Modeling and Robust Iterative Learning Control of a Quadruped Robot

M. Hosseinzadeh Yamchi*, M. Shafiee*, H. A. Talebi*

* Amirkabir University of Technology, Tehran, Iran
(e-mail: mhosseinzade@aut.ac.ir, mshafie@aut.ac.ir
alit@aut.ac.ir, Tel: +98-021-64543340).

Abstract: In this paper, the problem of modeling and control of a planar eight-degrees of freedom quadruped robot is investigated. First a new methodology for dynamic modeling of quadruped robot introduced based on the tree structure of the quadruped robot and using the Euler-Lagrange method. Then, a new control scheme is proposed based on the presented model. The control scheme is composed of a robust control term and an iterative learning controller. The robust controller is designed based on the Lyapunov theorem to overcome the degrading effect of the disturbances and uncertainties exist in the system model. Whereas the iterative learning strategy accounts for enhancing the performance of the quadruped robot. The effectiveness of the proposed approach is demonstrated via simulations performed on a obtained model for a eight-link quadruped.

1. INTRODUCTION

The problem of modeling and control of legged locomotion systems has recently received increased attention due to their higher mobility than conventional wheeled vehicles. Although wheeled robots are very popular, they suffer from several limitations that reduce their efficiency. For instance, they can reliably navigate only in some limited types of terrain. In contrast, legged robots provide great flexibility in choosing the type of the terrain they can proceed.

Quadruped robots are one of the important types of legged robots. Due to the importance and practical application of quadrupeds, they have been attracted researchers and scientists in recent years. Similar to the biped robots [1], [2], [10] quadrupeds have many theoretical and practical challenges such as nonlinear dynamics, over-actuation and control in a given time that researchers try to tackle them. Controlling and maintaining the stability of quadruped robots during its walking is one of the challenging problems in robotics [3],[4],[5]. Although the problem of gait generation and control of the quadruped robots have been considered increasingly by many researchers[11],[12], but the problem of modeling of the quadrupeds is not considered so much [13].

Quadruped robots have multiple closed chains in their structure and it makes the modeling problem of these robots very difficult. These closed chains impose constraints on the dynamic equations of motion. Depending on the walking phase of the quadruped, number of closed chains and therefore number of constraints vary.

The control problem of quadruped is also of great importance. Legged robots and specifically quadrupeds are used in outdoor environment that usually contain rough terrains and so disturbances, noises, foot slippage and ... are inevitable. To overcome these problems robust scheme should be considered.

Lei sun et al. [6] considered control problem of the quadruped robot under full support phase and simplified it as the problem of the tracking control of the trunk motion. They proposed a robust scheme to achieve the tracking goal. Similar idea is used by Aghabalie et al. [2] to control biped motion.

In this paper the problem of modeling and control of quadruped motion is considered. First the mathematical model of the planar quadruped robot with eight degrees of freedom (two in each leg) is obtained using the Euler-Lagrange method. Then control of the quadruped is considered and a new control scheme is proposed. These scheme deals with the tracking problem and robust control of the quadruped motion and in addition a learning term in the controller is considered that enhances the tracking and robustness of the system. The validity of the proposed schemes has been shown by simulation results. The rest of the paper is organized as follows: Dynamic modeling of the quadruped is presented in section II. In Section III control schemes is presented, trajectory generation is presented in Section IV and Section V provides simulation results. Finally the paper is concluded in Section VI.

2. DYNAMIC MODELING OF QUADRUPED

It is well known that a quadruped gait consists of several periodic phases. Hence, depending on the number of legs of quadruped contacting the ground, the constraints are imposed to the system will be different and so do dynamic equations of motion in each phase.

The schematic of the quadruped is shown in Fig.1. In this paper, we consider a planar eight degrees of freedom quadruped with two degrees of freedom in each leg. Due to
the complexity of the modeling problem, we consider several simplifying assumptions:

i. The quadruped has only planar motion

ii. The legs are rigid and have point mass

iii. The joints are frictionless

iv. The toes do not slip on the ground

First, we present the concept of tree structure for the modeling of closed-chain robots and then the model of the quadruped in two independent phases will be derived.

2.1 The tree structure for modeling closed-chain robots

Suppose the robotic system is composed of L joints and n links with N active joints and the other L - N joints being passive. Then, the number of independent closed loops B is equal to L - N + 1. We assume that the number of actuated joints is equal to the number of degrees of freedom of the mechanism. Equivalent tree structure is constructed by virtually cutting each loop at one of its passive joints [9]. Dynamic equations of equivalent tree structure are obtained by Euler-Lagrange method as indicated in the following equation:

\[ M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = \tau + d \quad (1) \]

Where M is the inertia matrix, C represents the centrifugal-Coriolis matrix and G represents the gravity effect. Dynamic equation of the closed-chain robot will be as follows:

\[ M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = \tau + \left( \dfrac{\partial\Phi(\theta)}{\partial\theta} \right)^T \lambda \quad (2) \]

\[ \Phi(\theta) = 0 \] is the vector containing independent constraint equations and \( \lambda = [\lambda_1, \ldots, \lambda_n] \) is the Lagrange multiplier vector. In the sequel the dynamic equations of the quadruped robot is obtained.

2.2 Dynamic model of the quadruped

In this phase all legs of the quadruped are in contact with the ground. In this case, the contacting points of the legs can be considered as passive joints. There are 3 independent closed-loops in this case and the closed-loops can be cut at the contacting points. Now by considering one of the legs as the reference and then cutting three loops in contacting points, we will have a system that do not contain closed-loops and is an open loop mechanism. The dynamic equation of the open-loop system is obtained using the Euler-Lagrange method as follows:

\[ M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = \tau \quad (3) \]

where it is not possible to present matrices M, C and G because of the limit in the page numbers. The constraint equations for the full support phase can be written as follows

\[ \Phi = \begin{bmatrix} l_1c_1 + l_2c_2 + l_3c_3 + l_4c_4 + l_5c_5 - L_{12} \\ l_1s_1 + l_2s_2 + l_3s_3 + l_4s_4 + l_5s_5 \\ l_1c_1 + l_2c_2 + l_3c_3 + l_4c_4 + l_5c_5 - L_{13} \\ l_1s_1 + l_2s_2 + l_3s_3 + l_4s_4 + l_5s_5 \\ l_1c_1 + l_2c_2 + l_3c_3 + l_4c_4 + l_5c_5 - L_{14} \\ l_1s_1 + l_2s_2 + l_3s_3 + l_4s_4 + l_5s_5 \end{bmatrix} = 0 \quad (4) \]

where \( c_i \) and \( s_i \) stand for \( \cos(\theta_i) \) and \( \sin(\theta_i) \) respectively, \( L_i \) is the horizontal distance between leg number \( i \) and \( J \). Finally, the dynamic equations of the quadruped in full support phase in the presence of disturbance will be as:

\[ M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = \tau + J^T \lambda + d \quad (5) \]

where

\[ J = \dfrac{\partial\Phi(\theta)}{\partial\theta} \quad (6) \]

**Remark 1** - The dynamic equations of the quadruped in other possible configurations will be similar to that of Equation (5) with the same Inertia, Centrifugal-Coriolis and the gravity matrix but with different constraint equations and hence different Jacobian matrix. For example, if Leg 2 in Fig. 1 be in the swing phase and other three legs be in contact with the ground, constraint equations could be written as:

\[ \Phi = \begin{bmatrix} l_2c_2 + l_3c_3 + l_4c_4 + l_5c_5 - L_{13} \\ l_2s_2 + l_3s_3 + l_4s_4 + l_5s_5 \\ l_2c_2 + l_3c_3 + l_4c_4 + l_5c_5 - L_{14} \\ l_2s_2 + l_3s_3 + l_4s_4 + l_5s_5 \end{bmatrix} = 0 \quad (7) \]

and the dynamic equations of the quadruped robot in this phase will be as (5) with Jacobian matrix specified using (6) and (7).

3. PROPOSED CONTROL SCHEME

The robust control design for quadruped robot is known to be a challenging problem. Most of the works however are based on simplifying assumptions in order to reduce the dynamic equations and ignore holonomic constraints. Moreover, we also model holonomic constraints. This is valid if the constraint equations in (4), are always satisfied, i.e. all of the feet of the quadruped maintain firm contact on the ground without slipping and lifting during the corresponding phase.

Since this condition is not always satisfied in practice, the constraint forces due to slipping, rough terrain, etc can be considered as an additive disturbance term acting on the robot's dynamic equation. Our robust control methodology is then required to reject such disturbances which make the quadruped robot able to move on variety of terrains.
Control signal is consist of two terms, a nonlinear robust controller and a learning controller:
\[ u = u_i + u_l \]
where \( u_i \) is the robust controller and \( u_l \) is the learning controller. First we present robust controller and then an iterative learning scheme is considered to improve the tracking performance and robustness against disturbances and uncertainties.

### 3.1 Robust Controller

Let \( \theta_d, \dot{\theta}_d \) and \( \ddot{\theta}_d \) denote the desired trajectory, and \( e = \theta - \theta_d, \dot{e} = \dot{\theta} - \dot{\theta}_d \) and \( \ddot{e} = \ddot{\theta} - \ddot{\theta}_d \) are position, velocity and acceleration errors, respectively. In the following evaluation signal describes the robust performance index:
\[ z = [p_e e, p_{\dot{e}} \dot{e}] \]

where \( p_e \) and \( p_{\dot{e}} \) are some positive scalars. The nonlinear compensator is given by:
\[ \tau = M(\theta)\ddot{\theta}_d + C(\theta, \dot{\theta})\dot{\theta}_d - J^T \lambda + u \]

where we suppose here that \( \lambda \) is obtained from sensor. Now by substituting the above controller in (5), one can get
\[ M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = d + u \]

denote \( x_i = e, x_j = e + \dot{e} \). Then above equation is transformed to the following standard state space representation:
\[ \begin{cases} \dot{x}_j = x_2 - x_j \\ \dot{x}_2 = \left[ M(\theta)^{-1} C(\theta, \dot{\theta}) - I \right] x_j + \left( I - M(\theta)^{-1} C(\theta, \dot{\theta}) \right) x_2 \\
+ M(\theta)^{-1} (u - G(\theta) + d) \end{cases} \]  

(8)

To study the stability of closed loop system (8), the following Lyapunov function candidate is considered:
\[ V(x_i, x_j) = \frac{x_j^T x_j + x_i^T M(\theta) x_i}{2} \]

Evaluating the derivative of \( V \) along the trajectories of the system (8) results in:
\[ \dot{V}(x_i, x_j) = x_j^T \dot{x}_j + x_i^T M(\theta) \dot{x}_j + \frac{1}{2} x_j^T M(\theta) x_j \]
\[ = x_j^T (\dot{x}_j - x_j) + x_i^T \left( \left( C(\theta, \dot{\theta}) - M(\theta) \right) x_j + M(\theta) x_i \right) \\
+ u - G(\theta) + d + \frac{1}{2} x_j^T \left( M(\theta) - 2C(\theta, \dot{\theta}) \right) x_j \]

\[ m \]

the most important property of robot dynamic equations, i.e. the so called skew symmetric property, we have
\[ x_i^T \left( M(\theta) - 2C(\theta, \dot{\theta}) \right) x_i = 0 \]

which result in
\[ \dot{V}(x_i, x_j) = -x_j^T x_j + x_i^T \left( \left( C(\theta, \dot{\theta}) - M(\theta) + I \right) x_j \right. \\
+ M(\theta) x_j + u - G(\theta) + d \]

From robust control theory we know that if \( \dot{V} \leq -\left\| x \right\|^2 + \gamma^2 \left\| \dot{x} \right\|^2 \), then \( \dot{e} = e = 0 \) is robust asymptotically stable equilibrium point, and the following \( L_2 \) gain is satisfied:
\[ \int_0^T \left\| e \right\|^2 dt \leq 2 \gamma \int_0^T \left\| \dot{e} \right\|^2 dt \]

So for the stability we should have:
\[ -x_j^T x_j + x_i^T \left( \left( C(\theta, \dot{\theta}) - M(\theta) + I \right) x_j + M(\theta) x_j \right) \]
\[ + u - G(\theta) + d \leq -p_i^2 \left\| x_j \right\|^2 + p_j^2 \left\| x_i \right\|^2 + \gamma^2 \left\| \dot{e} \right\|^2 \]

or
\[ -x_j^T x_j + x_i^T \left( \left( C(\theta, \dot{\theta}) - M(\theta) + I \right) x_j + M(\theta) x_j \right) \]
\[ + u - G(\theta) + d \leq p_i^2 \left\| x_j \right\|^2 + p_j^2 \left\| x_i \right\|^2 + \gamma^2 \left\| \dot{e} \right\|^2 \leq 0 \]  

(9)

From above statements, it can be easily seen that the robust control input can be chosen as:
\[ u = -\left( C(\theta, \dot{\theta}) - M(\theta) + I \right) x_j - M(\theta) x_j \]
\[ + G(\theta) + 2p_i x_j - k x_j \]

(10)

substituting (10) in (9) results
\[ -x_j^T x_j + x_i^T \left( 2p_i^2 x_j - k x_j + d \right) + p_i^2 \left\| x_j \right\|^2 \]
\[ + p_j^2 \left\| x_i \right\|^2 + \gamma^2 \left\| \dot{e} \right\|^2 \leq 0 \]

and finally
\[ -(1 - p_i^2 - p_j^2) \left\| x_i \right\|^2 - \left( k - \frac{l}{4\gamma} - p_j^2 \right) \left\| x_j \right\|^2 - 2\gamma^2 \left\| \dot{e} \right\|^2 \leq 0 \]

hence, if we choose \( p_i, p_j \) and \( k \) properly, the stability condition is satisfied.

### 3.2 The Iterative Learning Controller

Iterative learning controller or ILC is a relatively new control technique for improving the transient response and tracking performance of the systems that execute the same trajectory, motion or operation over and over [7]. One of the important specifications of the ILC is that it does not require information about the plant and also ILC can overcome repetitive disturbances.

In the case of the quadruped, we suppose that the quadruped should walk in the specified terrain repeatedly, such as a watchdog that guard from a region. So we can use ILC to learn quadruped walk in specified region.

In this paper we used a simple iterative learning law known as PD type ILC. The learning law is
\[ u_i = u_i + k_p \delta \dot{e}_i + k_v \delta e_i \]

where the index \( k \) is the iteration number. As it will be indicated in the next section, by properly choosing the coefficients \( k_i \) and \( k_v \), the learning controller will help to enhance the performance of the system and as the number of iterations increase.
iterations increased, the perfect tracking is achieved. Also ILC helps the system to maintain it’s performance in the presence of disturbances and uncertainties.

4. TRAJECTORY GENERATION

The trajectory of the quadruped robot used in the paper is generated intuitively. In the full support phase, we have only three degrees of freedom. We considered three trajectories for the three joints and designed these trajectories by considering physical considerations such as link legs, step length and... . Then, we solved the constraint equations (4) and obtained the desired trajectories for remaining joints. However it should be noticed that this paper does not concern with the trajectory generation problem, instead it deal with the tracking problem of the generated trajectories.

5. SIMULATION RESULTS

n this section, Simulation results for an eight-link quadruped robot are presented. We only presented the simulation results under full support phase to demonstrate the effectiveness of the proposed control algorithm. For all simulations, we set \( k = 30, p = 0.8 \), and the learning coefficients as \( k_p = 10, k_d = 6 \). We supposed that the disturbance signal is acting on each joint of the quadruped. We also considered model uncertainties in the dynamics of the quadruped as follows:

\[
\Delta M(\theta) = 0.5M(\theta), \Delta C(\theta, \dot{\theta}) = 0.3C(\theta, \dot{\theta}), \\
\Delta G(\theta) = 0.3G(\theta)
\]

The disturbance signal shown in fig.2 is applied and it is observed that after little iteration the controller can overcome the disturbances and uncertainties and achieve the perfect tracking.

Fig.3 through 10 shows position tracking of the joints in the presence of the disturbance and uncertainties in the dynamic model. Fig.11 shows the control effort in this case and position tracking error is indicated in fig.12 . It is obvious that the performance of the controller in the presence of the disturbances is good and nearly perfect tracking is achieved.
6. CONCLUSION

In this paper, dynamic model of a eight-link quadruped robot under holonomic constraints was obtained by using tree structure of the quadruped and using the Euler-Lagrange method. Moreover, by relaxing simplifying assumptions, a nonlinear robust control strategy using the Lyapunov method in combination with a iterative learning controller was presented. Simulation results demonstrated that the effect of external additive disturbance acting on the robot was significantly reduced using our proposed controller, and all joint variables tracked the reference trajectories.

REFERENCES


Appendix

Using Euler-Lagrange formulation, Inertia matrix, coriolis and centripital matrix and gravity related matrices are obtained as follows.

\[
M' = \begin{bmatrix} m_{ij} \end{bmatrix}_{9x9}
\]

\[
m_{11} = I_{x1} + I_{y1} + I_{z1} + \sum_{i=3}^{9} m_i
\]

\[
m_{12} = m_{12} = m_{13} \cos (\theta_1 - \theta_2) + I_{z2} \sum_{i=3}^{9} m_i \cos (\theta_1 - \theta_2)
\]

\[
m_{22} = m_{22}
\]

\[
m_{33} = m_{33}
\]

\[
m_{ij} = m_{ji}
\]

\[
M = \begin{bmatrix} m_{ij} \end{bmatrix}_{9x9}
\]
\[
m_{12} = m_{13} = \sum_{i=0}^{n} \left( m_{i,13} + m_{i,12} \right) \cos(\theta_1 - \theta) \\
m_{14} = m_{41} = \sum_{i=0}^{n} \left( m_{i,14} + m_{i,41} \right) \cos(\theta_1 - \theta) \\
m_{15} = m_{15} = \sum_{i=0}^{n} \left( m_{i,15} + m_{i,15} \right) \cos(\theta_1 - \theta) \\
m_{16} = m_{16} = \sum_{i=0}^{n} \left( m_{i,16} + m_{i,16} \right) \cos(\theta_1 - \theta) \\
m_{17} = m_{17} = \sum_{i=0}^{n} \left( m_{i,17} + m_{i,17} \right) \cos(\theta_1 - \theta) \\
m_{18} = m_{18} = \sum_{i=0}^{n} \left( m_{i,18} + m_{i,18} \right) \cos(\theta_1 - \theta) \\
m_{19} = m_{19} = \sum_{i=0}^{n} \left( m_{i,19} + m_{i,19} \right) \cos(\theta_1 - \theta) \\
m_{21} = m_{21} = \sum_{i=0}^{n} \left( m_{i,21} + m_{i,21} \right) \cos(\theta_1 - \theta) \\
m_{22} = m_{22} = \sum_{i=0}^{n} \left( m_{i,22} + m_{i,22} \right) \cos(\theta_1 - \theta) \\
m_{23} = m_{23} = \sum_{i=0}^{n} \left( m_{i,23} + m_{i,23} \right) \cos(\theta_1 - \theta) \\
m_{24} = m_{24} = \sum_{i=0}^{n} \left( m_{i,24} + m_{i,24} \right) \cos(\theta_1 - \theta) \\
m_{25} = m_{25} = \sum_{i=0}^{n} \left( m_{i,25} + m_{i,25} \right) \cos(\theta_1 - \theta) \\
m_{26} = m_{26} = \sum_{i=0}^{n} \left( m_{i,26} + m_{i,26} \right) \cos(\theta_1 - \theta) \\
m_{27} = m_{27} = \sum_{i=0}^{n} \left( m_{i,27} + m_{i,27} \right) \cos(\theta_1 - \theta) \\
m_{28} = m_{28} = \sum_{i=0}^{n} \left( m_{i,28} + m_{i,28} \right) \cos(\theta_1 - \theta) \\
m_{29} = m_{29} = \sum_{i=0}^{n} \left( m_{i,29} + m_{i,29} \right) \cos(\theta_1 - \theta) \\
m_{30} = m_{30} = \sum_{i=0}^{n} \left( m_{i,30} + m_{i,30} \right) \cos(\theta_1 - \theta) \\
other m_i \text{ is equal to } 0 \\
C = [c_3] \\
c_{12} = \sum_{i=0}^{n} \left( m_{i,12} + m_{i,12} \right) \sin(\theta_1 - \theta) \\
c_{13} = \sum_{i=0}^{n} \left( m_{i,13} + m_{i,13} \right) \sin(\theta_1 - \theta) \\
c_{14} = \sum_{i=0}^{n} \left( m_{i,14} + m_{i,14} \right) \sin(\theta_1 - \theta) \\
c_{15} = \sum_{i=0}^{n} \left( m_{i,15} + m_{i,15} \right) \sin(\theta_1 - \theta) \\
c_{16} = \sum_{i=0}^{n} \left( m_{i,16} + m_{i,16} \right) \sin(\theta_1 - \theta) \\
c_{17} = \sum_{i=0}^{n} \left( m_{i,17} + m_{i,17} \right) \sin(\theta_1 - \theta) \\
c_{18} = \sum_{i=0}^{n} \left( m_{i,18} + m_{i,18} \right) \sin(\theta_1 - \theta) \\
c_{19} = \sum_{i=0}^{n} \left( m_{i,19} + m_{i,19} \right) \sin(\theta_1 - \theta) \\
c_{21} = \sum_{i=0}^{n} \left( m_{i,21} + m_{i,21} \right) \sin(\theta_1 - \theta) \\
c_{22} = \sum_{i=0}^{n} \left( m_{i,22} + m_{i,22} \right) \sin(\theta_1 - \theta) \\
c_{23} = \sum_{i=0}^{n} \left( m_{i,23} + m_{i,23} \right) \sin(\theta_1 - \theta) \\
c_{24} = \sum_{i=0}^{n} \left( m_{i,24} + m_{i,24} \right) \sin(\theta_1 - \theta) \\
c_{25} = \sum_{i=0}^{n} \left( m_{i,25} + m_{i,25} \right) \sin(\theta_1 - \theta) \\
c_{26} = \sum_{i=0}^{n} \left( m_{i,26} + m_{i,26} \right) \sin(\theta_1 - \theta) \\
c_{27} = \sum_{i=0}^{n} \left( m_{i,27} + m_{i,27} \right) \sin(\theta_1 - \theta) \\
c_{28} = \sum_{i=0}^{n} \left( m_{i,28} + m_{i,28} \right) \sin(\theta_1 - \theta) \\
c_{29} = \sum_{i=0}^{n} \left( m_{i,29} + m_{i,29} \right) \sin(\theta_1 - \theta) \\
c_{30} = \sum_{i=0}^{n} \left( m_{i,30} + m_{i,30} \right) \sin(\theta_1 - \theta) \\
other c_i \text{ is equal to } 0