Design Optimization and Dynamic Balance Control of a 6-DoF Wheeled Biped Robot

K. Sinaei*, MH. Sarfi*, E. Hosseinian††

Abstract—Wheeled bipedal robots are evolutionary form of mobile robots, capable of maneuvering in complex environments. In this project, dynamic characteristics of a 6-Degree of Freedom wheeled legged robot has been investigated and an optimal design for the knee mechanism was proposed. By employing the Genetic algorithm (GA), a 4-bar mechanism has been designed for the knee joint of the robot, this design enhanced the robot’s dynamic properties. Furthermore, it benefits the robot mechanically by increasing its maximum payload. Due to system’s inherent instability, a control unit is required for dynamic balancing of the robot during its planar motions. To address dynamic instability, a novel optimal control scheme has been introduced. The proposed control unit enables the robot to keep its balance while moving at desired speed and height in forward and backward direction. The cart and inverted pendulum model was exploited as an equivalent linearized model of the robot to design and implement a real robot. The simulated robot has been tested in a simulation environment to evaluate the performance of the proposed control method. Transient time response of the robot when following an input velocity and height command denotes that the controller’s steady state error is negligible and there are mild vibrations in its tilt angle. Overall, with the introduced method the robot is capable of keeping its balance, controlling its forward velocity and also rejecting disturbance forces.

Keywords—Wheeled bipedal robot, Genetic algorithm, optimal control, Linear Quadratic Regulator, cart and inverted pendulum, parallel mechanisms

I. INTRODUCTION

Nowadays robots play an essential role in the automation logistics of many corporations. Versatility of robots include package delivery, warehouse assistant, industrial inspection, and even as social companions [1] [2] [3]. Designing requirements of robots include avoidance of indoor obstacles and efficient maneuvering objects. Ground robots could be classified into leg (or foot) base robots and wheel-based robots [7]. A myriad of researches has been carried out in the development of robots from both categories. For example, TORO is a human-size torque-controlled bipedal robot evolved from its former version (DLR biped) [8]. iCub is another example for leg-based robot, which is a 104 cm humanoid robot that has a size of about three-year-old child, capable of crawling, walking and manipulating objects [9]. The leg-based robots have good obstacle surmounting performance with low moving efficiency and slow speed but difficult to achieve stability. On the other hand, wheeled robots have larger moving speed and efficiency but poor adaptability in complex pavements [10]. uBot is one example of wheel-based robots that was fabricated at the Laboratory for Perceptual Robotics [11]. It has two coaxial wheels and two arms, enabling manipulation of objects. It could also recover its balance using multiple contact points [11].

Drawbacks from both of the abovementioned configuration invented the idea of combining wheeled robots with the configuration of a leg-type robot. During the past decade, several prototypes of robots with 6 degrees of freedom (6-DoF) robots have been designed and built. Key characteristics of these robots are enhanced transportation with higher energy efficiency and the capability of discovering uneven terrains such as inclined surfaces and stairs. On top of that, they can squat or extend their legs, giving tunability in robot’s height control. This feature improves robot motion and enhance its capabilities in domestic applications where the robot needs to pass below tables or shelves.

In 2015 DARPA Robotics Challenge (DRC) Hubo team employed a novel approach in which they used pair of extra wheels at each shank of the humanoid robot to readily complete the task. The configuration used in the DRC-Hubo robot is a good representative of a hybrid robot that combines bipedal gait with wheeled rolling motion, and the problem.[12]. Ascento is another wheeled legged robot developed at Autonomous Systems Lab of the ETH [13]. It uses two actuators for driving the wheels and another two for moving the legs. Ascento has a parallel linkage in its leg configuration, making it economical with less inertia and reduced mechanical complexity. Thanks to its Linear-quadratic Regulator (LQR) whole-body controller, it has higher performance for outdoor environments [13]. SR-600 is another 6-DoF robot, developed at Harbin Institute of Technology. It is a five-bar linkage with two motors at each

### TABLE I. LIST OF SYMBOLS AND NOTATIONS USED IN THIS PAPER

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>Mass of wheels</td>
<td>kg</td>
</tr>
<tr>
<td>m</td>
<td>Mass of robot’s body</td>
<td>kg</td>
</tr>
<tr>
<td>l</td>
<td>Pendulum length in cart-inverted pendulum model</td>
<td>m</td>
</tr>
<tr>
<td>(\varphi)</td>
<td>Pendulum angle in cart-inverted pendulum model</td>
<td>rad</td>
</tr>
<tr>
<td>g</td>
<td>Gravitational acceleration</td>
<td>m/s²</td>
</tr>
<tr>
<td>x</td>
<td>Robot’s pelvis position (equal to cart’s position)</td>
<td>m</td>
</tr>
<tr>
<td>(\beta)</td>
<td>Pitch angle of robot’s pelvis</td>
<td>rad</td>
</tr>
<tr>
<td>u</td>
<td>Input force in cart-inverted pendulum model</td>
<td>N</td>
</tr>
<tr>
<td>(\Psi)</td>
<td>Knee motor angle</td>
<td>rad</td>
</tr>
<tr>
<td>P</td>
<td>Robot’s shank and hip length</td>
<td>m</td>
</tr>
<tr>
<td>(\theta_i)</td>
<td>Robot’s joint angle</td>
<td>rad</td>
</tr>
<tr>
<td>(\nu_d)</td>
<td>Desired velocity (controller input command)</td>
<td>m/s</td>
</tr>
<tr>
<td>d</td>
<td>Robot height (controller input command)</td>
<td>m</td>
</tr>
<tr>
<td>(T_{RH})</td>
<td>Knee motor torque</td>
<td>N.m</td>
</tr>
<tr>
<td>K</td>
<td>Length of link O-O₂ in knee mechanism</td>
<td>m</td>
</tr>
<tr>
<td>L</td>
<td>Length of link O-O₂ in knee mechanism</td>
<td>m</td>
</tr>
<tr>
<td>R</td>
<td>Cost function of knee mechanism optimization</td>
<td>-</td>
</tr>
<tr>
<td>Q</td>
<td>Cost on state (in LQR)</td>
<td>-</td>
</tr>
<tr>
<td>R</td>
<td>Cost on controlled input (in LQR)</td>
<td>-</td>
</tr>
<tr>
<td>I</td>
<td>Identity matrix</td>
<td>-</td>
</tr>
</tbody>
</table>

* Corresponding author
† School of Mechanical Engineering at the University of Tehran
‡ [kasra.sinaei, mhsarfi, ehosseinian]@ut.ac.ir
hip and another two for driving the wheels. This robot was modeled as an inverted pendulum and its dynamic balance was achieved by placing a constraint for the location of center of gravity (CoG) [14]. Xin et al. proposed a more human-like wheeled biped robot with 6 degrees of freedom in each leg [15]. It can generate whole-body motion for a given input using the Cart-Linear inverted pendulum model for rolling motion and under-actuated Linear Inverted Pendulum for contact changes during gait. The well-known LIPM model introduced by Kajita [16] cannot generate walking trajectories for robots with single point contact (all wheeled biped robots have contact with ground in a single point), therefore, support polygon could not be defined for this kind of robots. As a result of that Xin used under-actuated LIPM model which is capable of generating walking trajectory with the point feet assumption. Boston Dynamics Handle [17] and Harbin Institute of Technology’s Wheeled legged Robot (WLR) [14] are other examples of humanoid wheeled robots with manipulators connected to the floating base. The addition of manipulator arms makes these robots a great assistive robot for a range of applications such as package delivery and transportation. Wheeled biped robot’s application has been extended to wearable robots as well. For instance, Torno et al. have proposed a design and control scheme for a sustainable wheeled-biped exoskeleton [18]. In addition to mentioned approaches toward stabilizing wheeled biped robot, balance control could also be achieved by adding extra degree of freedom to the robot. As suggested by Larimi et al. a reaction wheel could be used for stabilizing a self-balancing robot, this method could be applied to a wheeled biped robot as well [19].

In this project, a method has been presented which is capable of controlling forward or backward velocity of a wheeled biped robot while maintaining its balance to overcome the inherent instability of the system so the main focus of this work is to stabilize the robot while it is following velocity and height commands. To achieve this, a Linear Quadratic Regulator has been used alongside a module that solves kinematics of the robot with analytic and geometric approach. Herein, an LQR controller has been utilized to stabilize the robot. LQR is an optimal controller that minimizes a quadratic cost function to generate an appropriate input of the plant. To solve LQR least-square problem dynamic programming has been used, which is an efficient recursive algorithm for online controllers [20]. In another part of the project, Genetic algorithm was employed to maximize knee joint mechanical performance by finding optimal linkage lengths of a 4-bar parallel mechanism. In order to test viability of the proposed control method, a simple yet accurate model of robot has been simulated using an open source physics engine (Choreonoid with AIST physics engine), and the robustness of the control system has been evaluated by recording its transient time response.

II. DESIGN

A. Multibody System Representation

As described in the previous section we are dealing with a hybrid wheeled biped robot. It consists of eleven links including its floating base so-called pelvis as shown in root of Fig. 1. The pelvis is parent of two other identical links which are the robot’s hip and each hip is parent of robot’s shank. Each leg has a parallel 4-bar linkage to be used to indirectly drive the knee joint. At the bottom of this hierarchy, the wheels are connected to the shanks. The link tree of the described system including parallel knee linkage is illustrated in Fig. 1.

In this part the main objective is to design a robot for indoor applications so we are keeping its size minimal. The pelvis should provide enough space for keeping electronic devices such as IMU sensor, Gyro sensor, motor drivers and microcontroller. To meet these requirements, pelvis dimensions were chosen to be 200mm × 100 mm × 70 mm. Shanks and hips are both 200 mm and the wheel’s diameters are 100 mm. These dimensions give the robot a standing height of 45 cm. While the employed mechanism for the knee joint limits the robot’s workspace it boosts the maximum payload of the robot. The goal of adding parallel links to the hip was to enhance the performance of the knee joint, the exerted torque to the knee motor and its energy consumption has been minimized and robot’s mechanical properties such as maximum payload has been enhanced. In the next section, procedure of finding the length of these links will be explained in detail.

B. Knee Parallel Mechanism

As in conventional humanoid robots, in this case knee joint is expected to bear larger loads and to move at high angular velocities to follow desired trajectories [21]. As such Genetic Algorithm (GA) [22] was used to find the optimal lengths of the four-bar linkage system depicted in Fig. 2. The cost function is the weighted sum of exerted torque and consumed energy during a constant-velocity rotation of the knee joint. For simplicity of analysis, link H is assumed to be the fixed link while motion of pelvis and shank (link N) has been approximated and considered as a constant equivalent force calculated using D’Alembert’s principle.
The pelvis for estimating the Tst of the links and pelvis are. 

which can be used in the LQR controller. 

In order to design a control loop with sufficient stability, it is essential to estimate the robot’s state including the pelvis pitch angle, angular and linear velocity, posture of the mechanism and angular velocity of the wheels. To measure these signals, an Inertial measurement unit (IMU) and a Gyroscopic sensor were attached to the pelvis. All joints of the robot except the wheels have absolute encoders. In summary, there are four position-controlled joints (hips and knees in both legs) and two torque-controlled joints in wheels with angular position and angular velocity measurements for the feedback.

III. SIMULATION

After optimization of kinematic parameters of the mechanism, multibody computer-aided model of the robot was developed. Choreonoid and Robot Operating System (ROS) have been used for the robot simulation. Using ROS communication protocols the generated libraries for simulation could be used directly with a real robot. Choreonoid was chosen as it owns powerful features in graphical user interface (GUI) in addition to reliable, intuitive plugins for different sensors [25]. The final result of the simulated robot has been shown in Fig. 4.

IV. CONTROLLER DESIGN

To control the robot during its forward and backward motion in horizontal plane we simplified the dynamics of the robot to obtain an approximated linear model. The simplified model used for designing controller is the cart and inverted pendulum model and it has been illustrated next to the exact model of the robot in Fig. 6.

In this model we assume that robot’s wheels’ motion is pure rolling so the wheels could be equivalent to the cart in the simplified model. The rest of the links and pelvis are equivalent to pendulum of simplified model. Dynamic equations of this system could be found in [26], here we chose pendulum’s tilt angle, pendulum’s angular velocity, cart’s position, and cart’s velocity in x direction as state variables: \( X = [\phi, \dot{\phi}, x, \dot{x}] \). State variables of the simplified model and their equivalence in real robot model has been shown in Fig. 6 as well. The main controller objective is to stabilize the system using its single controlled output which is the applied force to the cart (\( u \)). To adjust the robot’s height, we solve the inverse kinematics of each leg in a way that the links (shank, hip and pelvis) CoM stays exactly above the wheels’ axis (dashed line in Fig. 4) when robot has zero tilt angle. Since robot’s leg will not move during the rest of the process, we can use the measurements of the IMU attached to the pelvis for estimating the robot’s tilt angle and angular velocity. In the next two parts of this article, mathematical model of the proposed control strategy will be presented.

A. LQR and Mathematical Model

Dynamic equations of motion for a cart and inverted pendulum has been shown in equations (3) [27]:

\[
\begin{align*}
\dot{\phi} &= \frac{mlg(t) - ml\ddot{x}(t)}{(1+m)l} + \frac{mg}{M+m} \\
M+m\ddot{x} &= u(t)
\end{align*}
\]

With the state variables defined in previous section we can obtain the state space model of the robot with full state feed-back that could be used in the LQR controller. In equation (4) is identity matrix. Substituting the values of M,
m, g, and l in equations (4) we can see that the system is controllable so this model could be exploited for designing controllers.

\[
\begin{align*}
X &= \begin{bmatrix}
0 & 1 & 0 & 0 \\
\frac{M+m}{M} & 0 & 0 & 1 \\
0 & 0 & 0 & 1 \\
-\frac{m}{M} & 0 & 0 & 0
\end{bmatrix} \begin{bmatrix}
0 \\
1 \\
0 \\
0
\end{bmatrix} + \begin{bmatrix}
0 \\
\frac{1}{M} \\
0 \\
0
\end{bmatrix} u \\
Y &= \|X\|
\end{align*}
\]

(4)

The main stabilizer in the control system architecture shown in Fig. 5, is the Linear Quadratic Regulator block. LQR is an optimal full-state feedback controller that optimizes a quadratic cost function over a finite or infinite horizon to calculate the control output [28]:

\[
J = \int (X^T Q X + u^T R u) dt
\]

(5)

Where J is the cost function; Q is a diagonal matrix called state cost and R is another diagonal matrix representing the cost on input. A recursive dynamic programming approach has been utilized to obtain online solution of the controller’s optimization problem [29]. This method is applicable for finite-horizon discretized systems. First, the state space model shown in equation (4) was discretized with respect to controller’s operating rate. As such, Euler approximation is used to convert continuous model into a discrete system. The computational efficiency of this approach makes it a preferable approach for online calculations [30].

**B. Inverse Kinematics**

In this section, the objective is to adjust angular position of robot’s joints. Two geometric constraints were defined such that the module could calculate the joint angle of the hip and the knee motor. Specifically, it collects the desired height of the pelvis as the input and generates the required joint angles \( \theta_1, \theta_2 \) and \( \theta_3 \). One geometric constraint is the position of CoM of robot’s links (Pelvis, Shank and Hip) as expressed by Eq. 6.

An analytical framework was developed to solve the robot’s inverse kinematic scheme. Using this framework, configuration of robot could be achieved in polar coordinate system for any given position of pelvis relative to wheels’ axis. First, we focused on the legs without considering the four-bar mechanism and explicit solution for the angles \( \theta_1, \theta_2 \) and \( \theta_3 \) were obtained analytically. (see Fig. 2) Then simple trigonometry gives the joint angle of the knee motor \( \psi \). The solution for \( \theta_3 \) is redundant. This means that as long as the balance controller offers the required output signal, the pelvis remains parallel with the flat ground and \( \theta_3 \) will remain at the expected position.

\[
\begin{align*}
P \cdot \sin(\theta_1) + P \cdot \sin(\theta_1 + \theta_2) &= d \\
m_{\text{shank}} \xi_{\text{shank}} + m_{\text{hip}} \xi_{\text{hip}} + m_{\text{pelvis}} \xi_{\text{pelvis}} &= 0 \\
\theta_1 + \theta_2 + \theta_3 &= \pi
\end{align*}
\]

(6)

Where \( \xi_1 = \frac{p}{2} \cdot \cos(\theta_1) \), \( \xi_2 = P \cdot \cos(\theta_1) + \frac{p}{2} \cdot \cos(\theta_1 + \theta_2) \), and \( \xi_3 = P \cdot \cos(\theta_1) + P \cdot \cos(\theta_1 + \theta_2) \).

Joint axis directions (See Fig. 1) have been selected following Denavit and Hartenberg rules [31]. Using Eqs. 6 analytical solution for \( \theta_2 \) could be obtained. There are four set of explicit solutions to this equation set. The solution satisfying the constraint for the knee joint was chosen(5\( \leq \theta_2 \leq 168 \)). This constraint assures that 4-bar linkage abstain from singular configuration. Using the law of cosines in \( O_1O_2O_4 \) we have:

\[
a = \sqrt{p^2+K^2-2P.K.\cos \theta_2}
\]

(7)

Where P and K are length of links \( O_2O_3 \) and \( O_1O_2 \) respectively (refer to Fig. 2). Using the law of sines in the same triangle gives us:
\[
\frac{\sin \psi_1}{K} = \frac{\sin \theta_2}{a} \rightarrow \psi_1 = \sin^{-1}\left(\frac{K}{a} \sin \theta_2\right)
\] (8)

Applying law of cosines in the \( O_1O_2O_3 \), we obtain:

\[
L = \sqrt{R^2 + a^2 - 2Ra \cos \psi_2} \rightarrow \psi_2 = \cos^{-1}\left(\frac{R^2 + a^2 - L^2}{2a}\right)
\] (9)

Combining equations (8) and (9), and knowing that the joint angle \( \psi = \psi_1 + \psi_2 \):

\[
\psi = \sin^{-1}\left(\frac{K}{a} \sin \theta_2\right) + \cos^{-1}\left(\frac{R^2 + a^2 - L^2}{2aR}\right)
\] (10)

Overall, the control scheme consists of three submodules as shown in Fig5. First submodule is to obtain the desired configuration of robot leg for any given operating height so that the CoM of robot links will be held exactly on top of the robot’s wheels’ rotation axis. The second submodule is the state estimator to manipulate the data received from IMU and Gyro sensor. The last subsystem in the controller is the LQR to compute the required torque for the wheel to stabilize the robot.

### Table II. Control Loop Pseudo Code

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a)</td>
<td>Algorithm Inputs ( V_o, d )</td>
</tr>
<tr>
<td>b)</td>
<td>Set LQR cost matrices and horizon: ( Q, R, N )</td>
</tr>
<tr>
<td>c)</td>
<td>Set discretized state space matrices of the model based on d: ( A, B )</td>
</tr>
</tbody>
</table>
| d)   | Solve kinematics of legs  
- solve for all \( \theta_j \) joint configurations  
- calculate \( \psi \) using the value of \( \theta_2 \) |
| e)   | Calculate LQR controlled output:  
- initialize \( P, u, K \) (empty list of size N)  
- \( P_0 = Q \)  
- for \( i = N-1, \ldots, 1, 0 \):  
  - \( P_{i+1} = Q + A^TP_iA - A^TP_iB(R + B^TP_iB)^{-1}B^TP_iA \)  
  - for \( i = 0, 1, \ldots, N-1 \):  
    - \( K_i = (R + B^TP_iB)^{-1}B^TP_iA \)  
    - \( u_i = K_iP_i \)  
- return \( u_{N-1} \) as controlled input |
| f)   | Go to d |

The pseudocode of the control algorithm is presented in Table II. State space matrices \( (A, B) \) vary with the length of pendulum so if there is a change in pelvis height, the algorithm is capable of updating controller’s state space model. Section (d) of the pseudo code is IK block in Fig. 5 while section (e) is to discretize finite-horizon LQR, being formulated via dynamic programming [29].

### V. RESULTS AND CONCLUSION

In this project, length of the links in four-bar mechanism has been optimized with GA to design the mechanism for the robot’s knee joint. The optimization goal was to reduce the size of the motor by minimizing its energy consumption and total applied torque. Compared to a simple revolute joint, the designed mechanism reduced the power consumption at the knee joint by 31%. The obtained configuration keeps motor, mounted on the knee, closer to the pelvis, making the equivalent pendulum’s mass in the model larger improving stability of the system.

An LQR has been utilized alongside with geometric inverse kinematics module (IK block in Fig. 5) to stabilize the robot tilt angle and to control its forward/ backward velocity. The controlled input is the applied torque to the robot’s wheels. Given any desired state, robot manages to reach the goal state robustly.

In the first simulation we set the desired velocity of the robot to 0.4 m/s and the robot starts moving forward from its idle posture. The settling time is roughly seconds with steady state error being approximately zero s shown in the plot (a) of Fig. 7. Fig. 7b represents angular configuration of the pelvis as a function of time reaching the desired zero degrees along with small fluctuations the tilt angle of the robot.

To measure the system’s sensitivity to disturbance another scenario has been used for simulation. The robot drops from an initial height of 40 cm above the ground level.

![Fig. 7. Controlled parameters of the robot](image1)

![Fig. 8. Robots performance while rejecting external disturbance force](image2)
Then all controllers commence their processes, initially robot manages to reach its idle height which has been set to 32 cm (because of the mechanisms limited work space, it should be in the range of (27,37) cm). Desired velocity of the robot was set to zero and an external disturbance force was applied to the pelvis. The robot could manage to recover its balance as shown in the Fig. 8. The bump at t=2s is caused by external force.

The proposed controller meets the transient-response specification during forward/backward movements in a flat horizontal plane. Robot height (d in Fig. 2) is an online parameter, this means the height could be changed while robot is moving. Furthermore, Fig. 7 and Fig. 8 shows that the controller is capable of recovering the balance when dropped from a height above the ground while rejecting external disturbance.

REFERENCES


