Modeling and Control of a Hybrid Wheeled Legged Robot: Disturbance Analysis

Fahad Raza (r.fahad@neuro.mech.tohoku.ac.jp)
Tohoku University https://orcid.org/0000-0003-4798-2358
Dai Owaki
Tohoku Daigaku
Mitsuhiro Hayashibe
Tohoku Daigaku

Research Article

Keywords: Balance Control, Wheel-legged Robots, Optimal Control, Self-balancing Robot

Posted Date: February 6th, 2020

DOI: https://doi.org/10.21203/rs.2.22794/v1

Modeling and Control of a Hybrid Wheeled Legged Robot: Disturbance Analysis

Fahad Raza, Dai Owaki, and Mitsuhiro Hayashibe *Senior Member, IEEE*

***Abstract*—The most common cause of injuries among older adults is falling. Recently, there have been numerous developments in assistive and exoskeleton systems. However, comparatively little work is being done on systems that may help people to keep an upright position and avoid falling over. In this preliminary work, we investigate the feasibility of the wheel-legged robot as a balance-assist system for the people who cannot maintain balance and walk because of an injury, old age, or neurological or physical disorder. We perform motion stability analyses of the wheel-legged robot under different conditions such as system modeling errors, sensor noise, and external disturbances. The linear quadratic regulator (LQR) control approach is adopted for balancing, steering, and translational position control of the robot. To validate our control framework and visualize results, the robot is modeled and tested in the Gazebo simulator using ROS (Robot Operating System). Subsequently, the simulation results demonstrate the effectiveness of the LQR control method under the translational and rotational pushes of the wheel-legged system for human-robot interaction.**

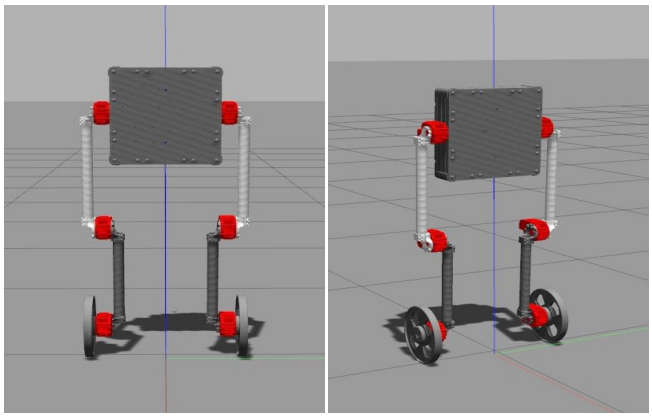
***Index Terms*—Balance Control, Wheel-legged Robots, Optimal Control, Self-balancing Robot.**

I. Introduction

The most common cause of injuries among older adults is falling [1]. These injuries not only affect the physical health of patients but also place an appreciable financial burden on governmental healthcare budgets. Fall-induced injuries are a major source of longstanding pain, physical disability, and death among older adults [2].

In recent years, numerous performance-augmenting and rehabilitation exoskeleton systems have been developed by robotics companies and researchers for able-bodied people and physically disabled patients, respectively [3]. As an example, a leg exoskeleton developed for able-bodied people provides positive mechanical power to the ankle during push-off [4]. Ma *et. al.* designed a wheeled-foot exoskeleton for paraplegic patients, allowing users to walk with a standing posture [5]. An assistive intelligent walker is developed to help the elderly, handicapped, or blind people by employing servo brakes [6]. A team from UC Berkeley developed a lower-extremity exoskeleton called BLEEX (Berkeley Lower Extremity Exoskeleton) [7]. It is the first field-operational exoskeleton that helps users carry an appreciable load with minimal effort.

As illustrated by the above mentioned works, there has been vast interest in developing robotic systems for performance augmentation, and the rehabilitation of people. However to the best of the authors' knowledge, there has been comparatively little work on developing systems that may help people to keep an upright position and avoid falling over. A rather rare example of the balance-assist systems is a lower body exoskeleton that consists of powered hip flexion/extension (HFE) and hip abduction/adduction (HAA) joints to support walking and balancing [8]. We therefore investigate the feasibility of a wheel-legged biped robot that helps prevent older adults and paraplegic patients from falling. Adaptability to various terrain is of critical importance to human-assistance robotic systems, as users live in different environments. We know that legged locomotion is robust in terms of climbing stairs and traversing rough terrain, whereas wheel mechanisms are energy efficient for traversing flat terrain. A hybrid wheel-legged robotic system has the advantages of both leg and wheel mechanisms, and it adapts its locomotion to the requirements of given terrain. Moreover, a biped wheel-legged system provides the complex balancing reactions needed to keep the center of mass (CoM) within the base of support of a human body.



(a) Front View. (b) Side View.

Fig. 1: Wheel-legged robot in the Gazebo simulator.

Wheel-legged systems have recently caught the attraction of the search and rescue, rapid exploration, and logistics robotics community; a fascinating example of such systems is a wheeled biped robot called Handle manufactured by the Boston Dynamics [9]. While Handle has shown remarkable balancing and manipulation feats, yet its locomotion and control frameworks have not been shared with the open research community. Another two-wheeled human assistant robot with sitting and running capabilities was proposed by [10]. To take the advantage of both wheeled and legged locomotion, a small wheel-legged robot PAW is presented by [11]. They designed the robot in such a way that its legs are capable of limited recirculation helping it to gain a cruise speed of up to 2 m/s. Suzumura *et. al.* modeled a quadruped wheel-legged mobile robot in three dimensions and

F. Raza and D. Owaki are with the Department of Robotics, Graduate School of Engineering, Tohoku University, Sendai 980-8579, Japan (email: r.fahad@neuro.mech.tohoku.ac.jp; owaki@tohoku.ac.jp).

M. Hayashibe is concurrently with the Graduate School of Biomedical Engineering and the Department of Robotics, Graduate School of Engineering, Tohoku University, Sendai 980-8579, Japan (email: hayashibe@tohoku.ac.jp).

Parameters	Description
m_c	Mass of the robot body
m_w	Mass of the robot wheel
Σ_I	Inertial coordinate frame
Σ_R	Robot fixed coordinate frame
Σ_C	Body (CoM) fixed coordinate frame
l	Height of the CoM from the origin of Σ_R
I_c	Inertia matrix of the robot body taken at its CoM
I_{wa}	Moment of inertia of the robot wheel about the wheel axis
I_{wd}	Moment of inertia of the robot wheel about its diameter
I_{ma}	Moment of inertia of the motor rotor about its axis
I_{md}	Moment of inertia of the motor rotor about its diameter
g	Gravitational acceleration constant
r	Wheel radius
b	Distance between the origin of Σ_R and the center of the wheel
α	Steering (yaw: angle between X_I and X_R) angle of the robot
β	Inclination (pitch: angle between Z_R and Z_C) angle of the CoM
γ	Gear reduction ratio
θ_r, θ_l	Angular displacements of left and right wheels
τ_r, τ_l	Motor torques of left and right wheels
c_r , c_l	Viscosity coefficients of left and right wheels
${}_I x_r$, ${}_I y_r$	X and Y positions of the origin of Σ_R in the inertial frame
p	Linear position of the robot frame Σ_R from O_I
v	Forward velocity of the robot in the X_R direction

TABLE I: System Parameters.

achieved high mobility control [12]. They showed that a zero-phase low-pass filter method generates a CoM pattern faster than the preview control approach. In addition, a research team from IIT Italy derived first order inverse kinematics and motion control scheme for a quadruped wheel-legged system without making any assumption on wheel camber angle [13]. More recently, Viragh *et. al.* developed trajectory optimization algorithm enabling the ANYmal robot fitted with wheels to walk and drive simultaneously [14].

It is clear that all the above mentioned researches focused on the design and control of wheel-legged systems. On the other hand, the main objective of this study is to evaluate the feasibility of the wheel-legged robot under different disturbances potentially occur during the use of the robot with humans. For this purpose, we start with studying the implications of the system modeling errors, external disturbances, and sensor noise on the motion stability of an off-the-shelf wheel-legged robot called Igor from Hebi Robotics [15]. We model the robot in the ROS-Gazebo environment and examine the effects of not only translational push disturbances but also rotational push disturbances on the robot, which are not well reported in the previous works.

II. Dynamic Modeling

Before designing the controller, we present the system dynamic model of the wheel-legged robot (Fig. 1). In our dynamic modeling of the robot, we assume a simplified inverted pendulum, where the body of the robot acts as a point mass. The origins of the robot-fixed frame Σ_R and body-fixed frame Σ_C are coincident and placed at the midpoint of the wheel axle. For all rotations, the counterclockwise direction is considered positive. The three-dimensional model of the robot is represented in Fig.2.

A. Equations of Motion

We use the Euler-Lagrange formulation to derive the equations of motion for the wheeled biped robot under the following important assumptions.

- There is no slip between the wheels of the robot and ground.
- The wheels are rigid nondeformable disks.
- The structure of the robot is symmetrical.
- The legs of the robot are massless.

The first two assumptions provide the necessary conditions for the kinematic modeling of the differential drive base of the robot.

We now begin by writing equations for the position of the center of mass (CoM) in inertial frame Σ_I. A left subscript is used from here onwards to clearly specify the coordinate frame of the vector quantities. The equations are

$$\begin{aligned} {}_I x_c &= {}_I x_r + l\cos(\alpha)\sin(\beta), \\ {}_I y_c &= {}_I y_r + l\sin(\alpha)\sin(\beta), \\ {}_I z_c &= r + l\cos(\beta). \end{aligned} \tag{1}$$

We obtain the translational velocity vector of the CoM of the robot simply by taking the time derivative of the above equations:

$${}_I V_c = \begin{bmatrix} {}_I\dot{x}_r + l\left(\dot{\beta}\cos(\alpha)\cos(\beta) - \dot{\alpha}\sin(\alpha)\sin(\beta)\right) \\ {}_I\dot{y}_r + l\left(\dot{\beta}\sin(\alpha)\cos(\beta) + \dot{\alpha}\cos(\alpha)\sin(\beta)\right) \\ -l\dot{\beta}\sin(\beta) \end{bmatrix}. \tag{2}$$

The rotational velocity of the robot frame Σ_R with respect to inertial frame Σ_I is given as

$${}_I\Omega_R = \begin{bmatrix} 0 \\ 0 \\ \dot{\alpha} \end{bmatrix}. \tag{3}$$

We take the rotational velocity of the CoM in the body-fixed Σ_C frame to get a diagonal inertia matrix and simplify our

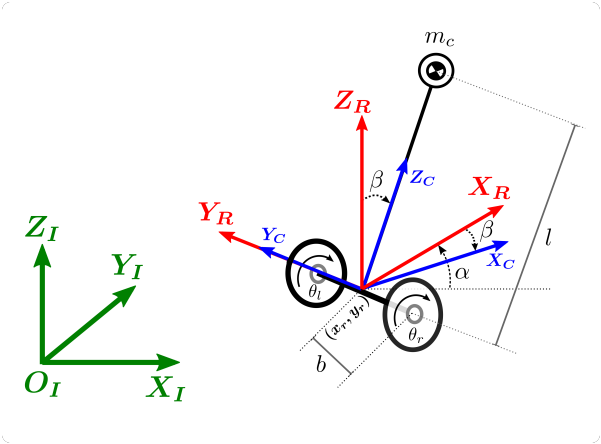


Fig. 2: Dynamic model of the robot.

kinetic energy equation:

$$ {}_C\Omega_c = \begin{bmatrix} -\dot{\alpha}\sin(\beta) \\ \dot{\beta} \\ \dot{\alpha}\cos(\beta) \end{bmatrix}. \tag{4} $$

The translational and rotational kinetic energy of the CoM is calculated as

$$ \begin{aligned} T_c &= \frac{1}{2} m_c \, {}_I V_c^T \, {}_I V_c + \frac{1}{2} \, {}_C\Omega_c^T \, I_c \, {}_C\Omega_c \\ &= \frac{1}{2} m_c \Big[{}_I\dot{x}_r^2 + 2 \, {}_I\dot{x}_r l \big(\dot{\beta}\cos(\alpha)\cos(\beta) - \dot{\alpha}\sin(\alpha)\sin(\beta)\big) \\ &\quad + {}_I\dot{y}_r^2 + 2 \, {}_I\dot{y}_r l \big(\dot{\beta}\sin(\alpha)\cos(\beta) + \dot{\alpha}\cos(\alpha)\sin(\beta)\big) \\ &\quad + (l\dot{\beta})^2 + l^2\dot{\alpha}^2\sin^2(\beta) \Big] \\ &\quad + \frac{1}{2}\big(I_{xx}\dot{\alpha}^2\sin^2(\beta) + I_{yy}\dot{\beta}^2 + I_{zz}\dot{\alpha}^2\cos^2(\beta)\big). \end{aligned} \tag{5} $$

The translational and rotational kinetic energy of the two wheels is given by

$$ \begin{aligned} T_w &= \frac{1}{2} m_w r^2 \dot{\theta}_r^2 + \frac{1}{2} m_w r^2 \dot{\theta}_l^2 \\ &\quad + \frac{1}{2}\big(I_{wa} + I_{ma}\gamma^2\big)\big(\dot{\theta}_r^2 + \dot{\theta}_l^2\big) + \big(I_{wd} + I_{md}\big)\dot{\alpha}^2. \end{aligned} \tag{6} $$

The potential energy of the wheel is considered zero, as we assume the wheel remains in contact with the ground everywhere. Consequently,

$$ U_w = 0. \tag{7} $$

The potential energy of the CoM is

$$ U_c = m_c g l \cos(\beta), \tag{8} $$

where g is the gravitational acceleration constant.
Rayleigh's dissipation function relating to viscous forces is given by

$$ F = \frac{1}{2} c_r \big(\dot{\theta}_r - \dot{\beta}\big)^2 + \frac{1}{2} c_l \big(\dot{\theta}_l - \dot{\beta}\big)^2. \tag{9} $$

The Lagrangian of a mechanical system is described by taking the difference between kinetic and potential energies of the system:

$$ \begin{aligned} L &= K.E - P.E \\ &= T_c + T_w - U_c. \end{aligned} $$

The equations of motion are then calculated by solving

$$ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} + \frac{\partial F}{\partial \dot{q}_i} = Q_i, \tag{10} $$

where q_i represents generalized coordinates and Q_i generalized forces.
For the wheel-legged robot, we choose the generalized coordinate vector $q = [{}_I x_r \; {}_I y_r \; \alpha \; \beta \; \theta_r \; \theta_l]^T$. After solving Eq. (10) by substituting for L and F, motion equations of the robot are expressed in familiar matrix form as

$$ M(q)\ddot{q} + V\dot{q} + H(q,\dot{q}) + G = E\tau, \tag{11} $$

where $M(q) \in \mathbb{R}^{6\times6}$ is the inertia matrix, $V \in \mathbb{R}^{6\times6}$ is a square matrix of viscous friction terms, $H(q,\dot{q}) \in \mathbb{R}^6$ is a vector of Coriolis and centrifugal terms, $G \in \mathbb{R}^6$ is a vector of the gravitational force, $E \in \mathbb{R}^{6\times2}$ is the torque selection matrix, and $\tau \in \mathbb{R}^2$ is the vector of control torques. These are expressed as

$$ M(q) = \begin{bmatrix} m_c & 0 & -c_4 & c_5 & 0 & 0 \\ 0 & m_c & c_3 & c_6 & 0 & 0 \\ -c_4 & c_3 & c_1 & 0 & 0 & 0 \\ c_5 & c_6 & 0 & I_{yy} + m_c l^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & c_2 \end{bmatrix}, $$

$$ V = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_r + c_l & -c_r & -c_l \\ 0 & 0 & 0 & -c_r & c_r & 0 \\ 0 & 0 & 0 & -c_l & 0 & c_l \end{bmatrix}, $$

$$ H(q,\dot{q}) = \begin{bmatrix} -(c_3\dot{\alpha}^2 + c_3\dot{\beta}^2 + 2c_6\dot{\alpha}\dot{\beta}) \\ -(c_4\dot{\alpha}^2 + c_4\dot{\beta}^2 - 2c_5\dot{\alpha}\dot{\beta}) \\ c_7\dot{\alpha}\dot{\beta} \\ -\frac{c_7}{2}\dot{\alpha}^2 \\ 0 \\ 0 \end{bmatrix}, $$

$$ G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -m_c g l sin(\beta) \\ 0 \\ 0 \end{bmatrix}, $$

$$ E = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -1 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \tau = \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix}, $$

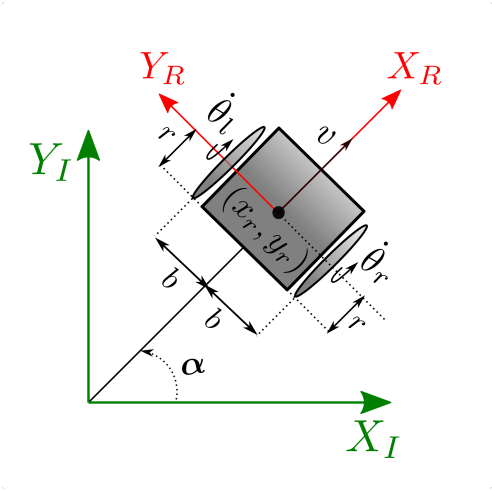


Fig. 3: Top view of differential drive robot.

where $c_1, c_2, \ldots, c_7$ are

$$
\begin{aligned}
c_1 &= 2\big(I_{wd} + I_{md}\big) + I_{xx} + \big(I_{zz} - I_{xx} - m_c l^2\big)\cos^2(\beta) \\
&\quad + m_c l^2, \\
c_2 &= m_w r^2 + I_{wa} + I_{ma}\gamma^2, \quad c_3 = m_c l \cos(\alpha)\sin(\beta), \\
c_4 &= m_c l \sin(\alpha)\sin(\beta), \quad c_5 = m_c l \cos(\alpha)\cos(\beta), \\
c_6 &= m_c l \sin(\alpha)\cos(\beta), \quad c_7 = \big(I_{xx} + m_c l^2 - I_{zz}\big)\sin(2\beta).
\end{aligned}
$$

B. Non-holonomic Constraints

It is well known that Eq. (11) is only valid for a system without non-holonomic constraints. Therefore, before any standard state-space based control design can be applied, an alternative approach is needed to represent the motion and constraints of the system [16].

Under rolling and no-slip conditions that engender non-holonomic constraints, the kinematic equations of the differential drive robot shown in Fig. 3 are [17]

$$
{}_I\dot{x}_r \sin(\alpha) - {}_I\dot{y}_r \cos(\alpha) = 0 \tag{12}
$$

$$
{}_I\dot{x}_r \cos(\alpha) - {}_I\dot{y}_r \sin(\alpha) = \frac{r}{2}\big(\dot{\theta}_r + \dot{\theta}_l\big) \tag{13}
$$

$$
\dot{\alpha} = \frac{r}{2b}\big(\dot{\theta}_r - \dot{\theta}_l\big) \tag{14}
$$

The above three equations can be written in matrix form as $A(q)\dot{q} = 0$, where

$$
A(q) = \begin{bmatrix} -\sin(\alpha) & \cos(\alpha) & 0 & 0 & 0 & 0 \\ \cos(\alpha) & \sin(\alpha) & b & 0 & -r & 0 \\ \cos(\alpha) & \sin(\alpha) & -b & 0 & 0 & -r \end{bmatrix}. \tag{15}
$$

Once we have equations for the constraints of the system, Eq. (11) is modified to account for k kinematic constraints as

$$
M(q)\ddot{q} + V\dot{q} + H(q,\dot{q}) + G = E\tau + A(q)^T\lambda, \tag{16}
$$

where $\lambda \in \mathbb{R}^k$ is a vector of Lagrange multipliers. The term $A(q)^T\lambda$ represents the vector of reaction forces at the generalized coordinate level.

We now use the standard method to remove the Lagrange multipliers from Eq. (16) as elaborated in [18]. We first define a full-rank matrix $S(q) \in \mathbb{R}^{n\times m}$ that lies in the null-space of matrix $A(q)$; here, $m = n - k$, and n is the number of generalized coordinates. It is noted that the choice of matrix $S(q)$ is not unique.

$$
S(q) = \begin{bmatrix} 0 & \cos(\alpha) & 0 \\ 0 & \sin(\alpha) & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & \frac{1}{r} & \frac{b}{r} \\ 0 & \frac{1}{r} & \frac{-b}{r} \end{bmatrix}. \tag{17}
$$

A new vector $u \in \mathbb{R}^m$ of *pseudo-velocities* is defined such that

$$
\dot{q} = S(q)u. \tag{18}
$$

In the literature, Eq. (18) is mentioned as the kinematic model of the constrained mechanical system. In this study, we select $u = [v\ \dot{\alpha}\ \dot{\beta}]^T$ so that the controller design can be achieved rather simply. Differentiation of the above equation leads to

$$
\ddot{q} = \dot{S}(q)u + S(q)\dot{u}. \tag{19}
$$

The substitution of Eqs. (18) and (19) into Eq. (16) yields

$$
\begin{aligned}
M(q)\dot{S}(q)u + M(q)S(q)\dot{u} + VS(q)u + H(q,\dot{q}) + G \\
= E\tau + A(q)^T\lambda.
\end{aligned} \tag{20}
$$

Finally, the Lagrange multipliers can be eliminated by premultiplying both sides of Eq. (20) by $S(q)^T$ for the reason that $S(q)A(q) = 0$, and we get the reduced dynamic model of m differential equations:

$$
\hat{M}(q)\dot{u} + \hat{V}u + \hat{H}(u,\dot{u}) + \hat{G} = \hat{E}\tau, \tag{21}
$$

where $\hat{M}(q) = S(q)^T M(q) S(q)$ is positive definite, $\hat{V} = S(q)^T V S(q)$, $\hat{H}(u,\dot{u}) = S(q)^T\big[M(q)\dot{S}(q)u + H(q,\dot{q})\big]$, $\hat{G} = S(q)^T G$, and $\hat{E} = S(q)^T E$.

Equation (21) is a set of nonlinear equations that cannot be used in designing a linear controller, and we thus have to linearize Eq. (21) about the equilibrium point ($\beta = 0$). Making a small-angle approximation results in

$$
\begin{aligned}
\sin(*) &\simeq (*), \\
\sin^2(*) &\simeq 0, \\
\cos(*) &\simeq 1, \\
(\dot{*})^2 &\simeq 0, \\
(*)(\dot{*}) &\simeq 0.
\end{aligned}
$$

By virtue of the above conditions, $\hat{H}$ is eliminated from Eq. (21) and we finally get

$$
\hat{M}(q)\dot{u} + \hat{V}u + \hat{G} = \hat{E}\tau, \tag{22}
$$

where $\hat{M}(q)$ and $\hat{G}$ have linearized elements. Furthermore, the above equation implies that

$$
\begin{aligned}
\dot{u} &= \hat{M}^{-1}\big[\hat{E}\tau - \hat{V}u - \hat{G}\big] \\
&= -\hat{M}^{-1}\big(\hat{V}u + \hat{G}\big) + \hat{M}^{-1}\hat{E}\tau.
\end{aligned} \tag{23}
$$

It is noted that θ_r and θ_l are completely decoupled from other state variables. Additionally we find $\dot{\theta}_r$ and $\dot{\theta}_l$ given the

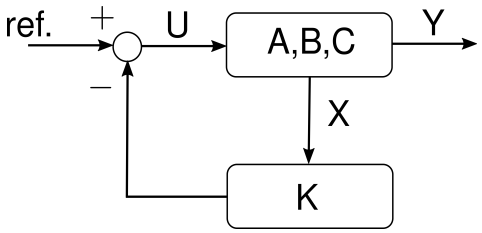


Fig. 4: State-feedback Control Scheme.

forward velocity $v = \frac{r}{2}(\dot{\theta}_r + \dot{\theta}_l)$ and Eq. (14). We can therefore now reduce the order of the system and define a configuration vector $q_r = [p \ \alpha \ \beta]^T$ as an actual control variable, where $p = {}_I x_r \cos(\alpha) + {}_I y_r \sin(\alpha)$.

C. State-space Modeling

State-space modeling is generally adopted to convert N^{th}-order system dynamics to a system of N first-order differential equations. We can formulate the state-space model as we have already obtained linearized equations of motion for the wheel-legged robot. A general linear-time-invariant state-space model is represented as

$$\begin{aligned} \dot{X} &= AX + BU \\ Y &= CX + DU, \end{aligned} \tag{24}$$

where $X \in \mathbb{R}^n$ is called the state vector, $Y \in \mathbb{R}^i$ is the output vector of the system, and $U \in \mathbb{R}^p$ is the system input. The constant matrices A, B, C, and D are respectively called dynamics, input, output, and feedforward matrices.
In this study, we define the state vector as

$$X = \begin{bmatrix} q_r \\ u \end{bmatrix}_{6\times 1}$$

and the control input as $U = \tau$. The state-space model of the robot is therefore given by

$$\dot{X} = \begin{bmatrix} \dot{q}_r \\ \dot{u} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{H_1}{D_1} & \frac{-H_2}{D_1} & \frac{H_3}{D_1} & \frac{H_4}{D_1} \\ 0 & 0 & 0 & b\frac{a_5}{D_2} & -b^2\frac{a_4}{D_2} & -br\frac{a_5}{D_2} \\ 0 & 0 & \frac{H_5}{D_3} & \frac{H_6}{D_3} & \frac{-H_7}{D_3} & \frac{-H_8}{D_3} \end{bmatrix} X + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ e_1 & e_1 \\ e_2 & -e_2 \\ e_3 & e_3 \end{bmatrix} \tau, \tag{25}$$

where

$$\begin{aligned} a_1 &= I_{wa} + I_{ra}\gamma^2 + m_w r^2, \\ a_2 &= I_{rd} + I_{wd} + 0.5 I_{zz} + m_w b^2, \\ a_3 &= I_{wa} + I_{ra}\gamma^2, \quad a_4 = c_r + c_l, \\ a_5 &= c_l - c_r, \quad a_6 = I_{yy} + m_c l^2 + m_c l r, \\ a_7 &= 2a_3 + r^2(m_c + 2m_w) + m_c l r, \end{aligned}$$

Parameters	Value	Unit
m_c	5.92	kg
m_w	0.35	kg
l	0.65	m
I_{xx}	0.0774	$kg.m^2$
I_{yy}	0.0454	$kg.m^2$
I_{zz}	0.0454	$kg.m^2$
I_{wa}	0.0018	$kg.m^2$
I_{wd}	0.000929	$kg.m^2$
I_{ma}	0.0	$kg.m^2$
I_{md}	0.0	$kg.m^2$
g	9.81	m/s^2
r	0.1016	m
b	0.25	m
γ	1	
c_r, c_l	0.1	Nm/(rad/sec)

TABLE II: Simulation parameters of the robot.

$$\begin{aligned} e_1 &= \frac{a_6 r}{D_1}, \quad e_2 = \frac{br}{D_2}, \quad e_3 = -\frac{a_7 r}{D_3}, \\ D_1 &= 2a_1\big(I_{yy} + m_c l^2\big) + I_{yy} m_c r^2, \\ D_2 &= 2\big(a_2 r^2 + a_3 b^2\big), \\ D_3 &= 2r\big[I_{yy}(a_1 + 0.5 m_c r^2) + a_1 m_c l^2\big], \\ H_1 &= -g(m_c l r)^2, \ \ H_2 = a_4 a_6, \\ H_3 &= a_5 a_6 b, \ \ H_4 = a_4 a_6 r, \\ H_5 &= g m_c l r[2a_3 + r^2(m_c + 2m_w)], \ \ H_6 = a_4 a_7, \\ H_7 &= a_5 a_7 b, \ \ H_8 = a_4 a_7 r. \end{aligned}$$

For a full-state-feedback system, the matrix C is an identity matrix and $D = 0$ if the system has no feedforward.

III. Optimal Control

In a simple state-feedback controller, we choose the closed-loop poles of a system by trial and error; however, these pole locations usually do not provide the optimal control performance. To overcome this problem, we design an LQR controller for our system, which allows us to find the optimal feedback gain vector K_{lqr}. The LQR places the poles in such a way that the closed-loop system optimizes the cost function

$$J_{lqr} = \int_0^\infty [X(t)^T Q X(t) + U(t)^T R U(t)] \ dt, \tag{26}$$

where $X^T Q X$ is the state cost with weight Q, while $U^T R U$ is the control cost with weight R [19]. The optimal state-feedback law is then given as

$$U = ref. - K_{lqr} X. \tag{27}$$

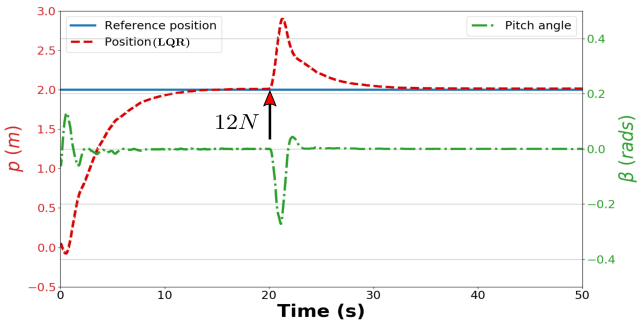
Here the necessary condition for optimality is

$$K_{lqr} = R^{-1} B^T P, \tag{28}$$

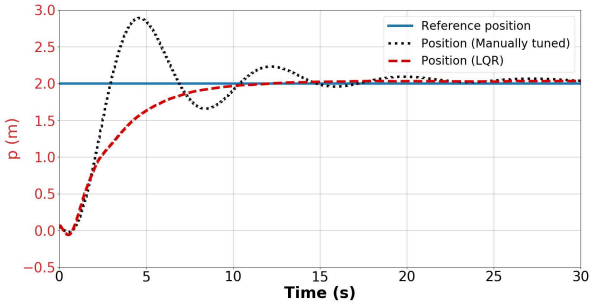
where P is calculated by solving the algebraic Riccati equation

$$0 = A^T P + P A + Q - P B R^{-1} B^T P. \tag{29}$$

It is also important to choose Q and R matrices carefully to



(a) LQR step response and translational push of 12 N.



(b) Manually tuned controller vs. LQR.

Fig. 5: Translational position step response.

get the best possible results. We use the well-known Bryson's method [20], which gives a simple and reasonable choice to determine Q and R as

$$Q = \begin{bmatrix} \frac{{w_1}^2}{(x_{11})^2_{max}} & 0 & \dots & 0 \\ 0 & \frac{{w_2}^2}{(x_{22})^2_{max}} & \dots & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \dots & \frac{{w_n}^2}{(x_{nn})^2_{max}} \end{bmatrix}, \tag{30}$$

where $(x_{ii})_{max}$ denotes the largest desired response for that component of the state vector. R is determined as

$$R = \rho \begin{bmatrix} \frac{{b_1}^2}{(u_{11})^2_{max}} & 0 & \dots & 0 \\ 0 & \frac{{b_2}^2}{(u_{22})^2_{max}} & \dots & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \dots & \frac{{b_j}^2}{(u_{jj})^2_{max}} \end{bmatrix}. \tag{31}$$

Here, $(u_{jj})_{max}$ is the maximum desired control input for the system and ρ is used as the last relative weighting between the control and state penalties. Furthermore, w_i and b_i are respectively the weights for each state and control input. We define w_i and b_i according to

$$\sum_{i=1}^{n} {w_i}^2 = 1,$$
$$\sum_{i=1}^{j} {b_i}^2 = 1.$$

Even though Bryson's method usually yields satisfactory results, it is often just the beginning of a trial-and-error iterative method of obtaining the desired closed-loop system response. Using Bryson's rule and then making alterations, we have $Q = diag([0.78\ \ 6.37\ \ 39.06\ \ 0.25\ \ 0.19\ \ 11.11])$, $\rho = 1$, and $R = diag([0.03\ \ 0.03])$ which leads to the satisfactory reference tracking of the system states. Finally, the optimal feedback gain

$$K_{lqr} = \begin{bmatrix} -3.53 & 10.10 & -52.50 & -7.86 & 1.66 & -17.87 \\ -3.53 & -10.10 & -52.50 & -7.86 & -1.66 & -17.87 \end{bmatrix}$$

is acquired using lqr command in *MATLAB (Mathworks Inc., Natick, MA, USA)* with system parameters listed in Table II. As the robot steers its heading using a differential drive mechanism, we note that the corresponding gain values for α and $\dot{\alpha}$ states in K_{lqr} have opposite signs.

Mass Change	Noise $\sigma = 0.0$			Noise $\sigma = 0.02$		
	p_{err}	α_{err}	β_{err}	p_{err}	α_{err}	β_{err}
-20%	13.72E-3	19.33E-5	73.77E-5	17.84E-3	9.11E-5	74.98E-5
-10%	25.25E-3	18.94E-5	23.91E-5	28.27E-3	21.7E-5	48.87E-5
0%	36.54E-3	22.49E-5	18.11E-5	36.63E-3	18.6E-5	39.15E-5
+10%	42.07E-3	25.04E-5	54.38E-5	44.97E-3	21.28E-5	68.07E-5
+20%	50.11E-3	28.74E-5	86.19E-5	52.19E-3	29.02E-5	92.32E-5

TABLE III: Steady-state errors in the presence of model uncertainties and Gaussian noise.

IV. Simulation and Results

A. ROS–Gazebo Simulation

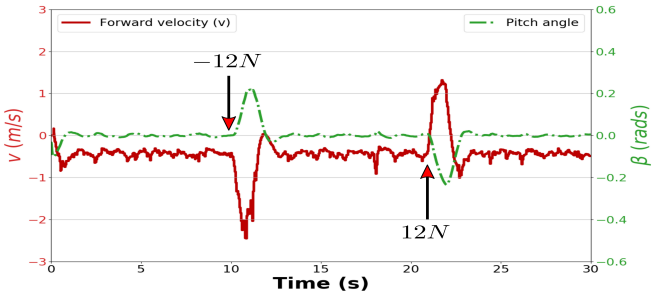
With the purpose of simulating our wheeled biped robot in the *Gazebo* simulator [21], we modeled our robot in URDF (Universal Robotic Description Format) using the dimensions of the Igor robot as shown in Fig. 1. The URDF model includes all the physical constraints of the real robot, such as joint limits, static friction, and damping coefficients, and the actuator's torque-speed characteristics. *Gazebo* is an open-source simulator that uses *ODE* [22] as its physics engine. ROS (Robot Operating System) is used to implement the control algorithm in C++, thus controlling the robot in *Gazebo*. The *ROS-Gazebo* simulation runs at a frequency of 1 kHz with a motion controller steering the wheeled biped robot in the desired direction while keeping the robot in an upright position at the same time.

B. Results and Discussion

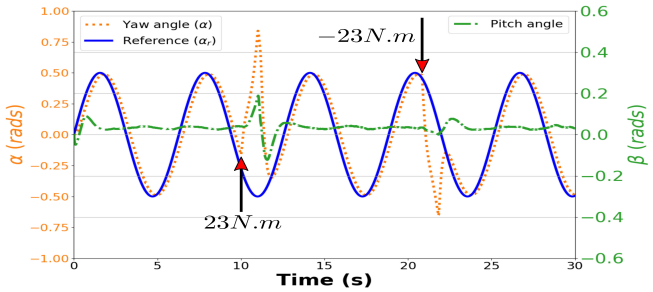
A series of tests are conducted to validate the effectiveness of the designed controller. Simple reference trajectories comprising step and sinusoidal signals are chosen for the robot. Given that an off-the-shelf robot employs a simple proportional-integral-derivative controller for balancing, we considered it necessary to compare the designed LQR with a manually tuned state-feedback controller of gain

$$K_{sf} = \begin{bmatrix} -1.09 & 0.51 & -21.37 & -0.87 & 0.03 & -8.26 \\ -1.09 & -0.51 & -21.37 & -0.87 & -0.03 & -8.26 \end{bmatrix}.$$

This is an important point of the study since the off-the-shelf robot is not using the LQR controller, thus by this comparison, we can check how much the disturbance rejection capability of the current robot can be potentially improved for the future



(a) Forward and backward push during translational motion.



(b) Counterclockwise and clockwise moment during yaw tracking.

Fig. 6: LQR tracking and disturbance rejection.

use. Results of *ROS-Gazebo* simulations are presented in Figs. 5 and 6.

Figure 5a shows the translational reference position, step response of the LQR, pitch angle, and system's recovery from a force of 12 N applied in the frontal plane at the CoM of the robot body. One can see that the robot leaned backwards because of the strong push but the controller brought the robot back to the reference point smoothly while keeping the pitch angle below 0.25 rad.

In another test, the results of which are shown in Fig. 5b, we compare the translational position step response of the optimal controller with the manually tuned state-feedback control method. It is clear from the simulation results that the LQR controller has a shorter settling time than the state-feedback controller. Most importantly, the LQR has no overshoot or oscillations, which is essential for human-robot interaction. Although state-feedback and LQR control laws have the same theoretical basis and one can argue that the state-feedback controller would perform better with some further fine tuning, the sole purpose of the comparison here is to determine how well our designed LQR performs against non-optimal off-the-shelf robot controllers.

To ensure the robustness of the LQR controller against the model uncertainties and sensor noise, various tests are performed and the results are summarized in Table III. To account for model uncertainties, we change the masses of the body m_c and wheel m_w along with the corresponding moment of inertia by $\pm 10\%$ and $\pm 20\%$. Two series of tests are conducted, namely tests without sensor noise and tests including Gaussian white noise with standard deviation $\sigma = 0.02$. Finally, steady-state errors for reference positions $p_r = 2$ meters, $\alpha_r = 0.78$ rad, and $\beta_r = 0$ rad are determined. The results clearly show that regardless of sensor noise, the steady-state error for the

External Force	Mass Change = 0%				Mass Change = -20%			
	LQR		Manual Tuning		LQR		Manual Tuning	
	$\int\|e_v\|dt$	t_r	$\int\|e_v\|dt$	t_r	$\int\|e_v\|dt$	t_r	$\int\|e_v\|dt$	t_r
12 N	1.660	2.8	4.117	5	1.597	3	4.782	6
8 N	1.038	2.6	2.664	4	1.169	3	3.093	5.9
4 N	0.549	2.4	1.439	3.4	0.620	2	1.517	4.8
External Torque	$\int\|e_\alpha\|dt$	t_r	$\int\|e_\alpha\|dt$	t_r	$\int\|e_\alpha\|dt$	t_r	$\int\|e_\alpha\|dt$	t_r
21 N.m	0.578	1.6	N/A	N/A	0.582	1.6	N/A	N/A
14 N.m	0.362	1.5	4.873	4	0.356	1.5	N/A	N/A
7 N.m	0.206	1.2	2.772	2.8	0.206	1.3	2.924	2.8

TABLE IV: Disturbance rejection comparison of the LQR and manually tuned controller. Forces and moments are applied at the CoM of the body during forward motion and sinusoidal yaw tracking, respectively. e_v, e_α, and t_r are respectively the linear velocity error, yaw angle error, and recovery time. N/A indicates that the robot fell as a result of an external disturbance.

translational position tracking decreases as the mass of the system is reduced. This should not come as surprise because it is conventional behavior of a proportional feedback control method. We also find that the pitch angle (β) tracking is sensitive to the model irregularities and the steady-state error can increase by a factor of 4 with a mere change of $\pm 20\%$ in the system mass. Meanwhile, we observe that yaw tracking of the robot improves as we reduce the mass and inertia of the system; however, this difference is not large.

Figure 6 depicts the robot's recovery from rather strong external disturbances during translational and rotational movements. Figure 6a shows the application of two external forces on the body while the robot is moving at constant linear velocity $v = -0.5$ m/s. The first force is applied in the robot's direction of motion while the second force is applied in the direction opposing the robot's direction of motion. Similarly, Fig. 6b shows the application of external torques about the z-axis of the robot body as rotational disturbance. Different disturbances are applied on the system and we find that the peak translational and rotational push that the robot can sustain with the current LQR is 12 N and 23 N.m, respectively.

The values in Table IV are used to quantitatively demonstrate the disturbance rejection of the LQR and compare it with the disturbance rejection of the manually tuned state-feedback controller. All errors are taken as the area under the curve from the time a disturbance is applied till the moment the robot recovers to its steady state. The table shows that the LQR has smaller errors and a shorter recovery time than the state-feedback controller. Additionally, the LQR endures much higher external torques as compared with the manually tuned controller. Furthermore, we have also tested the system's disturbance rejection capability in case of overestimation of the mass and inertia values during controller design process. As system becomes lighter, it gets more prone to external disturbances. Simulation results point out that the error, and recovery time increase slightly in case of translational disturbances for the LQR as compared to the manually tuned controller, where the change is more significant. On the other hand, alterations in the mass and moment of inertia did not effect the performance of the controllers during rotational disturbances.

Finally, to check whether the LQR controller can handle the translational and rotational accelerations simultaneously, we

Fig. 7: Igor performing a slalom in the *Gazebo* simulator.

perform a slalom maneuver as shown in Fig. 7. We observe that the robot can carry out the slalom easily at peak forward velocity of 1 m/s.

V. Conclusion

We performed dynamic modeling of a self-balancing robot using the Euler-Lagrange method to control the translation, rotation, and pitch of the robot. Furthermore, non-holonomic constraints due to the differential-drive wheel system were integrated into the mathematical model of the robot. A linear quadratic regulator based on the mathematical model was then designed for the motion control of the robot. The objective of this study is to simulate and devise the control law for the wheel-legged robot ahead of implementing the aforementioned control strategy on the actual system. Instead of relying on differential-equation based *MATLAB* simulations to verify our controller of choice, we used the physics-engine based *Gazebo* simulator because it includes rigid-body dynamics, collision detection, and friction and thus provides a better approximation of the real physical system. One advantage of our approach is the immediate application of it on the off-the-shelf Igor robot. Several simulation test runs were carried out in the presence of model uncertainties, external disturbances, and sensor noise to quantitatively investigate the robustness of the motion controller in detail. Later, a manually tuned state-feedback controller was designed and compared with the LQR. As a result, it became clear that the robot with the LQR endures almost twofold stronger external torques than with the simpler manually tuned controller. Thanks to the LQR controller, the robot can track reference trajectories within a short time period. Also, the robot performs extremely well under the sensor noise, and external translational and rotational perturbations of up to 12 N and 21 N.m, respectively.

As future work, we plan to conduct a series of experiments on a real robot. We will also combine our controller with a full-motion planning algorithm and measure the performance of the controller for complex maneuvers. Additionally, it would be interesting to evaluate the extent to which acceleration of the knee joints mitigates external disturbances.

VI. Availability of data and material

All relevant data are within the paper

VII. Competing interests

The authors declare that they have no competing interests.

VIII. Funding

Not applicable.

IX. Author's Contributions

FR and MH designed the study, performed the analyses, and helped in the manuscript writing. FR designed and performed the simulations. MH and DO contributed to the discussion of the results. All authors read and approved the final manuscript.

X. Acknowledgements

Not applicable.

References

[1] E. R. Burns, J. A. Stevens, and R. Lee, "The direct costs of fatal and non-fatal falls among older adults - united states." *Journal of Safety Research*, vol. 58, pp. 99–103, 2016.

[2] P. Kannus, H. Sievänen, M. Palvanen, T. Järvinen, and J. Parkkari, "Prevention of falls and consequent injuries in elderly people," *The Lancet*, vol. 366, no. 9500, pp. 1885–1893, 2005.

[3] J. L. Pons, *Wearable Robots: Biomechatronic Exoskeletons*. Wiley, 2008.

[4] L. M. Mooney, E. J. Rouse, and H. M. Herr, "Autonomous exoskeleton reduces metabolic cost of human walking during load carriage," *Journal of NeuroEngineering and Rehabilitation*, 2014.

[5] Q. Ma, L. Ji, and R. Wang, "The development and preliminary test of a powered alternately walking exoskeleton with the wheeled foot for paraplegic patients," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 2, pp. 451–459, 2018.

[6] Y. Hirata, A. Hara, and K. Kosuge, "Motion control of passive intelligent walker using servo brakes," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 981–990, 2007.

[7] A. B. Zoss, H. Kazerooni, and A. Chu, "Biomechanical design of the berkeley lower extremity exoskeleton (bleex)," *IEEE/ASME Transactions on Mechatronics*, vol. 11, no. 2, pp. 128–138, 2006.

[8] T. Zhang, M. Tran, and H. Huang, "Design and experimental verification of hip exoskeleton with balance capacities for walking assistance," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 1, pp. 274–285, 2018.

[9] Boston Dynamics., "Introducing handle." [Online]. Available: https://www.youtube.com/watch?v=-7xvqQeoA8c

[10] S. Jeong and T. Takahashi, "Wheeled inverted pendulum type assistant robot: inverted mobile, standing, and sitting motions," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007, pp. 1932–1937.

[11] J. A. Smith, I. Sharf, and M. Trentini, "Paw: a hybrid wheeled-leg robot," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 4043–4048.

[12] A. Suzumura and Y. Fujimoto, "Real-time motion generation and control systems for high wheel-legged robot mobility." *IEEE Transactions On Industrial Electronics*, vol. 61, no. 7, pp. 3648–3659, 2014.

[13] M. Kamedula, N. Kashiri, and N. G. Tsagarakis, "On the kinematics of wheeled motion control of a hybrid wheeled-legged centauro robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 2426–2433.

[14] Y. de Viragh, M. Bjelonic, C. D. Bellicoso, F. Jenelten, and M. Hutter, "Trajectory optimization for wheeled-legged quadrupedal robots using linearized zmp constraints," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1633–1640, 2019.

[15] Hebi Robotics., "Igor." [Online]. Available: https://www.hebirobotics.com/robotic-kits

[16] Nilanjan Sarkar, Xiaoping Yun, and R. Vijay Kumar, "Control of mechanical systems with rolling constraints: Application to dynamic control of mobile robots," 1992.

[17] Alonzo Kelly, "A vector algebra formulation of kinematics of wheeled mobile robots," 2010.

[18] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer, 2009.

[19] K. J. Astrom and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2008.

[20] A. E. Bryson, *Applied Optimal Control: Optimization, Estimation and Control*. Hemisphere Publishing Corporation, 1975.

[21] Open Source Robotics Foundation., "Gazebo." [Online]. Available: http://gazebosim.org/

[22] S. Russell, "ODE – open dynamics engine." [Online]. Available: http://www.ode.org