al.* [33] bears the most similarity to the current work.

The main contribution of this paper is the use of the Extended Kalman Filter to solve the constrained inverse kine-

R. F. Souto is a M.Sc. candidate at School of Electrical Engineering, State University of Campinas - UNICAMP, Campinas, SP - Brazil. e-mail: rfsouto@dt.fee.unicamp.br.

G. A. Borges and A. R. S. Romariz are with the Electrical Engineering Department, University of Brasília, Caixa Postal 4386, CEP 70919-970, Brasília, DF - Brazil. e-mails: gaborges@ene.unb.br, romariz@ene.unb.br.

978-1-4244-3804-4/09/$25.00 ©2009 IEEE

matic problem during gait generation. This filter, commonly used to state estimation, is applied as an optimizer. In this sense, the computation of the joint variables is made by filter measures and the supporting leg constraints are explicitly considered as pseudo-measures.

This paper is organized as follows. In section II, we describe the kinematic model and the structure of the robot, stressing our notation and the convention adopted. In section III, it is presented the gait generator. In section IV, it is proposed the optmizer using the Extended Kalman filter. Simulation and experimental results are discussed in sections V and VI, respectively. At last, conclusions are drawn in section VII.

II. KINEMATIC MODELLING OF THE QUADRUPED ROBOT

In order to derive the kinematic and differential kinematic models, we model each leg separately and then the whole body. In the Forward Geometric Model (FGM) method, each leg was considered a robotic manipulator with three links, connected through rotational joints and one effector, thus forming the kinematic chain. The end effector corresponds to the final piece of the chain, in other words, the foot. The other end (base) corresponds to the shoulder, and connects the i-th leg to the robot body ($i = 1, \ldots, 4$). Axis orientation of each coordinate system follows the Denavit-Hartenberg convention, as shown in Figs. 1 and 2.

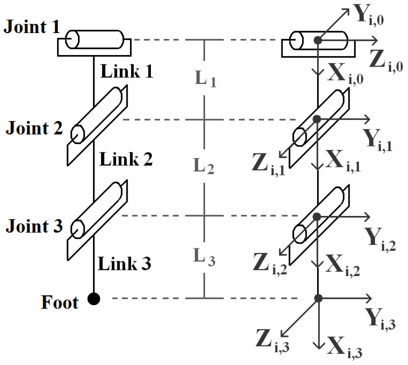

Fig. 1. Denavit-Hartenberg convention for the i-th leg.

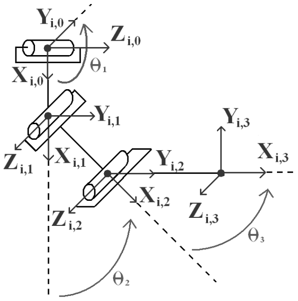

Fig. 2. Convention for the joint variables for the i-th leg.

The set of variables representing the joints of the i-th leg is written as

$$\mathbf{q}_i = [\ q_{i,1}\ q_{i,2}\ q_{i,3}\ ]^T = [\ \theta_{i,1}\ \theta_{i,2}\ \theta_{i,3}\ ]^T, \tag{1}$$

where $\theta_{i,n}$, $n = 1,2,3$, represents the actuation angle of the n-th joint on the base. Thus, the platform has a total of 12 actuation degrees of freedom.

Position and orientation of a foot relative to the shoulder is then represented through the following homogeneous transformation matrix $\mathbf{H}_3^0(\mathbf{q}_i)$:

$$\mathbf{H}_3^0 = \mathbf{H}_{i,1}(q_{i,1})\mathbf{H}_{i,2}(q_{i,2})\mathbf{H}_{i,3}(q_{i,3}), \tag{2}$$

$$\mathbf{H}_{i,n}(q_{i,n}) = \begin{bmatrix} \mathbf{R}_n^{n-1} & \mathbf{t}_n^{n-1} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}. \tag{3}$$

Matrix $\mathbf{H}_{i,n}$ depends on one joint variable only, $q_{i,n}$, and relates the position and the orientation of the n-th joint coordinate system with the previous $(n-1)$-th joint in the kinematic chain. Submatrix $\mathbf{R}_n^{n-1}$ corresponds to rotation of the n-th joint coordinate system relative to the $(n-1)$-th joint coordinate system, while vector $\mathbf{t}_n^{n-1}$ indicates the relative position between its origins. Not apparent in this simplified notation, the matrix $\mathbf{H}_{i,n}$ also depends on geometric variables of the manipulator, denoted by $L_{i,n}$.

Considering the geometry of the robot under study, we have:

$$\mathbf{H}_{i,1} = \begin{bmatrix} cos(\theta_{i,1}) & 0 & sin(\theta_{i,1}) & L_{i,1}cos(\theta_{i,1}) \\ sin(\theta_{i,1}) & 0 & -cos(\theta_{i,1}) & L_{i,1}sin(\theta_{i,1}) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{H}_{i,2} = \begin{bmatrix} cos(\theta_{i,2}) & -sin(\theta_{i,2}) & 0 & L_{i,2}cos(\theta_{i,2}) \\ sin(\theta_{i,2}) & cos(\theta_{i,2}) & 0 & L_{i,2}sin(\theta_{i,2}) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{H}_{i,3} = \begin{bmatrix} cos(\theta_{i,3}) & -sin(\theta_{i,3}) & 0 & L_{i,3}cos(\theta_{i,3}) \\ sin(\theta_{i,3}) & cos(\theta_{i,3}) & 0 & L_{i,3}sin(\theta_{i,3}) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The FGM provides foot position relative to the shoulder from joint variables, according to

$$\xi_i^o = \mathbf{g}(\mathbf{q}_i, \lambda_i) = \mathbf{H}_3^0\ [0\ 0\ 0\ 1]^T \tag{4}$$

where $\xi_i^o = [\ x_i\ y_i\ z_i\ ]^T$ are the coordinates of the i-th foot extremity and $\lambda_i = [\ L_{i,1}\ L_{i,2}\ L_{i,3}\ ]^T$ describes the physical dimensions of the i-th leg.

To obtain the foot velocity vector from the joint variables, we just need to derive (4), which results in the Forward Kinematic Model (FKM)

$$\dot{\xi}_i^o = \mathbf{J}_i\ \dot{\mathbf{q}}_i, \tag{5}$$

where $\mathbf{J}_i$ is the Jacobian of the i-th leg:

$$\mathbf{J}_i = \frac{\partial \mathbf{g}(\mathbf{q}_i, \lambda_i)}{\partial \mathbf{q}_i}.$$

The Inverse Geometric Problem (IGP), that is, obtaining the joint variables from the foot position, represents a hard analytical problem due to the non-linearities of the matrix $\mathbf{g}(\mathbf{q}_i, \lambda_i)$ relative to $\mathbf{q}_i$. In this work, the joint variables are calculated in an interactive way as described in section IV.

On the other hand, the Inverse Kinematic Problem (IKP) can be derived from the inverse Jacobian, as follows:

$$\dot{\mathbf{q}}_i = (\mathbf{J}_i)^{-1} \dot{\xi}_i^o. \tag{6}$$

This model, the Inverse Kinematic Model (IKM), has numerical issues when $\mathbf{J}_i$ is singular. In this sense, the Kalman filtering methodology proposed in this paper avoids this problem elegantly, as described in section IV too.

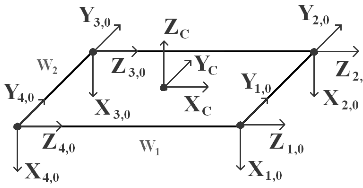

Fig. 3. Coordinate systems for the shoulders and the body center.

In order to simplify the derivation of the robot kinematic model, we adopted a coordinate system in the middle of the body (see Fig. 3). It can be observed that there is just a rotation and a translational operation between the shoulder coordinate system $X_{i,0} \times Y_{i,0} \times Z_{i,0}$ and the robot's $X_c \times Y_c \times Z_c$. Then, the foot position relative to the middle of the robot is given by

$$\xi_i = \mathbf{R}_0^c \xi_i^o + \mathbf{t}_0^c. \tag{7}$$

When we write the four foot coordinate systems relative to the robot coordinate system, we can combine the four matrices ξ_i in a unique matrix $\xi^T = [\xi_1{}^T \; \xi_2{}^T \; \xi_3{}^T \; \xi_4{}^T]$. Similarly, the vector $\mathbf{q}$ has all the robot joint variables, and λ is the correspondent to the geometric parameters. So, the FGM is obtained directly using (4) and the FKM can be described by one equation, as follows:

$$\dot{\xi} = \mathbf{J}\,\dot{\mathbf{q}}, \tag{8}$$

where

$$\mathbf{J} = \frac{\partial \mathbf{g}(\mathbf{q}, \lambda)}{\partial \mathbf{q}} = \begin{bmatrix} \mathbf{J}_1 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{J}_2 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{J}_3 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{J}_4 \end{bmatrix}$$

Naturally, this model does not consider the constraints imposed by the supporting legs. This issue will be considered more specifically in next sections.

III. GAIT GENERATOR

The gait generator provides at each discrete time k a vector ξ_k^* of the desired extremity coordinates of the four legs. In order to do this, the gait generator distinguishes between supporting and swing legs. Supporting legs are in contact with ground and are responsible to provide traction to the robot. Swing legs move in the air following a sinusoid-form trajectory in order to reach a given contact point in the ground. Hence, it becomes important to include into the model an operator identifying the supporting and the swing legs. Denote $\mathbf{E}$ a diagonal matrix

$$\mathbf{E}_k = \begin{pmatrix} e_1 & 0 & 0 & 0 \\ 0 & e_2 & 0 & 0 \\ 0 & 0 & e_3 & 0 \\ 0 & 0 & 0 & e_4 \end{pmatrix},$$

with $e_i = 1$ if the i-th leg is a supporting leg, or $e_i = 0$ for swing leg.

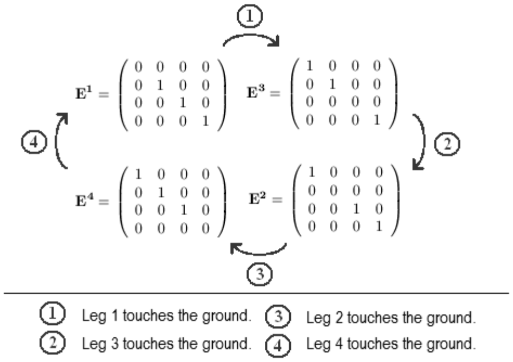
①
$\mathbf{E}^1 = \begin{pmatrix} 0&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{pmatrix}$
$\mathbf{E}^3 = \begin{pmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&0&0 \\ 0&0&0&1 \end{pmatrix}$
④
②
$\mathbf{E}^4 = \begin{pmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&0 \end{pmatrix}$
$\mathbf{E}^2 = \begin{pmatrix} 1&0&0&0 \\ 0&0&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{pmatrix}$
③
① Leg 1 touches the ground.
③ Leg 2 touches the ground.
② Leg 3 touches the ground.
④ Leg 4 touches the ground.

Fig. 4. Change of $\mathbf{E}$ matrix according to walk triggering events.

Indeed, $\mathbf{E}$ is a time varying, event-driven matrix whose entries indicate the condition of each leg. In the model, it should change according to events associated to the touch of the legs in the ground. These events can be triggered by touch sensors installed in the extremity of each leg, allowing an automatic update of $\mathbf{E}$ matrix in closed-loop. On the other hand, open-loop update of $\mathbf{E}$ matrix can also be done. In this case, the events are generated by the algorithm which generates the trajectory of the swing legs. For instance, a walking pattern can be generated by the automata shown in Fig. 4. In this figure, $\mathbf{E}^j$, $j = 1, \ldots, 4$, represents the values to be used in $\mathbf{E}$ for the j-th phase of a walking cycle. The triggering events are also shown in the figure. $\mathbf{E}$ matrix is useful in the Kalman filter-based optimizer, presented in section IV.

The gait generator computes ξ_k^* according to the situation of each leg and the type of trajectory command the robot is performing. For instance, for a "go ahead" trajectory command, the gait generator follows as below:

- For the swing leg(s), ξ_k^* corresponds to a sinusoid-like trajectory. Consider d the distance between the start (leg leaving the ground) and the end (leg arriving on the ground) contact points of the movement, the height of the trajectory (sinusoid amplitude) is $d/5$;
- For the supporting legs, ξ_k^* corresponds to a straight line trajectory of length $\frac{d}{3}$ in the opposite sense of motion of the robot.

The main advantage of using a parameterized gait generator is that its parameters can evolve according to desired speed of the robot. Fig. 5 shows a sequence for $\mathbf{E}$ for leg 1. The other legs present a similar sequence, but in a dephased order, according to triggering events.

IV. THE MOTION OPTIMIZER

The main task of the motion optimizer consists in establishing the joint angles of each leg in a way that its extremity will be at ξ_k^*. Indeed, the optimizer employs a simple rule to compute the configuration $\mathbf{q}_k$ as

$$\mathbf{q}_k = \mathbf{q}_{k-1} + \delta_{\mathbf{q}_k} \tag{9}$$

where $\delta_{\mathbf{q}_k}$ is the configuration change also computed by the optimizer. This is achieved by minimizing the cost function

$$V(\mathbf{q}_k) = \left\| \xi_k^* - \mathbf{g}(\mathbf{q}_k, \lambda) \right\|^2 \tag{10}$$

constrained by

$$\mathbf{h}(\mathbf{q}_k) = \mathbf{0}. \tag{11}$$

The constraint is associated to legs which are in contact to ground. Indeed, it is desired that the body of the robot keeps a given configuration with the ground, despite the movement of the legs. For instance, the coordinates (x_i, y_i, z_i) of the supporting legs extremities should be in the same plane with respect to robot reference frame. This can be represented by a plane equation:

$$ax_i + by_i + cz_i = 1 \tag{12}$$

where a, b and c are parameters of the plane, being the same for the supporting legs. These parameters can even be time varying. Since this constraint is to be respected only by the legs with foot in contact with ground, indicated by $\mathbf{E}$ matrix, $\mathbf{h}(\mathbf{q}_k)$ is given by

$$\mathbf{h}(\mathbf{q}_k) = \mathbf{E}\left(\mathbf{W}\ \mathbf{g}(\mathbf{q}_k, \lambda) - \mathbf{1}_{4\times 1}\right), \tag{13}$$

where

$$\mathbf{W} = \begin{pmatrix} [\ a\ b\ c\ ] & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \\ \mathbf{0}_{1\times 3} & [\ a\ b\ c\ ] & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & [\ a\ b\ c\ ] & \mathbf{0}_{1\times 3} \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0} & [\ a\ b\ c\ ] \end{pmatrix}$$

Classical optimizers can be used to solve the problem of minimizing (10) under the constraint (11). However, such optimizers are hard to code and need to present feasible results withing a sampling step. The main proposition of this work is to use a Kalman filter, commonly used in estimation problems of stochastic state variables [34]. In this case, it is used as an optimizer. The main strength of using this algorithm is that its parameters are easy to adjust. Consider the following state-space stochastic model:

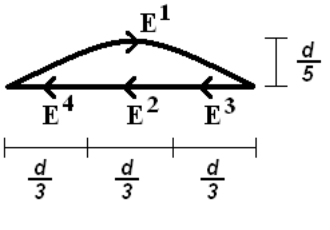

Fig. 5. The four phases in a cycle of leg 1.

$$\begin{cases} \mathbf{q}_k = \mathbf{q}_{k-1} + \mathbf{w}_k & \text{(process model)} \\ \xi_k = \mathbf{g}(\mathbf{q}_k, \lambda) + \mathbf{v}_k & \text{(measurement model)} \end{cases} \tag{14}$$

where $\mathbf{w}_k \sim N(0, \sigma_w^2 \mathbf{I})$ and $\mathbf{v}_k \sim N(0, \sigma_v^2 \mathbf{I})$ are process and measurement noises, respectively. The process model corresponding to the assumption that $\mathbf{q}_k$ changes are Gaussian variables, with variance σ_w^2. The bigger σ_w^2, the larger can be the change from $\mathbf{q}_{k-1}$ to $\mathbf{q}_k$. The measurement model represents the coordinates of the extremities of the legs according to configuration $\mathbf{q}_k$. Therefore, it is the FGM.

The above model has only mathematical meaning. The use of the Kalman filter for estimating $\mathbf{q}_k$ according to the above model is equivalent to applying the unconstrained least squares normalized gain algorithm [35]. The filter equations are computed in two steps:

Prediction step:

$$\hat{\mathbf{q}}_{k|k-1} = \hat{\mathbf{q}}_{k-1} \tag{15}$$

$$\mathbf{P}_{k|k-1} = \mathbf{P}_{k-1} + \sigma_w^2 \mathbf{I} \tag{16}$$

Correction step:

$$\tilde{\mathbf{G}}_k = \mathbf{P}_{k|k-1} \mathbf{J}_{k-1}^T \left(\mathbf{J}_{k-1} \mathbf{P}_{k-1} \mathbf{J}_{k-1}^T + \sigma_v^2 \mathbf{I} \right)^{-1}$$

$$\tilde{\mathbf{q}}_k = \hat{\mathbf{q}}_{k|k-1} + \tilde{\mathbf{G}}_k \left(\xi_k^* - \mathbf{g}(\hat{\mathbf{q}}_{k|k-1}, \lambda) \right)$$

$$\tilde{\mathbf{P}}_k = \left(\mathbf{I} - \tilde{\mathbf{G}}_k \mathbf{J}_k^T \right) \mathbf{P}_{k|k-1}$$

It should be pointed out that $\tilde{\mathbf{G}}_k$ is computed in the same way as *damped least squares*, which minimizes the effects of singular $\mathbf{J}_{k-1}^T$. In this case, the damping parameter is given by the model measurement noise variance σ_v^2. This allows more intuitive adjusting of the damping parameter, since it is directly related to the minimum of $V(\tilde{\mathbf{q}}_k)$. The smaller this parameter, the smaller will be $V(\tilde{\mathbf{q}}_k)$. Finally, $\tilde{\mathbf{q}}_k$ is an estimate to $\mathbf{q}_k^* = \mathbf{g}^{-1}(\xi_k^*, \lambda)$ which does not consider the constraint (11). In order to consider the constraint, the pseudo-measurement concept is used. This concept allows an elegant solution to constrained estimation using stochastic filters. Hence, the pseudo-measurement model is given by

$$\mathbf{0} = \mathbf{h}(\mathbf{q}_k) + \mathbf{u}_k \tag{17}$$

where $\mathbf{u}_k \sim N(0, \sigma_u^2 \mathbf{I})$ represents the pseudo-measurement noise.

Considering the pseudo-measurement model, a new correction is applied to $\tilde{\mathbf{q}}_k$:

$$\mathbf{A}_k = \frac{\partial \mathbf{h}(\tilde{\mathbf{q}}_k)}{\partial \tilde{\mathbf{q}}_k} \tag{18}$$

$$\mathbf{G}_k = \tilde{\mathbf{P}}_k \mathbf{A}_k^T \left(\mathbf{A}_k \tilde{\mathbf{P}}_k \mathbf{A}_k^T + \sigma_u^2 \mathbf{I} \right)^{-1} \tag{19}$$

$$\hat{\mathbf{q}}_k = \tilde{\mathbf{q}}_k - \mathbf{G}_k \mathbf{h}(\tilde{\mathbf{q}}_k) \tag{20}$$

$$\mathbf{P}_k = \left(\mathbf{I} - \mathbf{G}_k \mathbf{A}_k^T \right) \tilde{\mathbf{P}}_k \tag{21}$$

If $\sigma_u^2 = 0$, the constrained is completely satisfied. For $\sigma_u^2 > 0$, it acts as a damping factor, and the constrained is not satisfied, but $\hat{\mathbf{q}}_k$ is closer to the constraint manifold than

$\tilde{\mathbf{q}}_k$. As a result, $\hat{\mathbf{q}}_k$ is an estimate of $\mathbf{q}_k^* = \mathbf{g}^{-1}(\xi_k^*, \lambda)$ and considers the constraints, and should be used as reference by the joint controllers of the robot.

V. SIMULATION RESULTS

The proposed algorithm is first evaluated in a simulation using Matlab. The simulation parameters were adjusted according to the real dimensions of our experimental platform (see Table I). Links $L_{i,n}$, widht W_1 and W_2 are defined in Fig. 1 and Fig. 3. For the forward motion of the robot, it is considered a slope terrain. Fig. 6 shows an screen of the simulation with the trajectory generated for all legs. In Fig. 7, it can be seen in detail the supporting leg keeping contact with the ground although the target points provided by the trajectory generator were in a straight line above the ground. This occurs because the optimizer makes correction in the straight line trajectory in order to satisfy the body configuration constraint (11). Indeed, it is a compromise between values of σ_u^2 and σ_v^2 whether the final leg configuration will be close to the desired trajectory or to the constraint manifold. The effect of σ_w^2 is only on how far $\hat{\mathbf{q}}_k$ will be from $\hat{\mathbf{q}}_{k-1}$. A similar effect has been verified in the swing leg, as seen in Fig. 8.

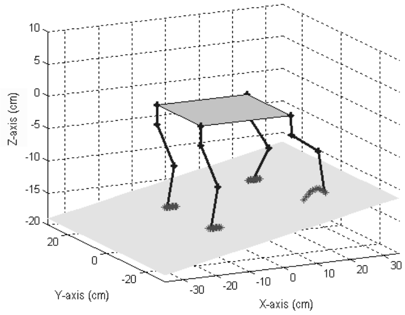

Fig. 6. Screen of the simulator in the slope simulation.

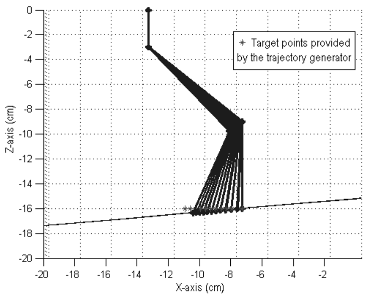

Fig. 7. Trajectory change of the foot of the supporting leg.

VI. EXPERIMENTAL RESULTS

The presented system was experimentally evaluated in a robot whose physical parameters match the ones used in the modelling presented above. The three DOF for each leg were provided by three HITEC HS-755HB servomotors. Each servo provides a maximum torque of 11 $kg \cdot cm$, and is capable of turning 60° in 0.28s. Its position reference is obtained through a 50 Hz PWM signal. A local controller board generates the PWM signals to the three servos of each leg. A PC is connected to the four local controllers using RS-485 bus through its serial port. All calculation is performed in real time by the PC, and data is sent to the local controllers at the end of each calculation cycle, at 115200 BAUD rate. The calculation cycle time is 50 ms. The resolution of angular positioning in each joint is 0.706°.

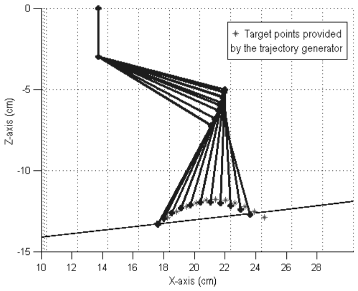

Fig. 8. Foot of the swing leg following the target points.

TABLE I

SIMULATION PARAMETERS

Body	(cm)	Legs	(cm)	Variances	Values
Width W_1	27.0	Links $L_{i,1}$	3.0	σ_w^2	10^{-3}
Width W_2	22.5	Links $L_{i,2}$	8.5	σ_v^2	10^{-8}
		Links $L_{i,3}$	7.0	σ_u^2	10^{-3}

The experiment consists in performing a forward movement at a constant height to the ground. A sequence of movements is shown in Fig. 9. In comparison to simulations, we observe that the body of the robot does not keep a constant distance to the floor, even though its legs move in accordance to the body configuration constraint. This is expected due to the action of gravity force, a factor not present in the kinematic modeling. However, the forward movement was successful, since the swing legs quickly performed their displacement, thus reducing the effect of dynamical imbalance.

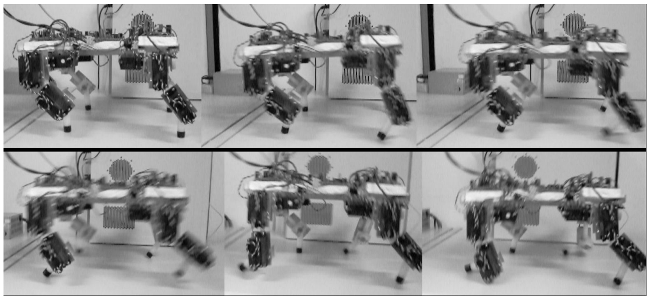

Fig. 9. Snapshots of one swing leg performing its displacement.

VII. CONCLUSIONS AND FUTURE WORKS

In this paper, a gait generator for four-legged robots is obtained through stochastic filtering. It has a simple and compact form, and considers leg and ground contact constraints. The algorithm has been able to solve the inverse kinematic problem in two different situations of the leg motion: the unconstrained case (swing leg) and the constrained case (supporting leg). The Kalman filter efficiency in the solution of the problem has been verified in computer simulations, in which the swing legs follows the desired trajectory, while the legs in contact with the ground obeys the constraints imposed by terrain slope. The current work consists in incorporating a constraint for stability, by considering Zero Moment Point. It will permit to consider gravity into the simulation. Further, new results in hybrid dynamical systems are being considered in this model.

VIII. ACKNOWLEDGMENTS

The authors would like to thank Igor Cardoso and Gauss Batista for the support in experimental evaluation. This work is partially supported by National Council of Technological and Scientific Development - CNPq, Brazil.

REFERENCES

[1] S. P. N. Singh and K. J. Waldron, "A hybrid motion model for aiding state estimation in dynamic quadrupedal locomotion", *in IEEE International Conference on Robotics and Automation*, 2007.

[2] G. Xu, Z. Haojun and Z. Xiuli, "Biologically inspired quadruped robot biosbot: modeling, simulation and experiment", *in Second IEEE International Conference on Autonomous Robots and Agents*, Palmerston North, New Zeland, 2004, pp 261-266.

[3] Y. Fukuoka, H. Kimura and A. H. Cohen, Adaptive Dynamic Walking of a Quadruped Robot on Irregular Terrain Based on Biological Concepts, *The International Journal of Robotics Research*, vol. 22, 2003, pp 187-202.

[4] Z. Liyao, Z. Haojun, Z. Xiuli and C. Zhifeng, "Realization of the biologically-inspired dynamic walking controller of a quadruped robot", *in IEEE Interntional Conference on Robotics, Intelligent Systems and Signal Processing*, Changsha, China, 2003, pp 937-941.

[5] Z. Xiuli, Z. Haojun, G. Xu, C. Zhifeng and Z. Liyao, "A biological inspired quadruped robot: structure and control", *in IEEE International Conference on Robotics and Biomimetics*, 2005, pp 387-392.

[6] K. Tsujita, T. Kobayashi, T. Inoura and T. Masuda, "A study on adaptive gait transition of quadruped locomotion", *in SICE Annual Conference*, Japan, 2008, pp 2489-2494.

[7] L. Righetti and A. J. Ijspeert, "Pattern generators with sensory feedback for the control of quadruped locomotion", *in IEEE International Conference on Robotics and Automation*, Pasadena, CA, USA, 2008, pp 819-824.

[8] J. Z. Kolter, M. P. Rodgers and A. Y. Ng, "A control architecture for quadruped locomotion over rough terrain", *in IEEE International Conference on Robotics and Automation*, Pasadena, CA, USA, 2008, pp 811-818.

[9] S. Aoshima, F. Sato and M. Shiraishi, "Master-slave quadruped walking robot", *in IEEE International Workshop on Robot and Human Communication*, 1997, pp 148-153.

[10] I. M. Koo, T. D. Trong, T. H. Kang, G. Vo, Y. K. Song, C. M. Lee and H. R. Choi, "Control of a quadruped walking robot based on biologically inspired approach", *in IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, 2007, pp 2969-2974.

[11] S. Takao, z. Gu, T. Ikeda and T. Mita, "Realization of dynamic walking and running of a cat type quadruped robot using variable constraint control", *in SICE Annual Conference*, Japan, 2003, pp 3053-3058.

[12] P. K. Pal and K. Jayarajan, Generation of Free Gait - A Graph Search Approach, *IEEE Transactions on Robotics and Automation*, vol. 7, 1991, pp 299-305.

[13] D. J. Pack and H. Kang, "An omnidirectional gait control using a graph search method for a quadruped walking robot", *in IEEE International Conference on Robotics and Automation*, 1995, pp 988-993.

[14] L. Sun, M. Q. H. Meng, W. Chen, H. Liang and T. Mei, "Design of quadruped robot based CPG and fuzzy neural network", *in IEEE International Conference on Automation and Logistics*, Jinan, China, 2007, pp 2403-2408.

[15] L. R. Palmer III and D. E. Orin, "Quadrupedal running at high speed over uneven terrain", *in IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, 2007, pp 303-308.

[16] D. W. Marhefka and D. E. Orin, "Fuzzy control of quadrupedal running", *in IEEE International Conference on Robotics and Automation*, San Francisco, CA, USA, 2000, pp 3063-3069.

[17] X. Chen, K. Watanabe, K. Kiguchi and K. Izumi, An ART-Based Fuzzy Controller for the Adaptive Navigation of a Quadruped Robot, *IEEE/ASME Transactions on Mechatronics*, vol. 7, 2002, pp 318-328.

[18] S. Bai, K. H. Low, G. Seet and T. Zielinska, "A new free gait generation for quadrupeds based on primary/secondary gait", *in IEEE International Conference on Robotics and Automation*, Detroit, Michigan, USA, 1999, pp 1371-1376.

[19] J. Estremera and P. G. Santos, Generating Continuous Free Crab Gaits for Quadruped Robots on Irregular Terrain, *IEEE Transactions on Robotics*, vol. 21, 2005, pp 1067-1076.

[20] J. R. Rebula, P. D. Neuhaus, B. V. Bonnlander, M. J. Johnson, J. E. Pratt, "A controller for the LittleDog quadruped walking on rough terrain", *in IEEE International Conference on Robotics and Automation*, Rome, Italy, 2007, pp 1467-1473.

[21] D. Papadopoulos and M. Buehler, "Stable running in a quadruped robot with compliant legs", *in IEEE International Conference on Robotics and Automation*, San Francisco, CA, USA, 2000, pp 444-449.

[22] S. Meek, J. Kim, and M. Anderson, "Stability of a trotting quadruped robot with passive, underactuated legs", *in IEEE International Conference on Robotics and Automation*, Pasadena, CA, USA, 2008, pp 347-351.

[23] J. Yang, Two-phase Discontinuous Gaits for Quadruped Walking Machines with a Failed Leg, *Robotics and Autonomous Systems*, vol. 56, 2008, pp 728-737.

[24] M. A. Jimenez, P. G. Santos, Attitude and Position Control Method for Realistic Legged Vehicles, *Robotics and Autonomous Systems*, vol. 18, 1996, pp 345-354.

[25] M. A. Jimenez, P. G. Santos, Attitude and Position Control for Non-rigid Walking Machines, *Mech. Mach. Theory*, vol. 33, 1998, pp 1013-1029.

[26] M. Y. Al-Zaydi and S. H. M. Amin, "Simulation kinematics model of a multi-legged mobile robot", *in IEEE International Conference on Advanced Robotics*, 1997.

[27] A. Gasparetto, R. Vidoni and T. Seidl, "Kinematic study of the spider system in a biomimetic perspective", *in IEEE International Conference on Intelligent Robots and Systems*, Nice, France, 2008, pp 3077-3082.

[28] E. Oyama and S. Tachi, "Modular neural net system for inverse kinematics learning", *in IEEE International Conference on Robotics and Automation*, San Francisco, CA, USA, 2000, pp 3239-3246.

[29] E. Oyama, N. Y. Chong, A. Agah, T. Maeda and S. Tachi, "Inverse kinematics learning by modular architecture neural networks with performance prediction networks", *in IEEE International Conference on Intelligent Robots and Systems*, Seoul, Korea, 2001, pp 1006-1012.

[30] N. Kohl and P. Stone, "Policy gradient reinforcement learning for fast quadrupedal locomotion", *in IEEE International Conference on Robotics and Automation*, New Orleans, LA, USA, 2004, pp 2619-2624.

[31] M. S. Erden, K. Leblebicioglu, Free Gait Generation with Reinforcement Learning for a Six-Legged Robot, *Robotics and Autonomous Systems*, vol. 56, 2008, pp 199-212.

[32] A. N. Pechev, "Inverse kinematics without matrix inversion", in IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 2008, pp 2005-2012.

[33] P. Lin, H. Komsuoglu and D. E. Koditschek, Sensor Data Fusion for Body State Estimation in a Hexapod Robot With Dynamical Gaits, *IEEE Transactions on Robotics*, vol. 22, 2006, pp 932-943.

[34] R.E. Kalman, A new approach to linear filtering and prediction problems, Transactions of the ASME, Journal of Basic Engineering, vol. 82, 1960, pp 35-45.

[35] L. Ljung, System Identification - Theory for the User, Prentice-Hall, 1999.