Wheeled-Legged Balancing Robot Development NoteBook

Team 3
Gabriel Gao(ngao4)

Fall 2023
TA: Tianxiang Zheng

Contents

1 Motor Selection

1.1 Wheel Motors

1. Better for direct drive motor(without gearbox).
2. Output torque should be approximately linear and stable at the low speed.
3. Accept high power input.

The reason is that we want to use this motor as a "torque controller", which can decrease the complexity of the whole model. Gaps between the gear sets may affect the efficiency of the torque output and the accuracy of the returned data. Also, we don't want the output unstable, resulting in oscillating or even disrupted equilibrium states.

1.2 Leg Motors

1. Stable data communication at high voltage or high power.
2. Excellent heat dissipation.
3. The peak of output torque $\geq 20N \cdot m$.

The reason is that we want the motor can receive and send the data or signal when we have instantaneous large torque. The instantaneous large torque will lead to the instantaneous large power, which might interfere with the data signal. When we use the motor to output a large torque, we need to ensure its excellent heat dissipation to avoid burning it. When the robot is descending stairs, we need to have enough torque to counter the falling momentum of the robot.

2 Establish Physical Model

2.1 Assumption

To simplify the model of the wheeled-legged robot, we can split the complex motion into wheel motion and leg motion. Wheel motion should include planar motion(balance in the stationary state, and move forward and backward) and rotation motion. Leg motion includes height adjustment(support phase) and jumping(jump phase).

2.2 Variable and Parameter Declaration

Label	**Meaning**	**Unit**
x_L, x_R	The displacement of the left and right wheels.	m
z	The distance between the body's center of mass and wheel motor rotation axis along the z-axis.	m
ϕ	The roll angle of the body.	rad
θ	The pitch angle of the body.	rad
ψ	The yaw angle of the body.	rad
T_L, T_R	The output torque of the left and right wheel motors.	$N \cdot m$
T	The output torque of the leg motors.	$N \cdot m$
N_L, N_R	The horizontal component of the force between wheels and the body (along the x-axis).	N
P_L, P_R	The vertical component of the force between wheels and the body (along the y-axis).	N
F_L, F_R	The frictions of the wheels when moving.	N

Table 1: Variables

Label	**Meaning**	**Unit**
m	The mass of the rotor in the wheel motors.	kg
M	The mass of the body.	kg
I_w	The moment of inertia of the rotor in the wheel motors.	$kg \cdot m^2$
I_x	The moment of inertia of the body rotated around the x-axis.	$kg \cdot m^2$
I_y	The moment of inertia of the body rotated around the y-axis.	$kg \cdot m^2$
I_z	The moment of inertia of the body rotated around the z-axis.	$kg \cdot m^2$
R	The radius of the wheel.	m
l	The distance between the body's center of mass and the rotation axis of the wheel motor.	m
D	The distance between the left and right wheels.	m
g	The acceleration due to the gravity measured.	m/s^2

Table 2: Parameters

2.3 Classical Mechanical Analysis for Wheel Motion

NOTE: If you use the Lagrange equation, you can get the same equations in the below part.

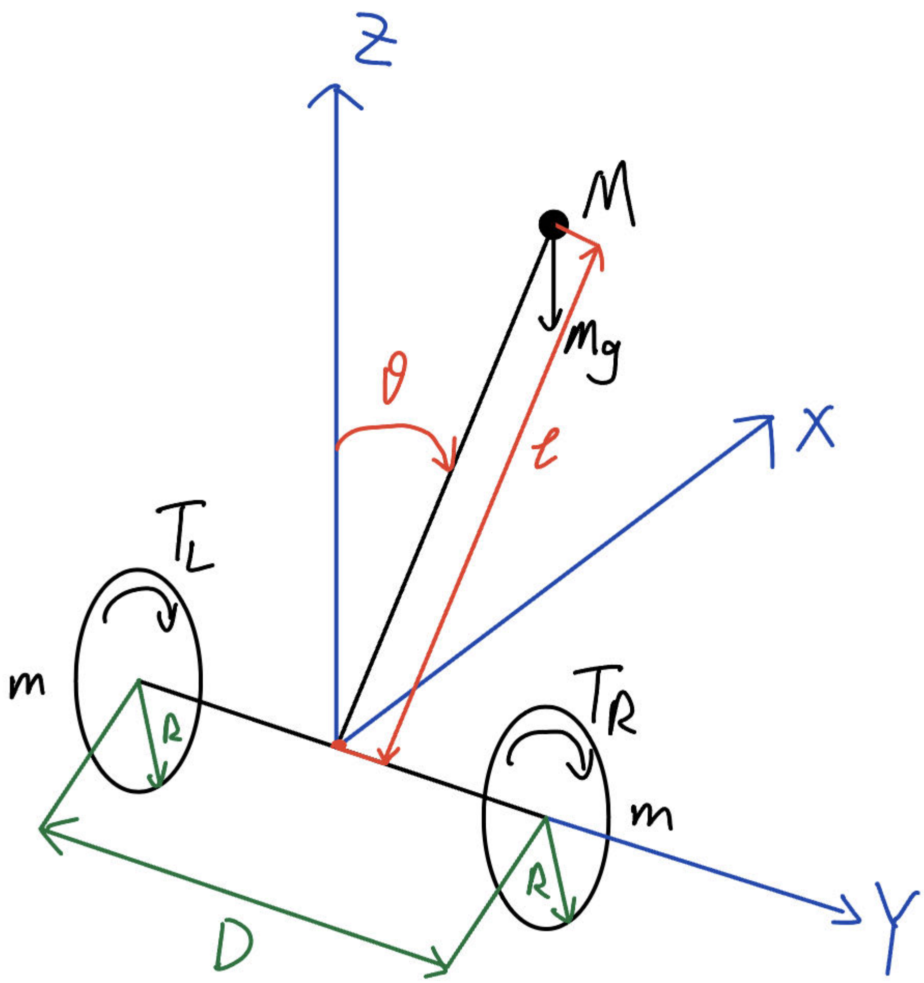

Figure 1: Simplified Wheel Motion Model

2.3.1 Assumption

1. The mass of the body can be represented at the center of the mass.

2. Ignore the mass of the leg linkage and the effect on wheel motion from leg movement.
3. No sliding on the wheels.

2.3.2 Planar Motion: Moving Forward and Backward

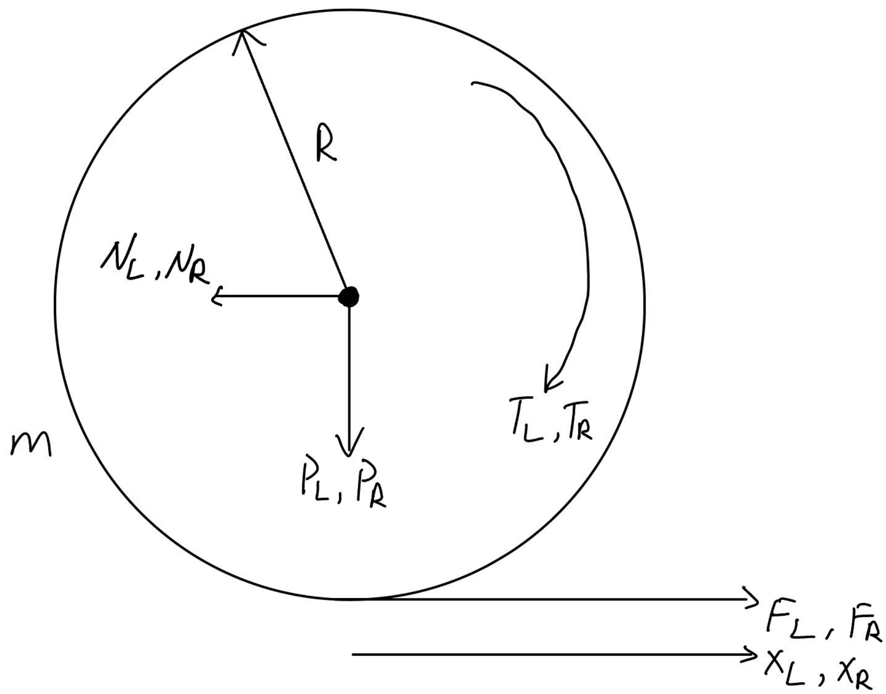

Figure 2: Wheel Model

For the net force:
Because we suppose the rotation for the left and right wheel motors is almost the same, we will use left wheel motor to do the calculation.

$$\begin{gathered} F_{net} = ma \\ F_L - N_L = m\ddot{x}_L \end{gathered} \tag{1}$$

For the net torque:

According to the transformation between rotation and linear motion:

$$\begin{aligned} \tau &= I\alpha \\ T_L - F_L * R &= I_w \frac{\ddot{x_L}}{R} \end{aligned} \tag{2}$$

Combine equation (1) and (2), we can eliminate F_L:
According (1):

$$F_L = m\ddot{x_L} + N_L \tag{3}$$

Plug (3) into (2):

$$\begin{aligned} T_L - (m\ddot{x_L} + N_L) * R &= I_w \frac{\ddot{x_L}}{R} \\ T_L - N_L R - mR\ddot{x_L} &= I_w \frac{\ddot{x_L}}{R} \\ (\frac{I_w}{R} + mR)\ddot{x_L} &= T_L - N_L R \\ \ddot{x_L} &= \frac{T_L - N_L R}{\frac{I_w}{R} + mR} \end{aligned} \tag{4}$$

we can also get the right wheel acceleration:

$$\ddot{x_R} = \frac{T_R - N_R R}{\frac{I_w}{R} + mR} \tag{5}$$

The acceleration of the whole robot is the average acceleration of the left and right wheels.

$$\begin{aligned} \ddot{x} &= \frac{\ddot{x_L} + \ddot{x_R}}{2} \\ \ddot{x} &= \frac{\frac{T_L - N_L R + T_R - N_R R}{\frac{I_w}{R} + mR}}{2} \\ \ddot{x} &= \frac{T_L + T_R - (N_L + N_R)R}{2(\frac{I_w}{R} + mR)} \end{aligned} \tag{6}$$

2.3.3 Planar Motion: Body Balance in Stationary State

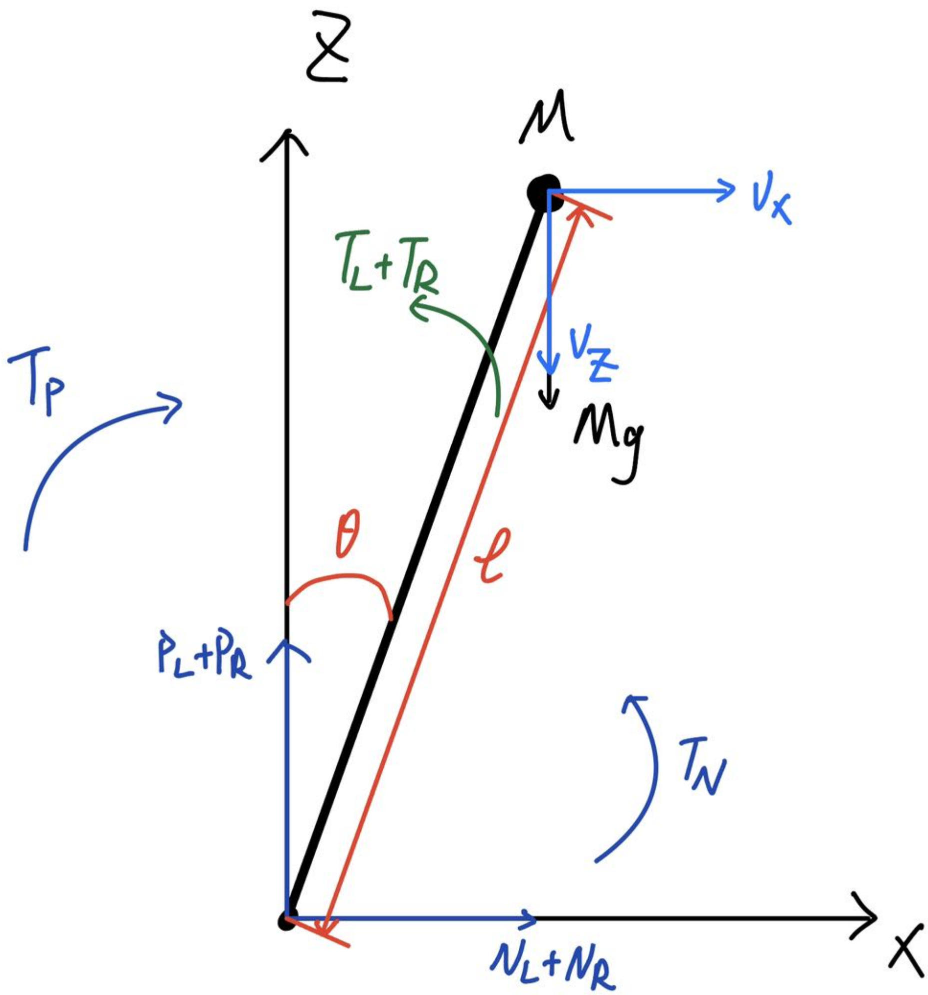

Figure 3: Body Self-Balance Model

We suppose the force can be moved to the body's center of mass by convention.
We can decompose the velocity of the body into the horizontal and vertical directions.

$$v_x = \frac{\partial}{\partial t}(x + lsin(\theta)) = \dot{x} + l * cos(\theta)\dot{\theta} \tag{7}$$

$$v_z = \frac{\partial}{\partial t}(l - l * cos(\theta)) = l * sin(\theta) * \dot{\theta} \tag{8}$$

For the net force in Horizontal:

$$\begin{aligned} F_{net} &= ma \\ N_L + N_R &= M\dot{v_x} \\ N_L + N_R &= M(\ddot{x} + l * cos(\theta)\ddot{\theta} - l * sin(\theta)\dot{\theta}^2) \end{aligned} \tag{9}$$

For the net force in Vertical:

$$\begin{aligned} F_{net} &= ma \\ Mg - (P_L + P_R) &= M\dot{v_z} \\ P_L + P_R &= Mg - M(l * sin(\theta)\ddot{\theta} + l * cos(\theta)\dot{\theta}^2) \end{aligned} \tag{10}$$

By applying $N_L + N_R$ and $P_L + P_R$ to the body, we can get T_N and T_P.

$$T_N = (N_L + N_R) * l * cos(\theta), \quad T_P = (P_L + P_R) * l * sin(\theta) \tag{11}$$

For the net torque along the y-axis:

$$\begin{aligned} \tau &= I\alpha \\ I_y\ddot{\theta} &= T_P - T_N - (T_L + T_R) \end{aligned} \tag{12}$$

Combining the formulas (9), (10), (11), (12), we can eliminate $N_L, N_R, P_L, P_R, T_N, T_P$ to get the complete motion model of the body.

$$\begin{aligned} I_y\ddot{\theta} &= Mg * lsin(\theta) - M\ddot{x} * lcos(\theta) - Ml^2\ddot{\theta} - (T_L + T_R) \\ (I_y + Ml^2)\ddot{\theta} &= Mg * lsin(\theta) - M\ddot{x} * lcos(\theta) - (T_L + T_R) \end{aligned} \tag{13}$$

Combining the formulas (6) and (9), we can eliminate N_L, N_R to get the complete motion model of the wheels.

$$\begin{aligned} (\frac{2I_w}{R^2} + 2m)\ddot{x} &= \frac{T_L + T_R}{R} - M\ddot{x} - M * lcos(\theta)\ddot{\theta} + M * lsin(\theta)\dot{\theta}^2 \\ (\frac{2I_w}{R^2} + 2m + M)\ddot{x} &= \frac{T_L + T_R}{R} - M * lcos(\theta)\ddot{\theta} + M * lsin(\theta)\dot{\theta}^2 \end{aligned} \tag{14}$$

When the robot can keep the balance at a steady state by changing a tiny pitch angle(θ), we can linearize the parameters.

$$cos(\theta) = 1, \quad sin(\theta) = \theta, \quad \dot{\theta}^2 = 0 \tag{15}$$

By plugging (15) into (13) and (14), we can get the system of equations.

$$\begin{cases} (\frac{2I_w}{R^2} + 2m + M)\ddot{x} = \frac{T_L + T_R}{R} - Ml\ddot{\theta} \\ (I_y + Ml^2)\ddot{\theta} = Mgl\theta - M\ddot{x}l - (T_L + T_R) \end{cases} \tag{16}$$

2.3.4 Rotation Motion

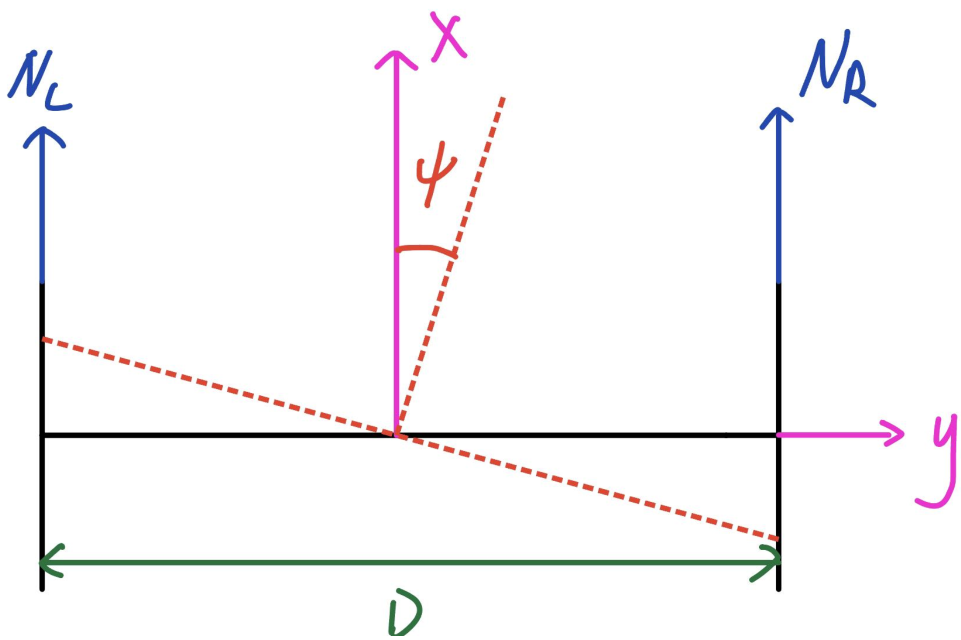

Figure 4: Rotation Model

For the net torque along z-axis:

$$\begin{aligned} \tau &= I\alpha \\ I_z\ddot{\psi} &= \frac{D}{2}(N_L - N_R) \end{aligned} \tag{17}$$

The relationship between the yaw angular acceleration and left and right wheel acceleration is:

$$\ddot{\psi} = \frac{\ddot{x_L} - \ddot{x_R}}{2} \tag{18}$$

Plugging equations (17) and (18) into equations (4) and (5), we can eliminate N_L and N_R :

$$\ddot{\psi} = \frac{T_L - T_R}{R(\frac{2I_z}{D} + \frac{I_w D}{R^2} + mD)} \tag{19}$$

2.4 Classical Mechanical Analysis for Leg Motion

NOTE: If you use the Lagrange equation, you can get the same equations in the below part.

2.4.1 Assumption

1. The mass of the robot is effectively concentrated at the center of mass.
2. The influence of wheeled motion on leg motion is neglected.
3. The mass of the leg links is ignored.
4. The robot maintains a balanced state throughout the leg motion process, i.e., the pitch angle remains zero.
5. The two joint motors on the same side respond synchronously with equal magnitudes of control torque but in opposite directions.

2.4.2 Optimization Calculations for the Five-linkage Structure of the Legs

During the leg motion, the two leg motors of the robot have equal magnitudes of rotation angles but opposite directions, reducing the degrees of freedom of the five-bar linkage mechanism to one. Therefore, we can analyze only half of the five-bar linkage mechanism.

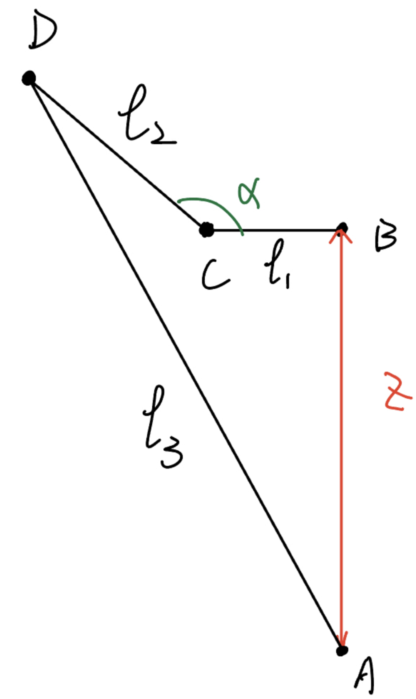

Figure 5: Simplified five-linkage Model

Label	**Meaning**	**Unit**
l_1	Half of the length of the body's side.	m
l_2	The length of the active rod.	m
l_3	The length of the connecting rod.	m
z	The height of the body.	m
z_{min}	The minimum height of the body.	m
z_{max}	The maximum height of the body.	m
Δz	The height scope of the body.	m
α	The angle of the leg motors.	$degree$
α_{min}	The minimum angle of the leg motors.	$degree$
α_{max}	The maximum angle of the leg motors.	$degree$

Table 3: Five-linkage Structure Parameters

According to Figure 5 and the Pythagorean theorem, we can get the geometric relationship.

$$l_3^2 = (z + l_2 sin(\alpha))^2 + (l_1 - l_2 cos(\alpha))^2 \tag{20}$$

Because the length of each linkage is known, we can simply equation (20) to get the relationship between z and α.

$$z = \sqrt{l_3^2 - (l_1 - l_2 cos(\alpha))^2} - l_2 sin(\alpha) \tag{21}$$

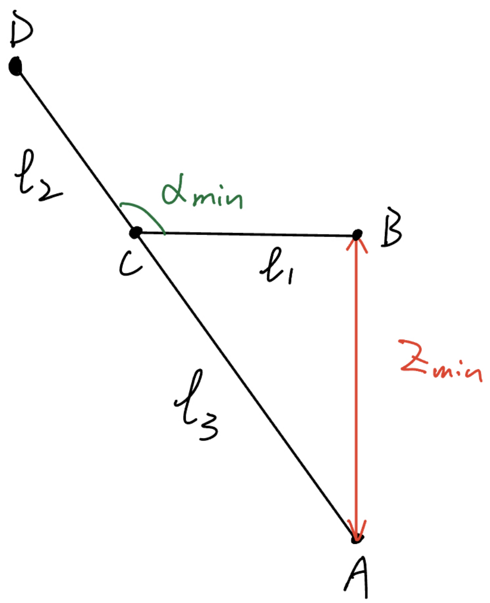

Figure 6: Minimum Height Model

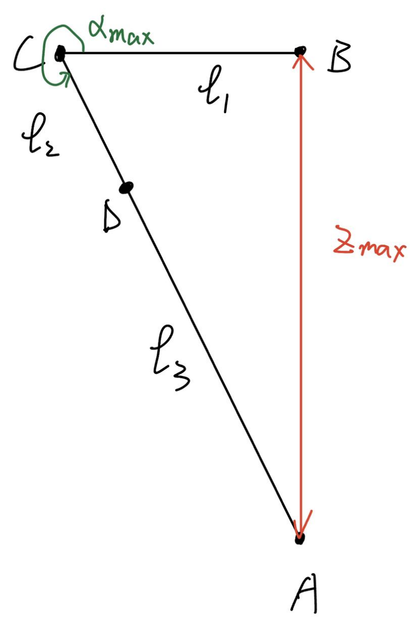

Figure 7: Maximum Height Model

According to Figure 6, we can get the geometric relationship:

$$\begin{cases} l_1 = -(l_3 - l_2)cos(\alpha_{min}) \quad \textit{(horizontal)} \\ z_{min} = (l_3 - l_2)sin(\alpha_{min}) = -tan(\alpha_{min})l_1 \quad \textit{(vertical)} \end{cases} \tag{22}$$

According to Figure 7, we can get the geometric relationship:

$$\begin{cases} l_1 = (l_3 + l_2)\alpha_{max} \quad \textit{(horizontal)} \\ z_{max} = -(l_3 + l_2)sin(\alpha_{max}) = -tan(\alpha_{max})l_1 \quad \textit{(vertical)} \end{cases} \tag{23}$$

According to equations (22) and (23), we use l_1, α_{min} and α_{max} to represent l_2, l_3 and Δz.

$$\begin{cases} l_2 = \frac{l_1}{2}(\frac{1}{\alpha_{max}} + \frac{1}{\alpha_{min}}) \\ l_3 = \frac{l_1}{2}(\frac{1}{\alpha_{max}} - \frac{1}{\alpha_{min}}) \\ \Delta z = z_{max} - z_{min} = l_1(tan(\alpha_{min}) - tan(\alpha_{max})) \end{cases} \tag{24}$$

Therefore, supposing l_1 is a constant, we can get the constrain of the α_{min} and α_{max}.

$$\begin{cases} 90^\circ < \alpha_{min} < 180^\circ \\ 270^\circ < \alpha_{max} < 360^\circ \end{cases} \tag{25}$$

Also, we can suppose the relationship between the angle(α) and the height of the robot(z) is approximately linear with the proportional coefficient $k \geq 0.98$.

2.4.3 Support Phase

When we do the classical mechanical analysis, We can equivalently represent the five-bar linkage of the leg as a rigid bar whose length varies with the joint motor angles, and establish a statics model to support this relationship.

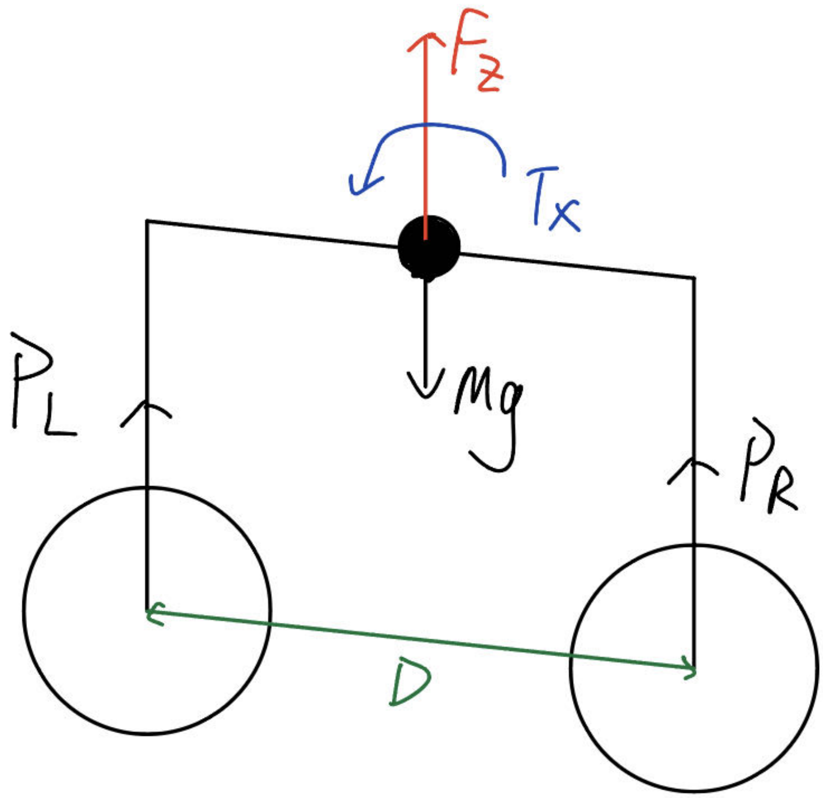

Figure 8: Simplified Support Phase Model

For the net force in the z-axis:

$$F_z = P_L + P_R - Mg \tag{26}$$

F_z is an abstract force that can control the movement along the z-axis to adjust the height of the robot.

$$F_z = M\ddot{z} \tag{27}$$

For the net torque along the x-axis

$$T_x = (P_R - P_L) * \frac{D}{2} \tag{28}$$

T_x is an abstract torque that can be used for controlling the robot's ground self-adaptive motion.

$$T_x = I_x \phi \tag{29}$$

By combining equation (26) and (28), we can get the following equations:

$$\begin{cases} P_L = \dfrac{F_z + Mg}{2} - \dfrac{T_x}{D} \\ P_R = \dfrac{F_z + Mg}{2} + \dfrac{T_x}{D} \end{cases} \tag{30}$$

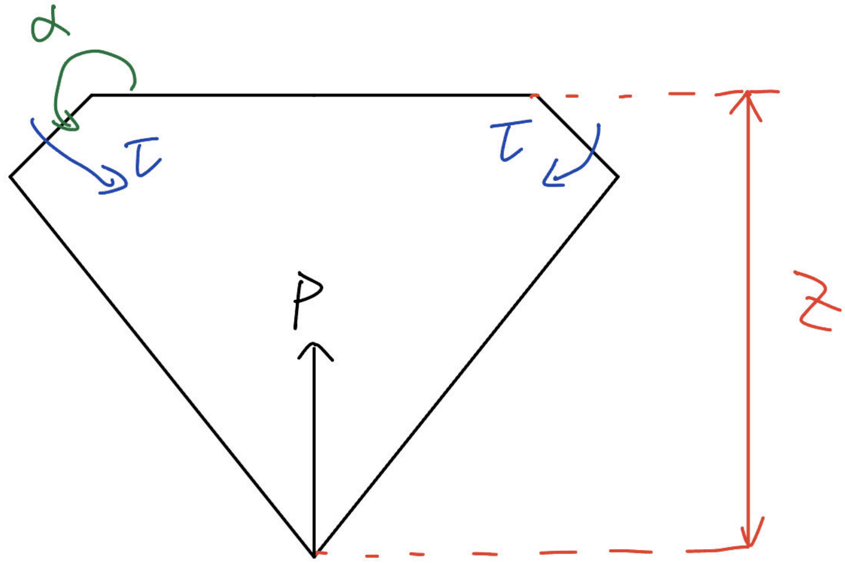

Figure 9: One side five-link model

Take one side leg as an example, we can establish the equations according to the principle of virtual work:

$$2\tau * \delta\alpha = P * \delta z \quad \text{P represents } P_L \text{ or } P_R. \tag{31}$$

By taking the derivative of equation (21), we can get:

$$\delta z = -[\frac{(l_1 - l_2 cos(\alpha)) l_2 sin(\alpha)}{\sqrt{l_3^2 - (l_1 - l_2 cos(\alpha))^2}} + l_2 cos(\alpha)]\delta\alpha \tag{32}$$

Plug in equation(32) to (31), we can get the output torque of the leg motors is:

$$\tau = -[\frac{(l_1 - l_2cos(\alpha))l_2sin(\alpha)}{\sqrt{l_3^2 - (l_1 - l_2cos(\alpha))^2}} + l_2cos(\alpha)] * \frac{P}{2} \tag{33}$$

3 Robot Control Algorithm

3.1 Wheel Motion Control Algorithm (LQR)

3.1.1 LQR Algorithm Analysis

According to equations (16) and (19), we can get the control equations:

$$\begin{cases} a\ddot{x} = T_L + T_R - b\ddot{\theta} \\ c\ddot{\theta} = d\theta - e\ddot{x} - (T_L + T_R) \\ \ddot{\psi} = f(T_L - T_R) \end{cases} \tag{34}$$

where

$$\begin{cases} a = R(\frac{2I_w}{R^2} + 2m + M) \\ b = MlR \\ c = I_y + Ml^2 \\ d = Mgl \\ e = Ml \\ f = \frac{1}{R(\frac{2I_z}{D} + \frac{I_wD}{R^2} + mD)} \end{cases} \tag{35}$$

The state space equation is

$$\begin{cases} \dot{X} = AX + Bu \\ Y = CX \end{cases} \tag{36}$$

We can determine the state vector X = $[x, \dot{x}, \theta, \dot{\theta}, \psi, \dot{\psi}]^T$ and the input vector u = $[T_L, T_R]^T$.

Then, we can convert equation(34) to the following form:

$$\begin{cases} \ddot{x} = \dfrac{-bd}{ac-be}\theta + \dfrac{c+b}{ac-be}(T_L+T_R) \\ \ddot{\theta} = \dfrac{ad}{ac-be}\theta - \dfrac{a+e}{ac-be}(T_L+T_R) \\ \ddot{\psi} = f(T_L-T_R) \end{cases} \tag{37}$$

Converting the above equation in the state space model:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-bd}{ac-be} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{ad}{ac-be} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{c+b}{ac-be} & \frac{c+b}{ac-be} \\ 0 & 0 \\ -\frac{a+e}{ac-be} & -\frac{a+e}{ac-be} \\ 0 & 0 \\ f & -f \end{bmatrix} \cdot \begin{bmatrix} T_L \\ T_R \end{bmatrix} \tag{38}$$

Label	**Value**	**Unit**
m	0.174	kg
M	3.848	kg
I_w	1.4741×10^{-4}	$kg \cdot m^2$
I_y	3.7049207×10^{-2}	$kg \cdot m^2$
I_z	6.563965×10^{-2}	$kg \cdot m^2$
R	0.03	m
l	0.152	m
D	0.3504	m
g	9.81	m/s^2

Table 4: Wheel Motion Parameters(with values)

After plugging in the values, we can get:

$$
\begin{aligned}
A &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-bd}{ac-be} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{ad}{ac-be} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -14.7416 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 114.0117 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
B &= \begin{bmatrix} 0 & 0 \\ \frac{c+b}{ac-be} & \frac{c+b}{ac-be} \\ 0 & 0 \\ -\frac{a+e}{ac-be} & -\frac{a+e}{ac-be} \\ 0 & 0 \\ f & -f \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 21.0112 & 21.0112 \\ 0 & 0 \\ -105.5102 & -105.5102 \\ 0 & 0 \\ 67.6110 & -67.6110 \end{bmatrix}
\end{aligned} \tag{39}
$$

To determine whether this system is controllable, we need to check whether the controllability matrix $C(A, B)$ is full rank.

$$C(A, B) = [B|AB|A^2B|...|A^5B], C(A, B) \in R^{6\times 12} \tag{40}$$

By using Matlab, $rank(C(A, B)) = 6$, this system is controllable.

Then we need to analyze the stability of the system. We can compute the eigenvalues of A to determine whether the current system is stable. If all eigenvalues are in the LHP(left Half Plane), the system is stable.
The eigenvalues of matrix A are:

$$\lambda = [0, 0, 10.6776, -10.6776, 0, 0]^T \tag{41}$$

This system is not stable, we need to design a controller to make this system stable. We will choose the LQR controller.

By using the LQR controller, we need to set the feedback gain matrix K:

$$K = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} \end{bmatrix} \tag{42}$$

Let $u = -KX$, we can rewrite the state space equation into $\hat{X} = AX - BKX = (A - BK)X$. We can let $A' = (A - BK)$ and change the value of each element in K to make the eigenvalues of A' all negative, which will make the whole system stable.

The cost function is:

$$J = \int_0^\infty (X^T Q X + u^T R u) dt \tag{43}$$

We want to change the feedback controller $u = -KX$ to make the cost function become minimal, J_{min}. Matrix Q is a positive semi-definite matrix, which represents the punishment to the state vector X(or the error state vector). If the element in the Q matrix is larger, the corresponding element in the state vector will decrease to 0 more quickly. Matrix R is a positive definite matrix, which represents the punishment to the input vector. It is used to balance the importance of control inputs. Larger weights(elements in the R matrix) are usually assigned to control inputs when you want the system to consume less energy on those inputs or when you want them to be smoother.

The feedback gain matrix K is:

$$K = R^{-1} B^T P \tag{44}$$

Where matrix P is gotten from the algebraic Riccati equation.

In this project, we can use the Matlab LQR function to get the K directly.

According to the test, matrix Q and R used in this project are shown below:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 20000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 90000 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 20000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, R = \begin{bmatrix} 1500 & 0 \\ 0 & 1500 \end{bmatrix} \tag{45}$$

By using Matlab, the feedback gain matrix K is:

$$K = \begin{bmatrix} -0.0183 & -2.5897 & -9.2569 & -1.1094 & 2.5820 & 0.1954 \\ -0.0183 & -2.5897 & -9.2569 & -1.1094 & -2.5820 & -0.1954 \end{bmatrix} \tag{46}$$

3.1.2 Simulink Test

We can establish the model in the Simulink to check whether the controller is useful. The model is shown in **Figure 10**.

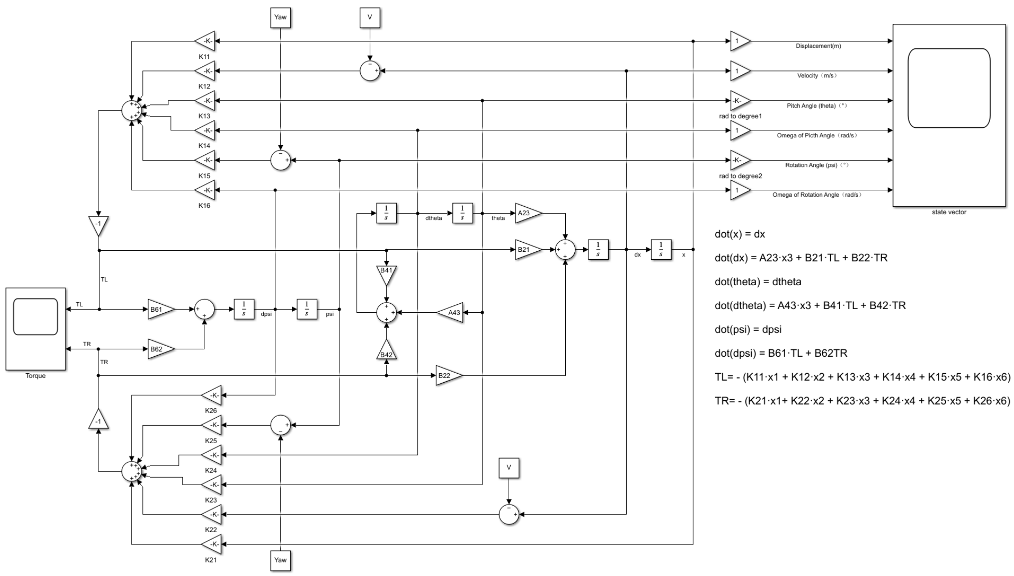

Figure 10: Simulink Wheel Model

Velocity Test

We set the target speed to 0.5m/s, and use the Simulink to simulate the robot behavior. The state vector simulation result is shown in **Figure 11** and the input vector simulation result is shown in **Figure 12**.

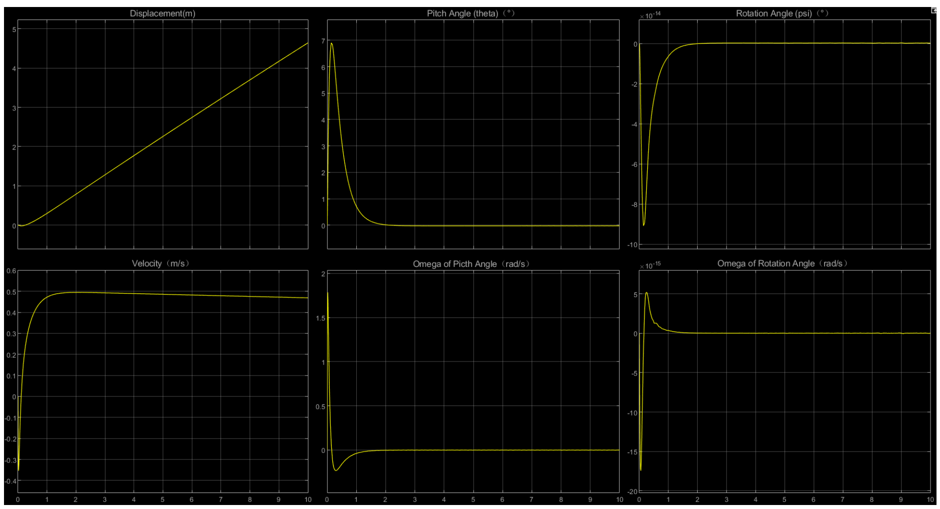

Figure 11: State Vector Simulation Result

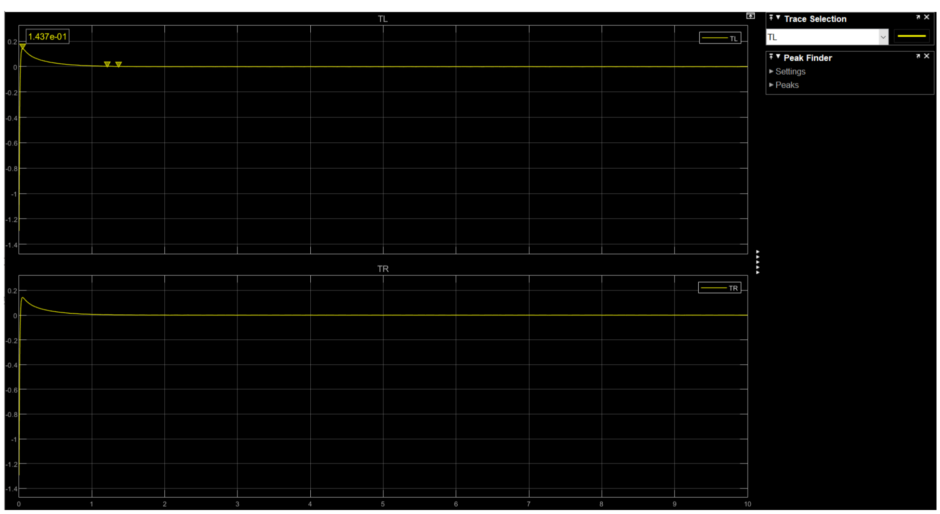

Figure 12: Input Vector Simulation Result

The robot uses about 1s to reach the target speed of 0.5m/s. During this process, the maximum pitch angle is about 7 degrees and then converges to 0. Meanwhile, the rotation angle is pretty small, and the peak torque of each wheel motor is 0.1437 Nm within the peak torque range of the selected motor, which matches our requirements.

Rotation Test

We set the target rotation angle to 30 degrees, and use the Simulink to simulate the robot behavior. The state vector simulation result is shown in **Figure 13** and the input vector simulation result is shown in **Figure 14**.

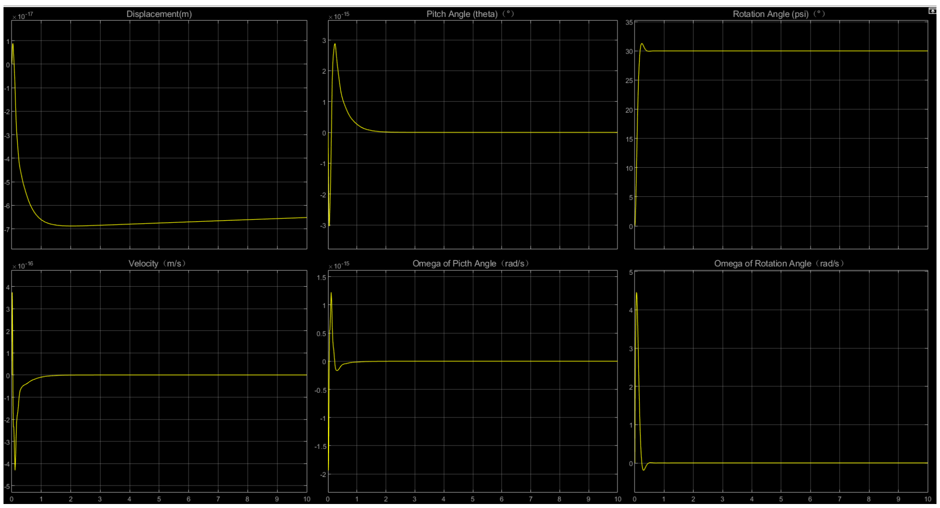

Figure 13: State Vector Simulation Result

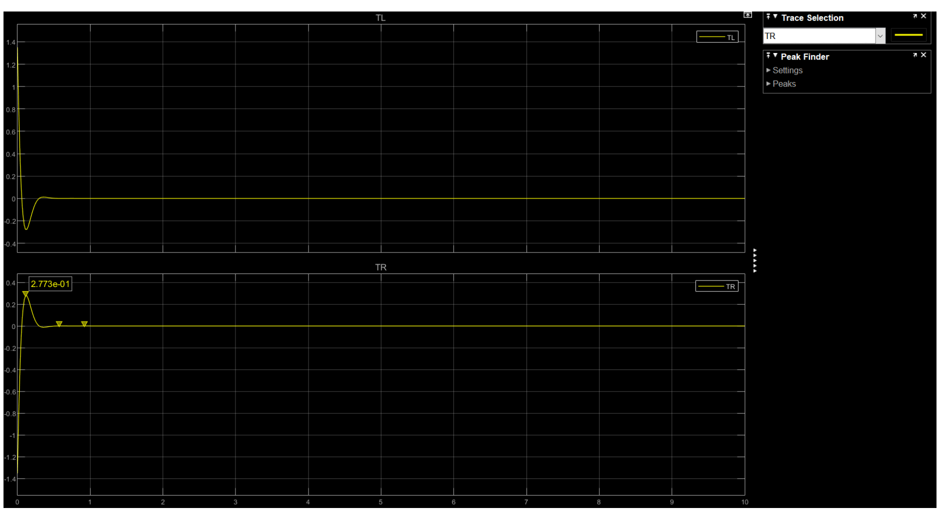

Figure 14: Input Vector Simulation Result

The robot uses less than 0.5s to reach the target rotation angle of 30 degrees. During this process, the maximum pitch angle omega, velocity, and displacement are almost 0. Meanwhile, the peak torque of each wheel motor is 0.2773 Nm within the peak torque range of the selected motor, which matches our requirements.

3.2 Leg Motion Control Algorithm (VMC)

3.2.1 VMC Algorithm Analysis

The leg motion control of the robot is implemented using the Virtual Model Control (VMC) method. This involves adding spring-damper virtual components to establish the virtual forces required for leg motion. Subsequently, the end-effector forces at the leg links are determined using these virtual forces, and finally, the required torque for each joint motor is calculated based on the end-effector forces.

As shown in Figure 8, we have a virtual force along with z-axis, F_z and a virtual torque along with x-axis, T_x. We should establish the spring-damper virtual components, respectively.

For F_z, we can set the spring constant as k_1 and set the damping constant as c_1. Therefore, we can get the equation:

$$F_z = k_1(z_{desired} - z) + c_1(0 - \dot{z}) = M\ddot{z} \tag{47}$$

For T_x, we can set the spring constant as k_2 and set the damping constant as c_2. Therefore, we can get the equation:

$$T_x = k_2\phi + c_2(0 - \dot{\phi}) = I_x\ddot{\phi} \tag{48}$$

We can plug equation (47) and (48) into the equation (30), and we can get:

$$\begin{cases} P_L = \dfrac{k_1(z_{desired} - z) - c_1\dot{z} + Mg}{2} - \dfrac{k_2\phi - c_2\dot{\phi}}{D} \\ P_R = \dfrac{k_1(z_{desired} - z) - c_1\dot{z} + Mg}{2} + \dfrac{k_2\phi - c_2\dot{\phi}}{D} \end{cases} \tag{49}$$

After we have the forces(P_L, P_R) on the end-effectors of the leg, we can get the torque for each leg motor based on the equation (33).

$$\tau = -[\frac{(l_1 - l_2cos(\alpha))l_2sin(\alpha)}{\sqrt{l_3^2 - (l_1 - l_2cos(\alpha))^2}} + l_2cos(\alpha)] * \frac{P}{2} \quad \text{, where P is } P_L \text{ or } P_R \tag{50}$$

In the equation, α is a real-time value, which is gotten from the encoders in the motor. It is hard to introduce this value in the simulation. Therefore, we will use the average value of α during the simulation motion to implement the model.

Label	**Value**	**Unit**
M	3.848	kg
I_x	7.0768703×10^{-2}	$kg \cdot m^2$
l_1	0.05	m
l_2	0.1	m
l_3	0.18	m
D	0.3504	m
g	9.81	m/s^2

Table 5: Leg Motion Parameters(with values)

3.2.2 Simulink Test

We can establish the model in Simulink to check whether the VMC algorithm works. The model is shown in **Figure 15**. Combining motor rating data with simulation test performance, we will choose $k_1 = 400, c_1 = 70, k_2 = 120, c_2 = 10$.

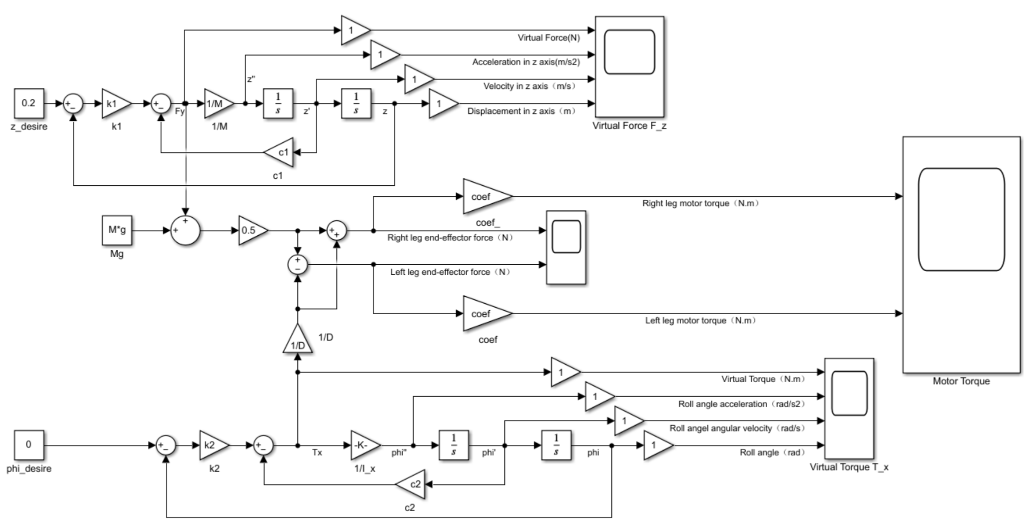

Figure 15: Simulink Leg Model

Elevation Test

We set the target elevation height to 0.2m, and use the Simulink to simulate the robot behavior. The variable simulation result is shown in **Figure 16** and the motor torque simulation result is shown in **Figure 17**.

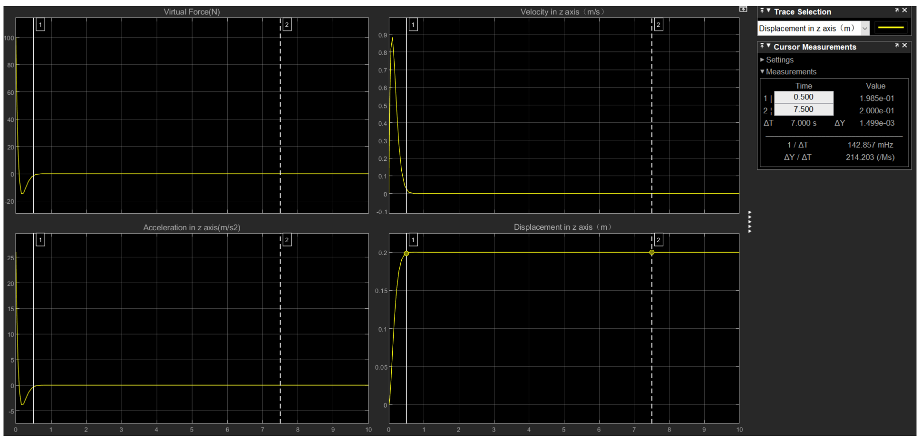

Figure 16: Variable Simulation Result

Figure 17: Motor Torque Simulation

The robot uses about 0.5s to reach the desired height. During this process, each motor's maximum torque is about 3 Nm within the peak torque range of the selected motor, which matches our requirements.

Adaptive Suspension Test

We set the initial roll angle be 0.2 rad and use the Simulink to simulate the robot behavior. The variable simulation result is shown in **Figure 18** and the motor torque simulation result is shown in **Figure 19**.

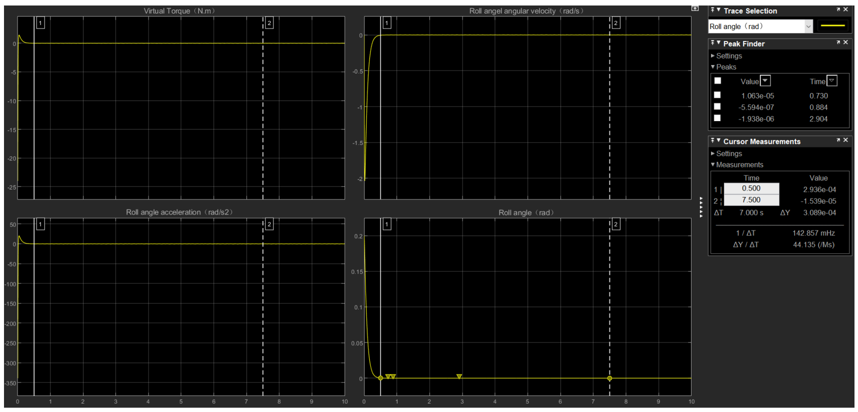

Figure 18: Variable Simulation Result

Figure 19: Motor Torque Simulation

The robot uses about 0.5s to come back to the balance state. During this process, the left motor's maximum torque is about 4.4 Nm, and the right motor's maximum torque is about 2.5 Nm within the peak torque range of the selected motor, which matches our requirements.

Hybrid Test

We combine the first two tests to do the hybrid test and use the Simulink to simulate the robot behavior. The elevation variable simulation result and adaptive suspension variable simulation result are shown in **Figure 20** and **Figure 21**. The motor torque simulation result is shown in **Figure 22**.

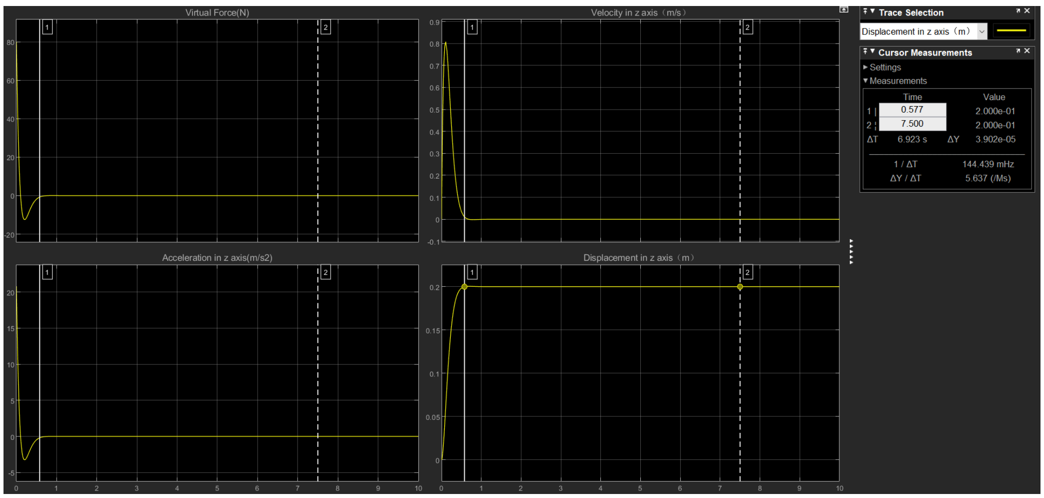

Figure 20: Elevation Variable Simulation Result

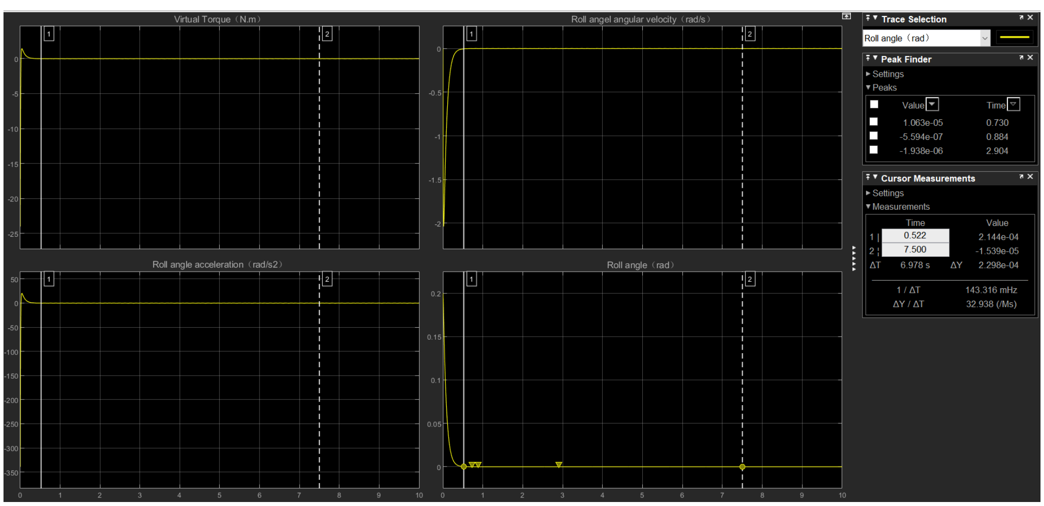

Figure 21: Adaptive Suspension Variable Simulation Result

Figure 22: Motor Torque Simulation

The robot uses about 0.5s to reach the desired height and go back the balance state. During this process, the left motor's maximum torque is about 6.4 Nm, and the right motor's maximum torque is about 2.3 Nm within the peak torque range of the selected motor, which matches our requirements.

4 Conclusion and Future Work

The development log record the entire process of establishing the leg-wheel robot model, researching the control algorithm, and conducting simulations. For physical modeling, I employed the classical mechanics analysis. For the control of wheel motors, I utilized the LQR control method. Regarding the leg motors, I implemented the VMC control method (spring-damping system).

After transplanting the algorithm onto the robot, I observed oscillations in its movements, possibly resulting from excessive compensation by the LQR algorithm. The main causes of this issue are twofold: one lies in the discrepancies between the robot and the physical model, and the other is that the current physical model does not adequately represent all the motion states of the robot. Furthermore, during testing, attempts were made to implement jumping, but the buffering of the leg motors was not ideal. Further modeling and analysis of the jumping state may be necessary.

Therefore, The next steps in the development of this project are as follows: 1. Remodeling the leg linkage motion and incorporating them into the existing model, followed by a reimplementation of the LQR algorithm. 2. Substituting the leg VMC model with both forward and inverse kinematics VMC models and integrating ground detection and modeling analysis for jumping states into the robot.