Dynamics of *QUAD V2*: A Low-Cost Quadrupedal Robot

TEXAS A&M UNIVERSITY Engineering

Ian Lansdowne

Motivation

Legged robots can be more agile and adaptable than robots with other forms of locomotion. The purpose of this project is to design and manufacture a low-cost robot capable of balancing and maneuvering on four legs to better understand the capabilities of walking robots.

Background

QUAD V2, rendered in Figure 1, was designed as a project with the Texas A&M Robotics Team and Leadership Experience (TURTLE) student organization to demonstrate quadrupedal locomotion with low-cost hardware. It was developed over 3 years and 4 design iterations to produce a robot meeting the specifications in Table 1. The material cost of *QUAD V2* is $2,400, much less to build than commercially available quadrupeds, which makes it a viable testbed for research in quadrupedal motion.

Figure 1. CAD model of *QUAD V2*.

Metric	Value
Mass	15 kg
Chassis Size (l×w×h)	77×22×18 cm
Leg Joint Length	19.2 cm
Maximum Joint Torque	32.3 N m
Joint Gear Reduction	19:1
Voltage	22.2 V
Battery Capacity	24 A h
Degrees of Freedom	12

Table 1. *QUAD V2* Specifications.

Mechanical Design

Each leg consists of 3 joints in series, each actuated by a brushless DC motor through a custom 3D-printed cycloidal gearbox (Figure 2). The legs are actuated with a tie-rod design inspired by Boston Dynamics' *Spot*. The chassis is designed for modularity and repairability with a PVC tube frame and detachable 3D-printed panels.

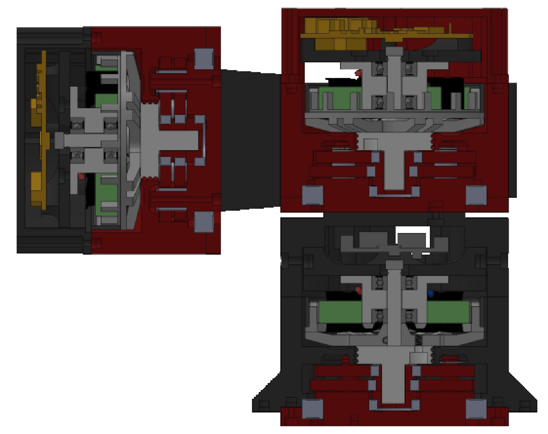

Figure 2. Section View of the Leg Actuators.

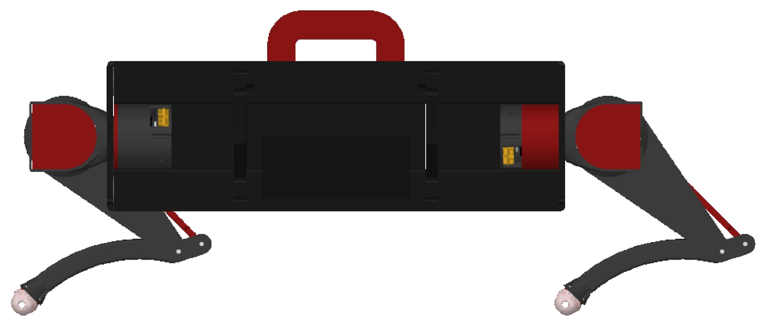

Figure 3. Section View of the Chassis.

Software Design

The robot is controlled by a Raspberry Pi 4 microprocessor running a real-time Linux kernel. The microprocessor communicates with the 12 motor controllers over a Controller Area Network (CAN) Bus to configure motor parameters and command servo actions (position, velocity, and torque). The software on the microprocessor is written in C++ using Eigen.

Control System

The inverse kinematics for the legs, with Equation 1 as the leg position input with respect to the chassis attachment point, produce joint angles for each leg in Equation 2. The sign of each angle varies depending on the mounting position of the leg.

$$\mathbf{p}_d = \left[x \; y \; z + r_f\right]^T \tag{1}$$

$$\boldsymbol{\theta} = \begin{bmatrix} \operatorname{arctan2}(z, y) - \arccos\left(\frac{l_1}{\sqrt{y^2+z^2}}\right) \\ \operatorname{arctan2}(\sqrt{y^2+z^2-l_1^2}, x) - \arccos\left(\frac{l_2^2+x^2+z^2-l_3^2}{2l_2\sqrt{x^2+y^2+z^2-l_1^2}}\right) \\ \arccos\left(\frac{l_1^2+l_2^2+l_3^2-x^2-y^2-z^2}{2l_2 l_3}\right) \end{bmatrix} \tag{2}$$

The inverse velocity kinematics take advantage of the fact that Jacobian is invertible. Therefore, the joint velocities can be calculated with Equation 4.

$$J^{-1} = \begin{bmatrix} \frac{\partial\theta_1}{\partial x} & \frac{\partial\theta_1}{\partial y} & \frac{\partial\theta_1}{\partial z} \\ \frac{\partial\theta_2}{\partial x} & \frac{\partial\theta_2}{\partial y} & \frac{\partial\theta_2}{\partial z} \\ \frac{\partial\theta_3}{\partial x} & \frac{\partial\theta_3}{\partial y} & \frac{\partial\theta_3}{\partial z} \end{bmatrix} \tag{3}$$

$$\dot{\boldsymbol{\theta}} = J^{-1}\dot{\mathbf{p}}_d \tag{4}$$

The inverse dynamics for the legs are computed using the Newton-Euler method described in Section 8.3 of [2]. The inverse dynamics in Equations 5 and 6 use spatial terms to describe the state, motion, and properties of the leg. τ is the motor output torque, $\mathcal{F}$ is a spatial (force-torque) wrench, $\mathcal{V}$ is a spatial (velocity) twist, $\mathcal{G}$ is a spatial inertia, $\mathcal{A}$ is a unitless screw axis at the joint center of mass, and T is an $SE(3)$ transformation. The adjoint (Ad and ad) operators are defined in [2].

$$\mathcal{F}_i = \underbrace{\mathrm{Ad}^{\mathrm{T}}_{T_{i+1,i}}(\mathcal{F}_{i+1})}_{\text{Propagated wrench}} + \underbrace{\mathcal{G}_i\dot{\mathcal{V}}_i}_{\text{Dynamic wrench}} \underbrace{-\mathrm{ad}^{\mathrm{T}}_{\mathcal{V}_i}(\mathcal{G}_i\mathcal{V}_i)}_{\text{Bias Wrench}} \tag{5}$$

$$\tau = \mathcal{F}_i^{\mathrm{T}}\mathcal{A}_i \tag{6}$$

Together, the inverse kinematics, inverse velocity kinematics, and inverse dynamics put the desired Cartesian foot motion into joint space. The joint angles, angular velocities, and torques are sent to the motor controllers to command the desired motion.
The desired robot motion is computed in the chassis frame, however. To compute the desired foot motion, the chassis twist, twist rate, and wrench are transformed to the feet frames. The foot force is approximated as an n^{th} of the desired wrench transformed to the foot frame, where n is the number of feet contacting the ground.

Margin of Static Stability

To compute the next desired step, an estimate is taken on which the robot is most likely to roll over. The rollover coefficient used is the normalized virtual power $\mathcal{K}_{ij}$ in Equation 7 [1]. The normalized virtual power is computed about each roll axis on the ground and is a function of robot motion. When $\mathcal{K}_{ij}$ is within a set threshold, the robot takes a step in the direction of lowest stability to re-stabilize itself. $\mathcal{S}_{ij}$ is the screw axis between feet i and j, $\hat{\mathcal{F}}$ is a normalized wrench on the chassis, and $\boldsymbol{\Delta}$ is the interchange operator defined in [1] which swaps the angular and linear terms.

$$\mathcal{K}_{ij} = \mathcal{S}_{ij}^T\boldsymbol{\Delta}\hat{\mathcal{F}} \tag{7}$$

Conclusion

The TURTLE Robotics *QUAD V2* is a uniquely inexpensive and capable robot, optimized for strength and modularity. The robot is particularly accessible as a low-cost option for research in quadrupedal robotics.

With the noted software and control developments beyond the initial design and assembly of the *QUAD V2* model, the robot will become a more useful and generalizable design. Future work beyond this point includes implementing an adaptive force distribution estimation for predicting foot forces, and computing a dynamic margin of stability to predict rollover in an inertial frame.

Source code and CAD files are available on GitHub:
github.com/turtle-robotics/quad-v2

References

[1] Joseph K. Davidson and Kenneth H. Hunt. *Robots and Screw Theory: applications of kinematics and statics to robotics*. Oxford University Press.

[2] Kevin M. Lynch and Frank C. Park. *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press.