Comprehensive Framework for Dynamic Quadruped Locomotion

Ibraheem Al-Tameemi, Ethan Chandler, Hushmand Esmaeili, Varun Walimbe

***Abstract*— This study presents a system for controlling the smooth, dynamic motion of quadruped robots. The research presents two control objectives (adaptive inverse dynamics and model predictive controller) for the smooth and stable locomotion of the proposed quadruped robot. The footstep planner and swing leg trajectory are generated during the swing phase to obtain the reference foot position, velocity, and acceleration. The inverse kinematics (IK) solution is then calculated to determine the appropriate joint configuration, velocity, and acceleration. During the swing phase, the robot's dynamics and adaptive control are also implemented to achieve the desired leg trajectory. During the stance phase, the model predictive control is designed to calculate the desired ground reaction forces, which is then fed to the stance dynamics in order to obtain the proper joint torque. In addition, a state estimator is employed to estimate the state of the system.**

I. Introduction

In recent years, legged robotics has been an active research field. Due to their distinctive motion shape, control mechanism, and adaptation to complicated situations, bio-inspired foot robots have become a topic of interest within the field of robot research. The stability and load capacity of quadruped robots are significantly superior to those of biped robots. The mechanism and control strategy complexity is less than that of hexapod and multi-legged robots. Thus, quadruped robots have numerous applications in military transportation, forest identification, and emergency rescue.

Achieving dynamic locomotion with legged robots in such real-life environments is challenging. The problem can be extended to locomotion in rough terrain, which introduce greater external disturbances, and even recovery from pushing or slippage. Different speeds or terrains may then require different gaits and even compliant legs.

Current solutions for highly dynamic locomotion have yielded impressive results. The Mini Cheetah quadruped developed by MIT proposed a novel control framework [1], which combines a newly devised whole-body impulse control (WBIC) and model predictive control (MPC). Although previous WBIC formulations focus on tracking center of mass (CoM) trajectories, WBIC incorporates both body posture stabilization and reaction forces formulation. The MPC then finds the reaction forces required by the WBIC. The control framework was tested on six different gaits in different environments. [5] propose an implementation of MPC to determine ground reactions for a quadruped robot. The main contribution is a controller that can stabilize a large number of gaits. Experimental results demonstrated control of gaits including stand, trot, pronk, bound, and others.

Despite major advancements in control frameworks for achieving dynamic locomotion, there is still a lack of research on the generation of a continuous time gait. Thus, defining a constant duty factor might cause a jerky gait transition. This paper proposes a control framework that relies on a continuous time function for the gait generation to achieve a smooth dynamic locomotion of the quadruped robot.

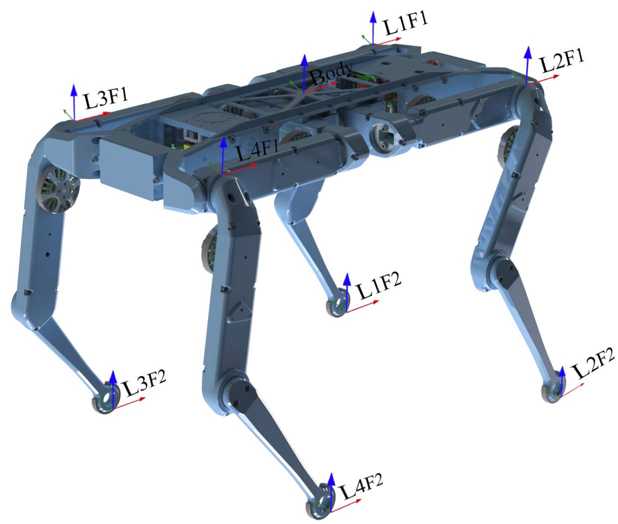

Fig. 1. **Solo-12 Frames**

This paper is organized as follows. Section II describes the basic system architecture and conventions used in this paper for the Solo-12. The swing phase is presented in Section III, followed by the swing phase control problem formulation in Section IV, while Section V contains the model predictive control. Section VI contains state estimation.VII section is the results, while VIII is the conclusions and future work.

IX section represents conclusion and future work.

II. Architecture

A. Solo-12

The Solo-12 quadruped is an open-hardware 12-DOF robot developed by Open Dynamic Robot Initiative [3]. The robot features 12 Antigravity 4004 300kV BLDC motors, which are driven by 6 MicroDriver v2's, with a Launchpad F28069M to provide FOC. The robot is high-performance and low-cost, making it desirable for research purposes.

B. Conventions

The body and local leg coordinate systems are defined in Figure 1, where the red, green, and blue basis vectors correspond to the x-axis, y-axis, and z-axis of each frame respectively. Additionally, all vectors in this paper are bold, upright, and lowercase ($\boldsymbol{a}$, $\boldsymbol{\omega}$), matrices are bold, upright, and uppercase ($\boldsymbol{A}$, $\boldsymbol{\Omega}$), and scalars are lowercase and italicized (a, ω). Finally, $\mathbf{1}_n$ is used to denote an $n \times n$ identity matrix.

III. Swing Phase

In swing phase, each leg can be modeled as an independent 3-DOF serial linkage. In this section,the kinematic solution, swing phase dynamics,foot planner, and swing leg trajectory are briefly covered.

A. Leg Forward Kinematics

The forward kinematics of each leg are given by the following equations:

$$\mathbf{x}^{\text{hip}}_{\text{foot tip}} = -\cos(\theta_1)(L_4\cos(\theta_2+\theta_3) - L_2 + L_3\cos(\theta_2)) \quad (1)$$

$$\mathbf{y}^{\text{hip}}_{\text{foot tip}} = -\sin(\theta_1)(L_4\cos(\theta_2+\theta_3) - L_2 + L_3\cos(\theta_2)) \quad (2)$$

$$\mathbf{z}^{\text{hip}}_{\text{foot tip}} = L_1 - L_4\sin(\theta_2+\theta_3) - L_3\sin(\theta_2) \quad (3)$$

B. Leg Inverse Kinematics

Assuming $q_3 > 0$, $-\sqrt{y^2+z^2-l_1^2} < 0$, the inverse kinematics of the legs is given by:

$$\alpha = \cos^{-1}\frac{|z|}{\sqrt{y^2+z^2}}, \quad \beta = \cos^{-1}\frac{l_1}{\sqrt{y^2+z^2}} \quad (4)$$

$$q_1 = \begin{cases} \alpha - \beta, & z > 0 \\ \pi - \alpha - \beta, & z < 0 \end{cases} \quad (5)$$

$$x' = x, \quad y' = -\sqrt{y^2+z^2-l_1^2}, \quad D = x'^2 + y'^2 \quad (6)$$

$$\phi = \cos^{-1}\frac{|x'|}{\sqrt{D}}, \quad \psi = \cos^{-1}\frac{l_2^2 + D - l_3^2}{2l_2\sqrt{D}} \quad (7)$$

$$q_2 = \begin{cases} \frac{\pi}{2} - \psi - \phi, & x' > 0 \\ -\frac{\pi}{2} - \psi - \phi, & x' < 0 \end{cases} \quad (8)$$

$$q_3 = \cos^{-1}\frac{l_2^2 + l_3^2 - D}{2l_2 l_3} \quad (9)$$

C. Inverse Kinematics Scheme

The inverse kinematics scheme is defined by three tasks. The first one is to keep the base at constant height and follow a reference horizontal velocity. The second is the keep the base orientation horizontal (no pitch or roll) and follow a reference yaw angular velocity. The third task is to follow the reference trajectory of the swing feet while maintaining the feet in stance phases fixed [2][6].

The functions for all three tasks are given by:

$$\dot{\mathbf{x}}_{\text{pos}} = \mathbf{R}^0_{\text{b}}\dot{\mathbf{q}}_{\text{pos}} \quad (10)$$

$$\dot{\mathbf{x}}_{\text{ang}} = \mathbf{R}^0_{\text{b}}\dot{\mathbf{q}}_{\text{ang}} \quad (11)$$

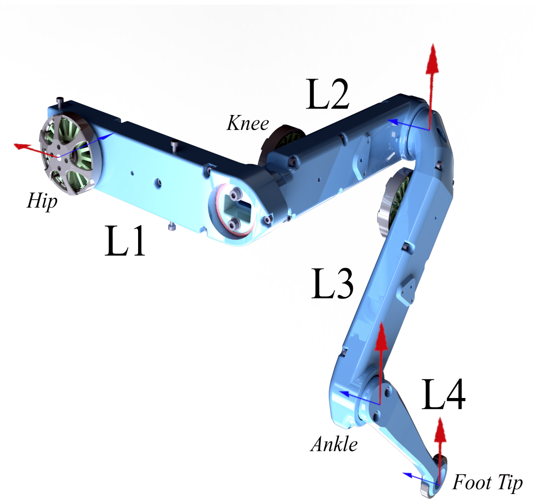

Fig. 2. **Leg Parameters**

$$\dot{\mathbf{x}}_1 = \mathbf{R}^0_{\text{b}}\dot{\mathbf{q}}_{\text{pos}} + \mathbf{T}^{\text{b}}_1 \times \mathbf{R}^0_{\text{b}}\dot{\mathbf{q}}_{\text{ang}} + \mathbf{J}_1\dot{\mathbf{q}}_1 \quad (12)$$

$$\dot{\mathbf{x}}_2 = \mathbf{R}^0_{\text{b}}\dot{\mathbf{q}}_{\text{pos}} + \mathbf{T}^{\text{b}}_2 \times \mathbf{R}^0_{\text{b}}\dot{\mathbf{q}}_{\text{ang}} + \mathbf{J}_2\dot{\mathbf{q}}_2 \quad (13)$$

$$\dot{\mathbf{x}}_3 = \mathbf{R}^0_{\text{b}}\dot{\mathbf{q}}_{\text{pos}} + \mathbf{T}^{\text{b}}_3 \times \mathbf{R}^0_{\text{b}}\dot{\mathbf{q}}_{\text{ang}} + \mathbf{J}_3\dot{\mathbf{q}}_3 \quad (14)$$

$$\dot{\mathbf{x}}_4 = \mathbf{R}^0_{\text{b}}\dot{\mathbf{q}}_{\text{pos}} + \mathbf{T}^{\text{b}}_4 \times \mathbf{R}^0_{\text{b}}\dot{\mathbf{q}}_{\text{ang}} + \mathbf{J}_4\dot{\mathbf{q}}_4 \quad (15)$$

where $\dot{\mathbf{q}}_{\text{pos}}$ and $\dot{\mathbf{q}}_{\text{ang}}$ are the base linear and angular velocities in the world frame, $\dot{\mathbf{x}}_i$ the velocity of the tip of the i_{th} foot in world frame, $\dot{\mathbf{q}}_{\text{pos}}$ and $\dot{\mathbf{q}}_{\text{ang}}$ the base linear and angular velocities in the base frame, $\dot{\mathbf{q}}_i$ the joint velocities of the i_{th} leg, $\mathbf{R}^0_{\text{b}}$ the rotation matrix from base to world frame, $\mathbf{T}^{\text{b}}_i$ the position of the i_{th} foot in the base frame, and $\mathbf{J}_i$ the Jacobian of the i_{th} foot.

The reference joint velocity configurations can be computed by rearranging the above equations for joint configuration velocity. The reference joint position is computed through a numerical method. The equations for the above computations are given by:

$$\begin{aligned}\mathbf{q}^{\text{cmd}}_i =& \mathbf{q}_i + \mathbf{J}^{-1}_i[(\mathbf{x}^{\text{des}}_i - \mathbf{x}_i) - (\mathbf{x}^{\text{des}}_{\text{pos}} - \mathbf{x}_{\text{pos}}) \\ & - (\mathbf{T}^{\text{b}}_i \times \mathbf{R}^0_{\text{b}})\mathbf{R}^0_{\text{b}}(\mathbf{x}^{\text{des}}_{\text{ang}} - \mathbf{x}_{\text{ang}})]\end{aligned} \quad (16)$$

$$\dot{\mathbf{q}}^{\text{cmd}}_i = \mathbf{J}^{-1}_i[\dot{\mathbf{x}}^{\text{des}}_i - \dot{\mathbf{x}}^{\text{des}}_{\text{pos}} - (\mathbf{T}^{\text{b}}_i \times \mathbf{R}^0_{\text{b}})\mathbf{R}^0_{\text{b}}\dot{\mathbf{x}}^{\text{des}}_{\text{ang}}] \quad (17)$$

$$\ddot{\mathbf{q}}_{IK} = \ddot{\mathbf{q}}^{\text{cmd}} = \mathbf{J}^{-1}(\ddot{\mathbf{x}}^{\text{cmd}} - \dot{\mathbf{J}}\dot{\mathbf{q}}) \quad (18)$$

where $\mathbf{q}^{\text{cmd}}_i$ and $\dot{\mathbf{q}}^{\text{cmd}}_i$ are the reference (command) joint positions and velocities for the i_{th} leg, $\mathbf{q}_i$ the current joint positions for the i_{th} leg obtained from the state estimator, $\mathbf{x}^{\text{des}}_i$ and $\mathbf{x}_i$ the desired and current joint positions for the i_{th} leg in the world frame, $\mathbf{x}^{\text{des}}_{\text{pos}}$ and $\mathbf{x}_i$ the desired and current base linear positions in the world frame, $\mathbf{x}^{\text{des}}_{\text{ang}}$ and $\mathbf{x}_{\text{ang}}$ the desired and current base angular positions in the world frame, $\dot{\mathbf{x}}_i^{\text{des}}$ the desired joint velocities for the i_{th} leg in the world frame, $\dot{\mathbf{x}}_{\text{pos}}^{\text{des}}$ and $\dot{\mathbf{x}}_{\text{ang}}^{\text{des}}$ the desired base linear and angular velocities in the world frame.

Note that the desired base linear and angular velocities are determined by the user (or by a global planner) and the desired foot positions and velocities are determined by the swing trajectory generator, which we expand on in the *Foot Step Planner* section. The position of the i_{th} foot in the base frame, $\mathbf{T}_i^{\text{b}}$, is given by the forward kinematics of the feet, which is expanded on in the *Leg Forward Kinematics* subsection. The rotation matrix from base to world frame, $\mathbf{R}_{\text{b}}^0$, as well as the current base and joints positions and velocities, are estimated using the State Estimator.

D. Inverse Dynamics

The swing dynamics of the leg is derived using the Lagrange method and verified using Recursive-Newton Euler is as follows:

$$\begin{aligned} M_{11} =& I_1 + I_2 + I_3 + \\ & m_3(r_3\cos(q_2+q3) - l_2 + l_3\cos(q_2)^2 + \\ & m_2(l_2 - r_2\cos(q_2))^2 \end{aligned}$$

$$M_{12} = 0$$

$$M_{13} = 0$$

$$M_{21} = 0$$

$$\begin{aligned} M_{22} =& I_2 + I_3 + m_2 r_2^2 + \\ & m_3(l_3^2 + 2\cos(q_3 l3 r_3 + r_3^2) \end{aligned}$$

$$M_{23} = I_3 + m_3 r_3(r_3 + l_3\cos(q_3)$$

$$M_{31} = 0$$

$$M_{32} = I_3 + m_3 r_3(r_3 + l_3\cos(q_3)$$

$$M_{33} = m_3 r_3^2 + I_3$$

$$\mathbf{M} = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \tag{19}$$

$$\begin{aligned} C_{11} =& -\dot{q_2}(m_3(r_3\sin(q_2+q_3) + \\ & l_3\sin(q_2))(r_3\cos(q_2+q_3) - \\ & l_2 + l_3\cos(q_2)) - m_2 r_2\sin(q_2) \\ & (l_2 - r_2\cos(q_2))) - \\ & \dot{q_3}(m_3(r_3\sin(q_2+q_3) + \\ & l_3\sin(q_2))(r_3\cos(q_2+q_3) - \\ & l_2 + l_3\cos(q_2)) - m_2 r_2\sin(q_2)(l_2 - \\ & r_2\cos(q_2))) \end{aligned}$$

$$\begin{aligned} C_{12} =& -\dot{q_1}(m_3(r_3\sin(q_2+q_3) + \\ & l_3\sin(q2))(r_3\cos(q_2+q_3) - \\ & l_2 + l_3\cos(q_2)) - \\ & m_2 r_2\sin(q_2)(l_2 - r_2\cos(q_2))) \end{aligned}$$

$$\begin{aligned} C_{13} =& -\dot{q_1}(m_3(r_3\sin(q_2+q_3) + \\ & l_3\sin(q_2))(r_3\cos(q_2+q_3) - \\ & l_2 + l_3\cos(q_2)) - \\ & m_2 r_2\sin(q_2)(l_2 - r_2\cos(q_2))) \end{aligned}$$

$$\begin{aligned} C_{21} =& \dot{q_1}(m_3(r_3\sin(q_2+q_3) + \\ & l_3\sin(q_2))(r_3\cos(q_2+q_3) - \\ & l_2 + l_3\cos(q_2)) - \\ & m_2 r_2\sin(q_2)(l_2 - r_2\cos(q_2))) \end{aligned}$$

$$C_{22} = 0$$

$$C_{23} = 0$$

$$\begin{aligned} C_{31} =& \dot{q_1}(m_3(r_3\sin(q_2+q_3) + \\ & l_3\sin(q_2))(r_3\cos(q_2+q_3) - \\ & l_2 + l_3\cos(q_2)) - \\ & m_2 r_2\sin(q_2)(l_2 - r_2\cos(q_2))) \end{aligned}$$

$$C_{32} = 0$$

$$C_{33} = 0$$

$$\mathbf{C} = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \tag{20}$$

$$G_1 = 0$$

$$\begin{aligned} G_2 =& g m_3(r_3\sin(q_2+q_3) + l_2\sin(q_2)) + \\ & g m_2 r_2\sin(q_2) \end{aligned}$$

$$G_3 = g m_3 r_3\sin(q_2+q_3)$$

$$\mathbf{G} = \begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix} \tag{21}$$

$$\boldsymbol{\tau} - \mathbf{J}(\mathbf{q})^T\mathbf{F}_{\text{tip}} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) \tag{22}$$

IV. Swing Phase Control

This controller's purpose is to ensure that the leg follows the desired trajectory during the swing. Adaption law is used to estimate the uncertain value of the leg links' moments of inertia and to input these values to the controller that would generate the reference torque.

A. Gait Generator

Our architecture makes use of a symmetric forward wave gait generator which outputs the phase offsets Ψ of each leg i for a given gait template, duty factor β and crab angle α. The generator works by linearly interpolating predefined phase offsets at known duty factors to find a computationally inexpensive approximation of the true continuous kinematic phase function.

$$\mathbf{\Psi}_i(\alpha,\beta) = \Bigg[4\left(\left(\beta-\frac{1}{2}\right)\mathbf{\Psi}\left(\alpha,\frac{3}{4}\right) - \left(\beta-\frac{3}{4}\right)\Psi\left(\alpha,\frac{1}{2}\right)\right)\Bigg]_{\mathrm{mod}\,i} \quad i\in[1,4] \tag{23}$$

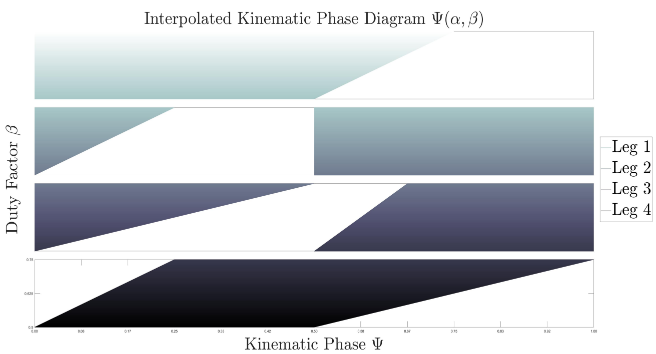

Fig. 3. **Interpolated Phase Offsets:** A trot gait with $\alpha = 0^\circ$ for $0.5 \le \beta \le 0.75$

The resulting gait function is smooth [13], and thus, we do not have to deal with the jerky gait transitions that hard coded gaits suffer from. With the ability to alter the actual duty factor and crab angle of the legs in real time, our approach offers the possibility of both higher stability and higher stable velocities because the gait generator can optimize the duty factor to reach a desired stability margin, without relying on changing the constraints of the leg trajectory generators.

B. Foot Step Planner

The goal of the footstep planner is to identify the trajectory of each foot while in swing phase. The target location at each cycle is given by:

$$\mathbf{r}_i^{\text{cmd}} = \mathbf{p}_{\text{shoulder},i} + \mathbf{p}_{\text{symmetry}} + \mathbf{p}_{\text{centrifugal}} \tag{24}$$

where,

$$\mathbf{p}_{\text{shoulder},i} = \mathbf{p}_k + \boldsymbol{R}_z\left(\psi_k\right)\mathbf{l}_i, \tag{25}$$

$$\mathbf{p}_{\text{symmetry}} = \frac{t_{\text{stance}}}{2}\mathbf{v} + k\left(\mathbf{v}-\mathbf{v}^{\text{cmd}}\right), \tag{26}$$

$$\mathbf{p}_{\text{centrifugal}} = \frac{1}{2}\sqrt{\frac{h}{g}}\mathbf{v}\times\boldsymbol{\omega}^{\text{cmd}} \tag{27}$$

Note that in the simplest case (without centripetal velocities, and while the robot is perfectly tracking the reference velocity), $\mathbf{r}_i^{\text{cmd}}$ is simply the summation of the position of the body w.r.t world, the projection of the position of the local hip frame onto the XZ world plane w.r.t the body, and half of the body displacement w.r.t world between the k and $k+1$ time-step.

To satisfy this trajectory at the potentially variable frequency of the gait scheduler, we have imposed ...

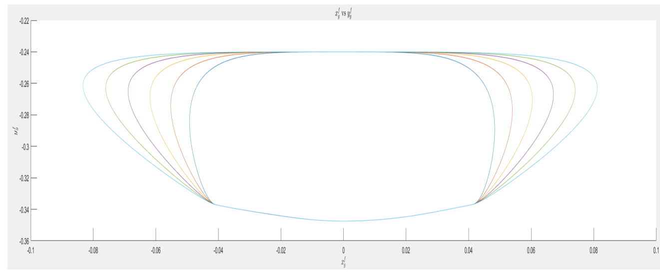

Fig. 4. **Footstep Planner:** Leg Trajectory

C. Adaptive Control

We formulate an adaptive controller which runs within the "swing" block of the "main locomotion controller". The two sub-blocks, "swing" and "stance", together control the overall foot trajectory of the quadruped. Both of these block take in input from the gait scheduler.

The controller scheme is given by:

$$\boldsymbol{\tau}_{\text{adaptive}} = \hat{\mathbf{M}}(\mathbf{q})\mathbf{U} + \hat{\mathbf{C}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} - \hat{\mathbf{g}}(\mathbf{q}) \tag{28}$$

$$\mathbf{U} = -K_p\mathbf{e} - K_d\dot{\mathbf{e}} + \ddot{\mathbf{q}}_{\text{des}} \tag{29}$$

$$\mathbf{e} = \mathbf{q} - \mathbf{q}_{\text{des}} \tag{30}$$

$$\dot{\mathbf{e}} = \dot{\mathbf{q}} - \dot{\mathbf{q}}_{\text{des}} \tag{31}$$

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q}) = \hat{\mathbf{M}}(\mathbf{q})\mathbf{U} + \hat{\mathbf{C}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} - \hat{\mathbf{g}}(\mathbf{q}) \tag{32}$$

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q}) = \hat{\mathbf{M}}(\mathbf{q}) - K_p\mathbf{e} - K_d\dot{\mathbf{e}} + \ddot{\mathbf{q}}_{\text{des}} + \hat{\mathbf{C}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} - \hat{\mathbf{g}}(\mathbf{q}) \tag{33}$$

$$\hat{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{e}} + \hat{\mathbf{M}}(\mathbf{q})[K_p\mathbf{e} + K_d\dot{\mathbf{e}}] = (\hat{\mathbf{M}} - \mathbf{M})\ddot{\mathbf{q}} + (\hat{\mathbf{C}} - \mathbf{C})\dot{\mathbf{q}} + (\hat{\mathbf{g}} - \mathbf{g}) \tag{34}$$

$$\ddot{\mathbf{e}} = -K_p\mathbf{e} - K_d\dot{\mathbf{e}} + \hat{\mathbf{M}}^{-1}(\mathbf{Y}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}))\tilde{\alpha} \tag{35}$$

$$\tilde{\alpha} = \hat{\alpha} - \alpha \tag{36}$$

where $\tilde{\alpha}$ is the estimation error while α contains the actual values of I_1, I_2, I_3.Furthermore let,

$$\mathbf{x} = \begin{bmatrix} \mathbf{e} \\ \dot{\mathbf{e}} \end{bmatrix}$$

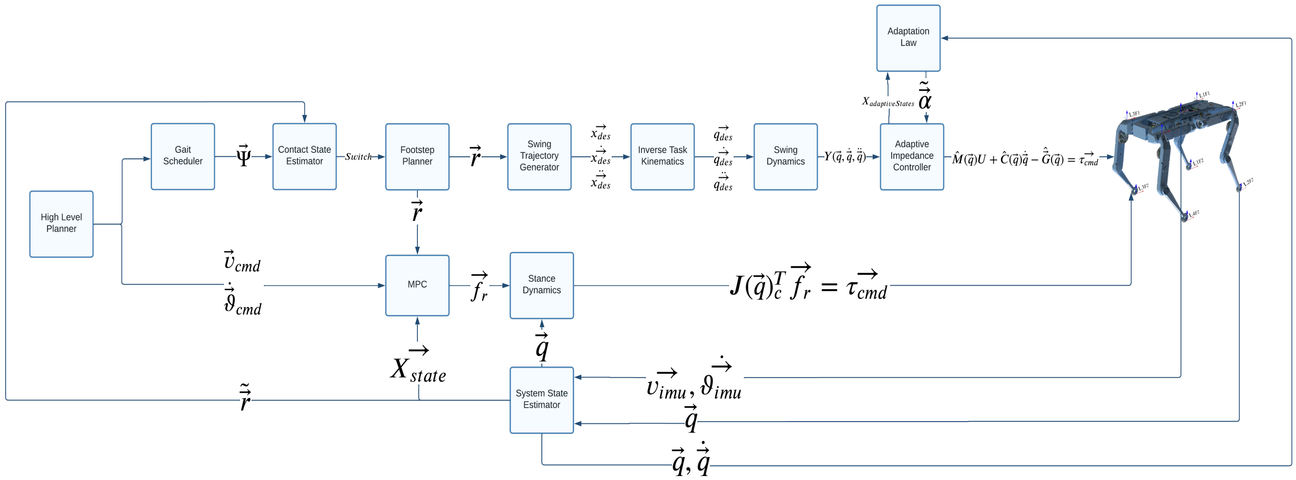

Fig. 5. **Overall Control Architecture**. Full control scheme

Then,

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{e}} \\ \ddot{\mathbf{e}} \end{bmatrix}$$

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & \mathbf{I} \\ -K_p & -K_d \end{bmatrix} \begin{bmatrix} \mathbf{e} \\ \dot{\mathbf{e}} \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix} \Phi \tilde{\alpha}$$

Using Lyapunov method, we design the dynamics of the adaptive law we proposed:

$$\dot{\tilde{\alpha}} = -\Gamma^{-1}(\Phi^T \mathbf{B}^T \mathbf{P} \mathbf{X}) \tag{37}$$

$$\Phi = \hat{\mathbf{M}}(\mathbf{q})^{-1} Y(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \tag{38}$$

in the equation (37) Γ is the 5x5 tuning matrix, Φ equals to product of matrices as seen in (38), **P** is calculated using the Lyapunov function. **B**, **P**, **X** in equation (38) are as follows:

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{P} = Lyapunov(\mathbf{A}_{closed}^T, \mathbf{Q}) \tag{39}$$

where,

$$\mathbf{A}_{closed} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -5 & 0 & 0 & -\omega & 0 & 0 \\ 0 & -5 & 0 & 0 & -\omega & 0 \\ 0 & 0 & -5 & 0 & 0 & -\omega \end{bmatrix}$$

$$\mathbf{X} = States_{Actual} - States_{Desired} \tag{40}$$

Hence,

$$\mathbf{X} = \begin{bmatrix} \theta_1 - \theta_{1d} \\ \theta_2 - \theta_{2d} \\ \theta_3 - \theta_{3d} \\ \dot{\theta_1} - \dot{\theta}_{1d} \\ \dot{\theta_2} - \dot{\theta}_{2d} \\ \dot{\theta_3} - \dot{\theta}_{3d} \end{bmatrix}$$

V. Model Predictive Control

The objective of our MPC is to find reaction forces which would allow our robot to follow a given trajectory. By applying several simplifications to the inverse dynamics (i.e., approximating the multi-body system as a rigid body with instantaneous states to keep the system time-varying linear), the approach keeps the formulation convex, and hence, the problem is fast to solve and furthermore, can be solved by a unique global minimum.

Our MPC Scheme is given by:

$$\begin{aligned} \min_{\mathbf{x},\mathbf{f}} \quad & \sum_{k=0}^{m} ||\mathbf{x}(k+1) - \mathbf{x}^{ref}(k+1)||_{\mathbf{Q}} + ||\mathbf{f}(k)||_{\mathbf{R}} \\ \text{s.t.} \quad & |f_x| \le \mu f_z \\ & |f_y| \le \mu f_z \\ & f_z > 0 \end{aligned} \tag{41}$$

where $\mathbf{x}(k+1)$ is the state at time $k+1$ and $\mathbf{f}(k)$ is the linearized lumped mass dynamics at time k. The constraints represent an approximation of friction cones at the feet of the robot to prevent slippage.

VI. State Estimation

After getting Joint velocity and joint position from encoders we use this as input to state estimator. We use Kalman filter for the state estimation of joints as well as state estimation of COM/body of the robot. The workflow of getting the accurate state is straightforward and s given below:

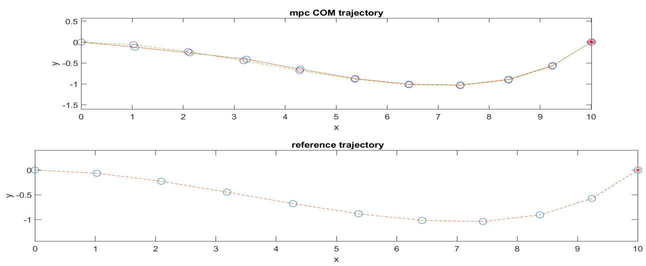

Fig. 6. **MPC:** MPC Using the Dynamical Model to Follow a Trajectory

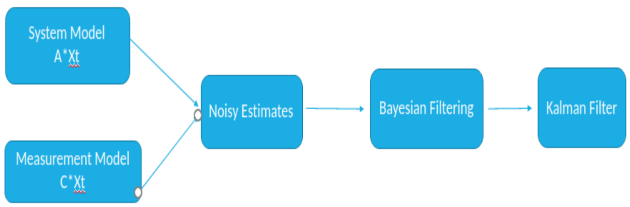

Fig. 7. **State Estimation:** Kalman Filter block diagram

As seen from the diagram the first step is to get the system and measurement model. critically speaking both the models give a noisy estimate of the states.This because the linear dynamical model is not able to capture all the physical interactions of the state. Measurement is noisy as the sensors on the robot give noisy measurements of the state. The noisy estimate are set of estimated states which are used to get the approximation of the true state of the joint. Now we have the two pieces of information i.e. the previous state and the measurement of the current state. This information is used to compute the current state of joint. To get the better estimate the linear dynamical model is transformed probabilistic model with the help of Bayesian filtering through the mathematical model Gaussian distribution. This information is then passed through Kalman filter update method to get the true sate world.

VII. Results

A. Experimental Setup

This project's simulation platform consists of the open-source Robot Operating System (ROS), Gazebo, and Rviz. During the initial stage of our project, we utilized ROS and Gazebo to spawn our robot and control the position of its legs in the physics simulator.

B. Simulation

A universal robot description format (URDF) of the robot was created using the STL model of the robot. The URDF was spawned in Gazebo using a launch file, and utilized a controller manager for primitive joint control. The URDF spawned in Gazebo is shown below:

VIII. Conclusions and Future Work

In this study, we present two control objectives to control the locomotion of the quadruped robot.The proposed

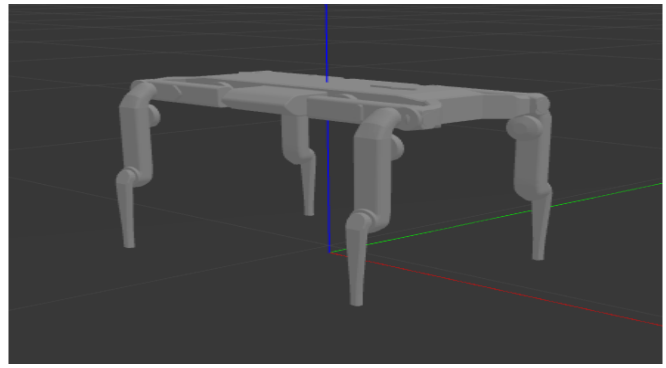

Fig. 8. **Solo-12 in Gazebo:** Spawned URDF in Gazebo

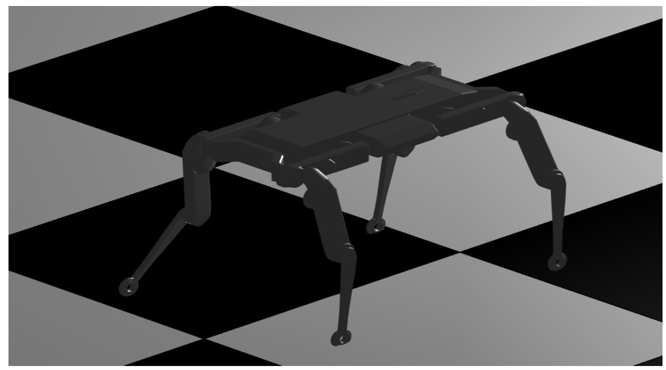

Fig. 9. **Matlab Simulation** Spawned URDF in Matlab

adaptive controller aims the leg of the robot to follow the desired trajectory, while the model predictive control works to predict the desired ground reaction force which is the input to the stance phase dynamics to obtain the required the torque to the joints.

References

[1] Kim, Donghyun & Carlo, Jared & Katz, Benjamin & Bledt, Gerardo & Kim, Sangbae. (2019). Highly Dynamic Quadruped Locomotion via Whole-Body Impulse Control and Model Predictive Control.

[2] P. -A. Léziart, T. Flayols, F. Grimminger, N. Mansard and P. Souères, "Implementation of a Reactive Walking Controller for the New Open-Hardware Quadruped Solo-12," 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 5007-5013, doi: 10.1109/ICRA48506.2021.9561559.

[3] https://github.com/open-dynamic-robot-initiative/open_robot_actuator_hardware

[4] Yan, Wei et al. "Whole-body kinematic and dynamic modeling for quadruped robot under different gaits and mechanism topologies." PeerJ Computer Science 7 (2021): n. pag.

[5] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018, pp. 1–9.

[6] H.-W. Park, P. M. Wensing, and S. Kim, "High-speed bounding with the MIT Cheetah 2: Control design and experiments:," The International Journal of Robotics Research, vol. 36, no. 2, pp. 167–192, Mar. 2017.

[7] D. Kim, J. Lee, J. Ahn, O. Campbell, H. Hwang, and L. Sentis, "Computationally-Robust and Efficient Prioritized Whole-Body Controller with Contact Constraints," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018.

[8] G. Bledt, P. M. Wensing, S. Ingersoll, and S. Kim, "Contact model fusion for event-based locomotion in unstructured terrains," in IEEE

International Conference on Robotics and Automation (ICRA), May 2018, pp. 1–8.

[9] D. Kim, S. Jorgensen, J. Lee, J. Ahn, J. Luo, and L. Sentis, “Dynamic Locomotion For Passive-Ankle Biped Robots And Humanoids Using Whole-Body Locomotion Control,” arXiv.org, Jan. 2019.

[10] S. Caron, Q.-C. Pham, and Y. Nakamura, “Stability of Surface Contacts for Humanoid Robots: Closed-Form Formulae of the Contact Wrench Cone for Rectangular Support Areas,” arXiv.org, Jan. 2015.

[11] J. Carpentier, M. Benallegue, N. Mansard, and J.-P. Laumond, “A kinematics-dynamics based estimator of the center of mass position for anthropomorphic system—a complementary filtering approach,”in IEEE-RAS 15th International Conference on Humanoid Robots(Humanoids), 2015, pp. 1121–1126.

[12] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt and S. Kim, ”Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control,” 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 1-9, doi: 10.1109/IROS.2018.8594448.

[13] K. Yoneda, K. Suzuki and Y. Kanayama, ”Gait planning for versatile motion of a six legged robot,” Proceedings of the 1994 IEEE International Conference on Robotics and Automation, 1994, pp. 1338-1343 vol.2, doi: 10.1109/ROBOT.1994.351302.