Proceedings of the 10th
World Congress on Intelligent Control and Automation
July 6-8, 2012, Beijing, China

A Novel Inverse Dynamics Control Strategy with Different Phases for the Quadruped Robot

Bin Li and Yajuan Guo

Jiangsu Electric Power Company Research Institute
Nanjing, Jiangsu Province,211103 , China

lixiebin2003@yahoo.com.cn

Xuesong Shao, Wei Wang and Jianqiang Yi

Institute of Automation
Chinese Academy of Sciences
Beijing, 100190, China

xuesong.shao@gmail.com, {wei.wang & jianqiang.yi}@ia.ac.cn

***Abstract*—Aiming to reduce the computation and implement compliant control, this paper proposes a novel inverse dynamics control strategy based on the floating-base rigid body system. The control strategy assumes that each leg of the quadruped robot organizes itself into an independent autonomous system, a serial robot. Based on this assumption, the kinematics and the dynamics models of the quadruped robot have been created. The dynamical model supposes two different models according to the leg's state. In the stance phase the serial robot affixes its base frame to the shank and iterates the rigid body dynamics algorithms from the knee joint to the body. When the serial robot is in the swing phase, the dynamics algorithm is propagated from the hip joint to the shank, whose computing direction is just the reverse against the direction of which the serial robot is in the stance phase. The quadruped system doesn't need the fixed base to the system and avoids calculating the virtual joints of 6-DOF. Therefore, the algorithm proposed in this paper makes real-time computation of the quadruped robot dynamics possible. In order to evaluate the efficiency of the inverse dynamical control strategy, experiments are accomplished based on a practical quadruped robot. The experiments, which were done on a rubber mat and on asphalt, demonstrated that the quadruped robot is able to walk adaptively.**

Index Terms – quadruped robot; inverse dynamics; floating-base system.

I. Introduction

In the past two decades, lots of investigations have been done focusing on quadruped robots. For example, J. Zico Kolter proposed complete control architectures [1] and Michael Mistry implemented the inverse dynamics control of the floating-base systems [2]. Both models of the kinematics and the dynamics, which relate the world coordinate system to the global environment, are based on the quadruped's body. In fact, the body-based models not only need the perception of the environments, such as the contact of the ground, but also require more attention to the global circumstances such as the placement of the feet in the next steps. Even though Jonas Buchi and Michael Mistry developed an approach that is capable of computing the inverse dynamics algorithm for the floating-base systems without measuring the contact forces [3], there are still lots of complicated works needing to be done in the decomposition of the constraint Jacobin and the calculation for eliminating the contact forces involved in dynamics.

In [4], Marc H. Raibert constructed a planar quadruped robot with springy legs and modeled the legs as an inverted pendulum system with the assumption that the quadruped can be regarded as four serial robots. Through introducing the concept of a virtual leg to use the legs in pairs, such as the trot, the pace and the bound, the control system used the one-legged algorithm based on the inverted pendulum model, a finite state machine and virtual legs to make the quadruped robot run with a trotting gait. In fact, the concept of a virtual leg can be extended to the walking gait that combines with the rigid-body dynamics algorithms.

Roy Featherstone explored a substantial collection of the most efficient algorithms for calculating rigid-body dynamics [25]. Based on the theories of the floating base rigid-body dynamics [5], the representation of the quadruped robot body unattached to the world is created through adding a virtual 6-DOF joint, which connects the fixed base and the floating base, between the base frame affixed to the robot and the world coordinate system affixed to the ground. However, the extra virtual 6-DOF joint adds a lot of new calculations, especially when the quadruped robot is considered as four separate serial robots.

Fig.1 is the picture of the quadruped robot in this research, in which each leg has one hip pitch joint and one knee pitch joint. The prototype of such a pitch-pitch type quadruped robot also possesses one passive compliant prismatic DOF at each toe. Addressing the features of the robot in this research, we propound a novel inverse dynamics control strategy which reduces the computations and implements the compliant control of the quadruped. The control strategy is divided into four parts: the forward kinematics, the inverse kinematics, the inverse dynamics, and the trajectory generation. The forward kinematics imagines the quadruped robot as four virtual serial robots whose base frames are affixed on the trunk. The inverse kinematics is treated as a numerical optimization, which finds the local optimal solution in the joint space according to the characteristics of the quadruped robot locomotion and reconciles the case of multiple solutions and the case of no solution. The inverse dynamics partitions a stride of a cycle as the stance phase and the swing phase. In different phase use

978-1-4673-1398-8/12/$31.00 ©2012 IEEE

different dynamics model, then there is no limitations on the contact forces at the feet and there is no need to calculate the extra virtual 6-DOF joints that is usual in floating-base rigid body system. The trajectory generation considers all the serial robots as a whole robot and integrates the stability of the walking robot into path planning. Through cutting down the calculation of the virtual 6-DOF joint, the inverse dynamics control strategy finds the required vector of joint torques and makes the real-time control possible.

Fig. 1

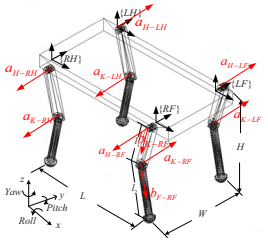


Fig. 2

Fig. 1(a) The quadruped robot with the hip pitch joints and the knee pitch joints;

Fig. 2 The kinematic model of the quadruped robot

This paper is organized as follows. In section Ⅱ the nomenclature and the robot model are introduced, and the robot model in kinematics is expatiated. The inverse dynamics control strategy will be explained in Section Ⅲ. Section Ⅳ gives the experiments and discusses the trajectory curves and the current curves. Then, Section Ⅴ concludes the paper.

II. Robot Model in the Kinematics

The robot shown in Fig. 1 has revolute joints at both the hips and the knees, and passive prismatic joints located in the feet.

A. The Robot Kinematics

Although the inverse dynamics control is widely used in fixed-base manipulator robots, the inverse dynamics application in the quadruped robot has been stymied because of its dependence on the accuracy of dynamics models and the amplified numerical problems that can arise due to matrix inversion [3]. In this research, we simplify the quadruped robot as four serial robots and assume that these robots fix their base frames on the trunk. These serial robots are open kinematic chains which consist of rigid links. The feet are the final links of the chain that make up the four serial robots. (See Fig. 2.)

Fig. 2 shows the skeleton of the kinematic model, on which the base frames $\{RF\}$, $\{RH\}$, $\{LH\}$ and $\{LF\}$ are located at the trunk. As the quadruped robot is divided into four separated parts, the base frames attach themselves to the trunk, which correspond to the four hip joints.

As all the joints are revolute joints, we utilize Rodriguez's rotation formula to compute the rotation matrix from an axis-angle representation [7]. The rotation matrix $e^{(\omega\times 1)t}$ can be expressed by:

$$e^{(\omega\times 1)\theta} = E + (\omega\times 1)\sin\theta + (\omega\times 1)^2(1-\cos\theta) \quad (1)$$

where $\|\omega\|=1$, the notation $\omega\times 1$ stands for the 3×3 cross-product tensor associated with the angular velocity vector ω, which is defined as $(\omega\times 1)v = \omega\times v$, for any Cartesian vector v. Equation (1) converts from angle-axis representation to rotation-matrix representation.

After using Rodriguez's rotation formula to map a rotation matrix into an angular vector, the orientations of the hip frame and the knee frame are represented by the 3×1 unit vectors a_{H-RF} and a_{K-RF}. The directions of these angular vectors are determined by the rotational directions of the relevant motors, which are illustrated in the Fig. 2. The vectors b_{K-RF} and b_{F-RF} are equivalent to the position vectors involving translational frames.

III. The Inverse Dynamics Control Strategy

The flow chart in Fig. 3 illustrates the ordered set of the forward kinematics, the inverse kinematics, the inverse dynamics, and the path generation. Under the skeleton of the forward kinematics, the path generator plans the foot locations in the task space. Then, the inverse kinematics translates those locations into joint angles which are expressed in the joint space. These discrete joint angles join in a trajectory, in a continuous form, through the smooth function that is a part of the path generation and appears between the inverse kinematics and the inverse dynamics in the flow chart. (See Fig. 3.) With the trajectory points the inverse dynamics derives the required torques at the corresponding joints.

A. The Inverse Kinematics

With the assistance of the inverse kinematics algorithm, the path generator to be mentioned in subsection C only needs to give the desired location of certain chosen points in task space. Normally, computing the set of joint angles to achieve the desired states depends upon solving the equations obtained from the forward kinematics. As the analytical methods need to consider the situations with multiple solutions or even no solution, we adopt the numerical methods to get the approximate joint angles in this research.

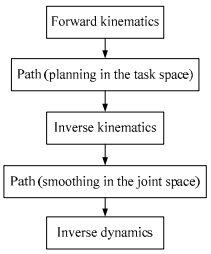


Fig. 3

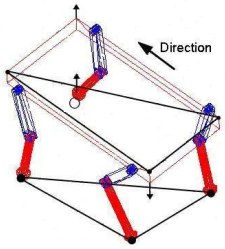


Fig. 4

Fig. 3 The flow chart of the inverse dynamics control

Fig. 4 The state of the quadruped robot swinging its leg

Let $f:\theta\in R^n\to SE(3)$, $f(\theta)$ represent the forward kinematics of the quadruped robot. The inverse kinematics formula can be stated as follows: $G\in SE(3)$, find $\theta\in R^n$ such that $f(\theta)=G$. Here f is the forward kinematics map, θ is the joint angle

vector and G is the desired position and orientation. If we introduce $F(\theta)$ and suppose that $F(\theta) = f(\theta) - G$, the inverse kinematics problem becomes an optimization problem which can be stated as the following: find $\theta \in R^n$ such that $\min\{\| F(\theta) \|\}$. Since G defines the forward kinematics constraints, the problem can be regarded as a nearest-neighbor nonlinear bounded optimization.

The algorithm for solving the optimization problem is listed as follows:

Step 1: import the position of the feet under the base frames and calculate the object G;

Step 2: initialize the joint vector with any given values θ_{ini} and propagate $f(\theta_{ini})$ to obtain the corresponding positions and orientations;

Step 3: evaluate the function $F(\theta)$ by computing the error between $f(\theta)$ and the object G, whose value is normalized by solving the potential function $P(\theta) = \| F(\theta) \|$;

Step 4: if the error is less than a constant value ε then stop the iterative routine, otherwise calculate the correction $\Delta\theta$;

Step 5: make $\theta = \theta_{ini} + \Delta\theta$ then propagate $f(\theta)$ and back to step 3.

The potential function $P(\theta)$ is

$$P(\theta) = \alpha \| p_{obj} - p_{cur} \|^2 + \beta \| w_{obj} - w_{cur} \|^2 \qquad (2)$$

Here p_{obj} represents the objective position vector of these frames like $\{RF\}$, $\{H - RF\}$ and $\{K - RF\}$, which are calculated from step 1, p_{cur} is the current position vector calculated from step 2 or from step 5, α and β are coefficients, w_{obj} is the objective angular vector calculated from 3×3 rotation matrix [6], which can be obtained from step 1, and w_{cur} is the current angular vector obtained from step 2 or from step 5.

B. The Inverse Dynamics

The trajectory points θ, $\dot{\theta}$, and $\ddot{\theta}$ can be obtained through the forward and the inverse kinematics, next, in this section, the required vector of joint torques, τ, needs to be found. In order to faster develop equations of motion, express them succinctly in symbolic form, and to derive dynamics algorithms quickly, we use the spatial vector with six dimensions instead of three dimensions. The motion of the links contained in the quadruped robot is expressed as the sum of six elementary motions: the linear velocities of v_{ox}, v_{oy}, and v_{oz}, and the angular velocities of ω_{ox}, ω_{oy}, and ω_{oz}. That is $\xi = [v_{ox}, v_{oy}, v_{oz}, \omega_{ox}, \omega_{oy}, \omega_{oz}]^T$.

The quadruped robot is a floating-base system in which the base is free to move, rather than being fixed in space [5]. Generally, the inverse dynamic control adds a fixed base to the system and a new virtual 6-DOF joint connecting the fixed base and the floating base. This method requires lots of extra computation on the virtual joints. As the inverse dynamics is a question of finding the forces required to produce a given acceleration in the quadruped robot, in a rigid-body system, there exists a strategy that doesn't need to calculate the extra virtual joints. The inverse dynamics algorithm based on recursive Newton-Euler relations imagines the quadruped robot as four separated robots as stated in the previous sections, i.e., a leg with a virtual body consisting of a serial robot, and defines a stride of a cycle as two states: the stance phase and the swing phase. When the serial robot is in the stance phase, the hip joint responds to the body's acceleration and gravity, as well the knee joint is answerable to the acceleration and gravity of all the parts up to the knee. When the serial robot is in the swing phase, the knee joint only needs to be in charge of the uplift of the shank and the hip joint responds to the thigh and the shank which are below the hip. The dynamic model, which we will go into detail next, chooses different parts to be the base according to the different phases, and then there is no need to calculate the extra virtual joints.

As the four serial robots are similar to each other in the dynamics, we will analyze one of them here to get insight on the relationships between different links. Fig. 5 shows the configuration of the serial robot, which is composed of the body, the hip joint, the thigh, the knee joint and the shank.

B.1) The Stance Phase

When the serial robot stands on the ground in the stance phase, the dynamic model supposes the shank to be the base. The origin of the base frame is located at the knee joint and then the body becomes the end. The hip joint responds mainly to the acceleration of the body and is resistant to the body's gravity. The knee joint supports the movements of the body and the thigh from below. (See Fig. 5(a).)

According to Newton-Euler's equations in [7] and the spatial vectors of six-dimensional vectors in [5], the inverse dynamics of the serial robot in the stance phase are given by

$$\begin{bmatrix} f_h \\ \tau_h \end{bmatrix} + \begin{bmatrix} f_b^g \\ \tau_b^g \end{bmatrix} = I_b^S \dot{\xi}_b + \xi_b \times I_b^S \xi_b \,, \qquad (3)$$

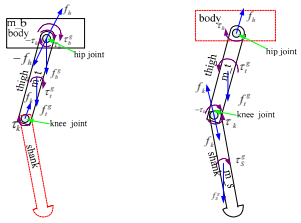
$$\begin{bmatrix} f_k \\ \tau_k \end{bmatrix} + \begin{bmatrix} f_t^g \\ \tau_t^g \end{bmatrix} - \begin{bmatrix} f_h \\ \tau_h \end{bmatrix} = I_t^S \dot{\xi}_t + \xi_t \times I_t^S \xi_t \,, \qquad (4)$$

$$u_h = \begin{bmatrix} P_h \times a_h & a_h \end{bmatrix} \begin{bmatrix} f_h \\ \tau_h \end{bmatrix}, \qquad (5)$$

$$u_k = \begin{bmatrix} P_k \times a_k & a_k \end{bmatrix} \begin{bmatrix} f_k \\ \tau_k \end{bmatrix} \,, \qquad (6)$$

where f_h and f_k are the forces, τ_h and τ_k are the torques of the hip joint and the knee joint respectively, the superscript g means the gravity, I_b^S is the spatial inertial tensor of the body and I_t^S is the spatial inertia tensor of the thigh; ξ_b and ξ_t are the spatial velocity of the body and the spatial velocity of the thigh, $\dot{\xi}_b$ represents the spatial acceleration of the body, and $\dot{\xi}_t$ represents the spatial acceleration of the thigh, the symbols a_h and a_k are the angular vectors stated in the section Ⅱ, which

are equal to a_{H-RF} and a_{K-RF}, respectively, P_h is the position vector of the hip joint relative to the base frame and P_k is the position vector of the knee joint relative to the base frame, the symbols u_h and u_k represent the joint actuator torque of the hip joint and the joint actuator torque of the knee joint, respectively. The spatial inertia tensor formed as I^S defines the relationship between the rigid body's velocity and momentum, which is a 6×6 matrix and can be calculated according to [5, 6, 7, 29].



(a) *(b)*

Fig. 5(a) The serial robot in the stance phase. Fig. 5(b) The serial robot in the swing phase

B.2) The Swing Phase

When the serial robot swings in the air, the base frame of the serial robot is located at the body and its origin coincides with the hip joint. It means that the root of the serial robot is the body and the hip joint responds to the upward lift of all the parts of the leg. The knee joint responds to the ascent of the shank and the resistance against the gravity of the shank.

The order of computing joint torques in the swing phase is the opposite direction of the order of computing joint torques in the stance phase, which propagates the forces and torques from the shank to the body. Fig. 5(b) shows the forces and the torques in the swing phase. The relationship between the forces and the torques is given by

$$\begin{bmatrix} f_k \\ \tau_k \end{bmatrix} - \begin{bmatrix} f_S^g \\ \tau_S^g \end{bmatrix} = I_S^S \dot{\xi}_S + \xi_S \times I_S^S \xi_S , \tag{7}$$

$$\begin{bmatrix} f_h \\ \tau_h \end{bmatrix} - \begin{bmatrix} f_t^g \\ \tau_t^g \end{bmatrix} - \begin{bmatrix} f_k \\ \tau_k \end{bmatrix} = I_t^S \dot{\xi}_t + \xi_t \times I_t^S \xi_t , \tag{8}$$

$$u_h = \begin{bmatrix} P_h' \times a_h & a_h \end{bmatrix} \begin{bmatrix} f_h \\ \tau_h \end{bmatrix}, \tag{9}$$

$$u_k = \begin{bmatrix} P_k' \times a_k & a_k \end{bmatrix} \begin{bmatrix} f_k \\ \tau_k \end{bmatrix}, \tag{10}$$

where ξ_S is the spatial velocity of the shank, $\dot{\xi}_S$ is the spatial acceleration of the shank, I_S^S corresponds to the shank's spatial inertia tensor, P_h' represents the position vector of the hip joint relative to the base of the body, and P_k' represent the position vector of the knee joint under the base frame, whose original location coincides with the hip joint.

These equations are evaluated link by link, starting from the link of the body in the stance phase or starting from the link of the shank in the swing phase, and work inward from end-effectors toward the base of the serial robot. The joint actuator torques can be derived by applying θ, $\dot{\theta}$ and $\ddot{\theta}$ to the recursive Newton-Euler equations.

C. The Path Generation

Finding the joint forces and torques not only needs the robot's geometrical and inertial parameters, but also should not exclude the robot's joint motions. The joint motions in joint space are converted from the natural description of the robot's feet in the task space in terms of the forward kinematics, the inverse kinematics and the path generation.

The path generation contains the path generator and the smooth function. The path generator starts from the path points defined by a human user in the task space and converts them into a set of desired joint angles by application of the inverse kinematics stated in the subsection A. As the quadruped robot is viewed as four separated serial robots in the kinematics and the dynamics, this path generator will consider the feet locations within a cycle in task space by integrating all the imagined separated robots into one.

The locations of the feet which are defined based on their respective base frames have two cases: the case of the leg in the stance phase and the case in the swing phase. When the serial robots stand on the ground, i.e. in the stance phase, the programming of the movement of the feet actually means the programming of the body's movement, as the feet are at fixed points on the ground. When the serial robots swing in the air, the programming of the feet's movement is the path of the feet.

The path generator also takes into account the stability of the quadruped, whose leg has two degree of freedoms. The robot with pitch-pitch type joints cannot stretch its feet out of the sagittal plane that is configured by the front leg and the hind leg at the same side. That means the classical stability criteria composed of the support polygon doesn't fit the type robot referenced in this research and the robot has difficulty in moving in the transverse plane. The quadruped robot guarantees its stability by retracting the diagonal leg to prevent the system from tumbling. (See Fig. 4.) The leg whose diagonal leg shrinks correspondingly swings and steps out compliantly in the Fig. 4.

Then the smooth function, which appears between the inverse kinematics and the inverse dynamics in the flow chart of the Fig. 3, connects the set of joint angles by cubic polynomials. Given the waypoints, there is a unique piecewise-cubic spline that smoothly passes through the points.

IV. THE EXPERIMENTS

As stated in the kinematics and in the dynamics, it is necessary to identify the stance phase and the swing phase. There are two methods to accomplish this. The first one uses the contact sensor and the other one tags the flag in the arrays of data offline. We utilize the second method in this research in order to ensure the idea can be widely used. The sequence of the foot lifting is denoted as RF-LH-LF-RH, in which the RF represents the right front leg, the LH represents the left hind leg,

the LF represents the left front leg, and the RH represents the right hind leg.

If we consider that the quadruped robot walks from the initial state showed in Fig. 2, the arrays of the location data, which are defined in the task space, are expressed as:

RF: [[0, -H, 0, 0]; [NS, -H+KAH, 10, 1]; [4NS, -H+d*KAH, 15, 1]; [6*NS, -H+2*KAH, 20, 0]; [5*NS, -H+KAH, 25, 0]; [3*NS, -H+MAH, 45, 0]; [2*NS, -H, 50, 0];[2*NS, -H, 70, 0]; [0, -H, 75, 0];[-NS, -H, 95, 0]; [-NS, -H, 100, 0]; [-NS, -H-KAH, 110, 1]; [2*NS, -H+d*KAH, 115, 1];[5*NS, -H+KAH, 120, 0]; [5*NS, -H+KAH, 125, 0]];

LH: [[0, -H, 0, 0]; [NS, -H+MAH, 20, 0]; [0, -H, 25, 0]; [-2*NS, -H-KAH, 35, 1]; [NS, -H+1.5*d*KAH, 40, 0]; [3*NS, -H+KAH, 45, 0]; [2*NS, -H, 50, 0]; [2*NS, -H, 70, 0]; [0, -H, 75, 0];[0, -H, 95, 0];[0, -H, 100, 0]; [0, -H+MAH, 120, 1];[-NS, -H, 125, 0]];

LF: [[0, -H, 0, 0]; [NS, -H, 20, 0]; [0, -H, 25, 0];[-2*NS, -H, 45, 0]; [-2*NS, -H, 50, 0]; [-2*NS, -H-KAH, 60, 1]; [NS, -H+d*KAH, 65, 1]; [4*NS, -H+KAH, 70, 0]; [3*NS, -H, 75, 0]; [0, -H+MAH, 95, 0]; [0, -H, 100, 0];[0, -H, 120, 0]; [-NS, -H, 125, 0]];

RH: [[0, -H, 0, 0]; [NS, -H, 20, 0]; [0, -H, 25, 0]; [-2*NS, -H, 45, 0]; [-2*NS, -H, 50, 0]; [-2*NS, -H+MAH, 60, 0]; [-2*NS, -H, 70, 0]; [-3*NS, -H, 75, 0]; [-3*NS, -H-KAH, 85, 1]; [-NS, -H+d*KAH, 90, 1]; [0, -H+KAH, 95, 0]; [0, -H, 100, 0];[0, -H, 120, 0]; [-NS, -H, 125, 0]].

All the arrays of the data define the foot locations of one cycle with the walking gait, where NS is the stride, H represents the height of the foot under its kinematics base frame, KAH is compensation value, d corresponds to the leg shrinking quantity when the relevant leg is in the swing phase, and MAH is the diagonal leg shrinking quantity to stabilize the quadruped robot. On the whole, the first elements of the arrays represent the stride of the leg and the second elements of the arrays represent the height of the feet compared to the relevant base frames. The third elements are time series, whose units can be the second or the millisecond. The time elements help the cubic polynomial to fit the data with continuous time. The fourth elements in the subarray label the swing phase with the flag 1 and the stance phase with the flag 0. As the swing phase or the stance phase is the period of time, the flag 1 or the flag 0 means that the time progresses from the start of one phase to the beginning of the next. For example, the swing phase starts with the RF leg from the time 10 through the time 15 till the time 20, i.e., during the time 10 till the time 20 the RF leg is in the swing phase.

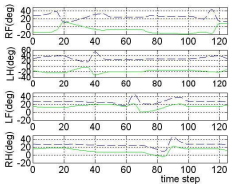


Fig. 6

Fig. 6 The joint angles in the joint space

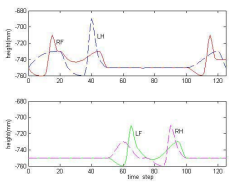


Fig. 7

Fig. 7 The foot locations in the task space

The dashed lines represent the angles of the knee joints and the solid lines represent the hip joint angles. The horizontal axis represents time steps, and the vertical axis represents the joint angle.

With the locational data the inverse kinematics converts them into the joint variables and the cubic spline connects them into continuous curves. (See Fig. 6) The values of these parameters are as follows: NS is 40mm, H is $\sqrt{(750)^2 - F^2}$ mm, KAH is 6mm, d is 3, and MAH is 20mm, where F represents the first element in the array.

In order to check the correctness of these joint angles, we employ the forward kinematics to transform these angles to the foot locations in the task space. (See Fig. 7.) The values in the vertical axis are negative, because the base frames located at the hip joint and the directions of these axes are opposite to the positional direction under the base frames.

The solid curve in the top graph is the RF foot locations in the task space and the dashed curve in the top graph is the RF's diagonal foot locations, the LH foot locations. The solid curve in the lower part of the graph represents the LF foot locations and the dashed curve is the RH foot locations. Fig. 8 gives the foot information in the sagittal plane and shows that the diagonal leg shrinks a little back to keep the robot's stability before the leg lifting.

From the Fig. 7 we can make sure that the quadruped robot with the joints motion can walk without tumbling. However, in order to generate the desired trajectory for the robot, it is important to find the actuator torques and forces. Through the path generation formulae, all the joint angles, angular velocities and angular accelerations can be passed to the inverse dynamics as time variables. When the forth elements is 1 in the subarrays, the dynamics adopts the model of the swing phase and when the forth elements is 0 the dynamics becomes the stance model.

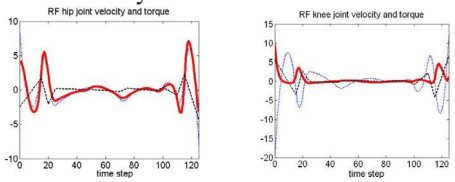
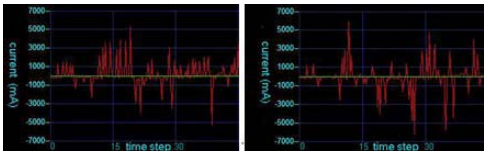


Fig. 8 The joint velocities, the joint accelerations and the torques

It is can be found that the torques of the hip joints follow tightly upon the hip joint velocities. The phenomenon indicates that the joint velocity is the main reason that responds to the joint torque. It can also be observed that the torque curves at the knee joints are similar to the angular accelerations' curves. From this point the knee joint torques are affected mostly by the joint accelerations.



(a) (b)

Fig. 9 The current curves at the right front hip joint in the swing phase, (a) The current curve under the inverse dynamics control, (b) The current curve under the high gain PD control

Fig. 9 indicates the current at the right front hip joint when the joint is in the swing phase. Fig. 9(a) shows the RFhip joint's current under the inverse dynamic control and the current curve in Fig. 9(b) is under the high gain PD control. The current in Fig. 9(a) is smoother than the current in Fig. 9(b) which accelerates and decelerates the corresponding motor frequently. With the current in Fig. 9(a) the driver of the joint runs the motor so strongly that the quadruped robot can walk on the rubber mat and on asphalt.

V. CONCLUSION

In this paper, we addressed the problem of how to reduce the computational load, proposed an approximated way of calculating inverse dynamics of a free floating base system without knowing the contact forces involved, and realized compliant control of the quadruped.

Starting from the kinematics, the model assumes the quadruped robot as four independent serial robots and affixes their respective base frames to the hip joints. The inverse dynamics continues using the structure that divides the quadruped robot up into four serial robots, supposes different models in terms of different phases, and eliminates the virtual joint with 6 DOFs which are usual in the rigid body system with a floating base. Although the idea of regarding the quadruped as four serial robots increases the number of the robots, it reduces greatly the number of calculations that are needed to be made for the virtual joints. The path generator integrates all the serial robots into one and considers them from the view point of stable movement. Then, the quadruped can walk robustly with the control strategy.

The experiments demonstrated that the idea of simplifying the quadruped robot into several serial robots was effective. The inverse kinematics was turned into an optimal problem to clear up the cases of no solution and multi-solution. In order to verify the correctness of this transformation, the joint angles were imported to the forward kinematics anew and the uninterrupted curves of the foot locations in Fig. 7 obviously certified that the idea, which transformed solving the equations into finding the local optimal solutions, was valid. The current comparison of the high gain PD control with the control in this paper showed that the inverse dynamics control strategy could run the motor more smoothly. The locations of the path generator exhibited the data array as a human-user interface in the task space. Through the interface, the quadruped robot can accomplish more advanced actions such as crossing over the obstacles and climbing the stairs.

REFERENCES

[1] J. Zico Kolter, Mike P. Rodgers, and Andrew Y. Ng, "A control architecture for quadruped locomotion over rough terrain" *Proceedings of 2008 IEEE, International Conference on Robotics & Automation,* pp. 811-818.

[2] Michael. Mistry, Jonas Buchli, and Stefan Schaal, "Inverse Dynamics Control of Floating Base Systems Using Orthogonal Decomposition" *Proceedings of the 2010 IEEE, International Conference on Robotics & Automation,* pp. 3406–3412.

[3] Jonas Buchli, Mrinal Kalakrishnan, Michael Mistry, Peter Pastor, and Stefan Schaal, "Compliant Quadruped Locomotion Over Rough Terrain" *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 814–820.

[4] Marc H. Raibert, Michael Chepponis, and H. Benjamin Brown, "Running on four legs as though they were one" *IEEE Journal of Robotics and Automation,* pp. 70–82. vol.2/1986

[5] Roy Featherstone, "Rigid Body Dynamics Algorithms" Springer Science + Business Media, LLC., 2007

[6] R. Murray, Z. Li, S. Sastry, A Mathematical Introduction to Robotic Manipulation, CRC, Boca Ranton, FL, 1993

[7] J. Craig, Introduction to robotics: mechanics and control, 3rd, (2004).

[8] P. De Santos, J. Estremera, E. Garcia, M. Armada, Including joint torques and power consumption in the stability margin of walking robots, *Autonomous Robots* 18 (1) (2005) 43-57.

[9] Xiuli Zhang, Haojun Zheng, "Walking up and down hill with a biologically-inspired postural reflex in a quadrupedal robot" *Autonomous Robots,* pp. 15-24. vol.25(1-2), 2008.

[10] Bin Li, Xun Li, Wei Wang, Yanfeng Tang Yiping Yang, "A Method Based on Central Pattern Generator for Quadruped Leg Control" *Proceedings of the 2009 IEEE, International Conference on Robotics and Biomimetics,* pp. 2035–2041. vol.15/2009.

[11] Xun Li, Wei Wang, Bin Li, Yanjun Wang, Yiping Yang, "Central Pattern Generators Based Adaptive Control for a Quadruped Robot" *Proceedings of the 2009 IEEE, International Conference on Robotics and Biomimetics,* pp. 2068–2073. vol.15/2009.

[12] P. De Santos, J. Galvez, J. Estremera, E. Garcia, "SIL04: a true walking robot for the comparative study of walking machine techniques", *IEEE Robotics & Automation Magazine,* Vol.10 (4) (2004) pp.23-32.

[13] P. González-de-Santos, E. Garcia, J. Estremera, "Quadrupedal locomotion: an introduction to the control of four-legged robots", Springer, 2006.

[14] J. Chestnutt, K. Nishiwaki, J. Kuffner, S. Kagami, "An adaptive action model for legged navigation planning", *IEEE-RAS International Conference on Humanoid Robots,* 2007, pp. 196-202.

[15] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, S. Schaal, "Fast, robust quadruped locomotion over challenging terrain", *Proceedings of 2010 IEEE, International Conference on Robotics & Automation,* pp. 2665-2670.

[16] M. Kalakrishnan, J. Buchli, P. Pastor, S. Schaal, "Learning locomotion over rough terrain using terrain templates", *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 167-172.

[17] M. Mistry, "The representation, learning, and control of dexterous motor skills in humans and humanoid robots", University of South California, Ph. D, (2009).

[18] D. Pongas, M. Mistry, S. Schaal, "A robust quadruped walking gait for traversing rough terrain", *Proceedings of 2007 IEEE, International Conference on Robotics & Automation,* pp. 1474-1479.

[19] J. Rebula, P. Neuhaus, B. Bonnlander, M. Johnson, J. Pratt, "A controller for the littledog quadruped walking on rough terrain", *Proceedings of 2007 IEEE, International Conference on Robotics & Automation,* pp. 1467-1473.

[20] I. S ucan, M. Kalakrishnan, S. Chitta, "Combining planning techniques for manipulation using realtime perception", *Proceedings of 2010 IEEE, International Conference on Robotics & Automation,* pp. 2895-2901.

[21] A. Shkolnik, M. Levashov, S. Itani, R. Tedrake, "Motion planning for bounding on rough terrain with the littledog robot" *Proceedings of 2010 IEEE, International Conference on Robotics & Automation,* pp.134-142

[22] M. Stolle, H. Tappeiner, J. Chestnutt, C. Atkeson, "Transfer of policies based on trajectory libraries", *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 2981-2986.

[23] F. Aghili, "A unified approach for inverse and direct dynamics of constrained multibody systems based on linear projection operator: Applications to control and simulation", *IEEE Transactions on Robotics* Vol.21 (5), 2005, pp.834-849.

[24] S. Lee, J. Kim, F. Park, M. Kim, J. Bobrow, "Newton-type algorithms for dynamics-based robot movement optimization", *IEEE Transactions on Robotics,* Vol.21 (4), 2005, pp. 657-667.

[25] R. Featherstone, "The acceleration vector of a rigid body", *The International Journal of Robotics Research* Vol.20 (11), 2001, pp.841-846.

[26] R. Featherstone, "Efficient factorization of the joint-space inertia matrix for branched kinematic trees", *The International Journal of Robotics Research,* Vol.24 (6), 2005, pp.487-500.

[27] R. Featherstone, "Plucker basis vectors", *Proceedings of 2006 IEEE, International Conference on Robotics & Automation,* pp. 1892-1897.