



Autonomous Mini Road Header

A robotic system for independent small-scale tunnel boring in maintenance and electrical infrastructure applications

Slide 1



PROJECT OVERVIEW

Engineering an Autonomous Boring Solution

Project Mission

Design and build a miniature road header robot capable of autonomously boring small-diameter tunnels for critical infrastructure maintenance and electrical circuit installation purposes.

The system integrates mobile navigation with precision robotic manipulation to create a self-sufficient boring platform suitable for confined spaces and repetitive excavation tasks.

Development Timeline

Duration: 4 months

Sessions: 14 dedicated work periods

Focus Areas: Hardware integration, sensor fusion algorithms, and control system implementation

System Architecture & Components



Mobile Base Platform

4-wheel DC motor configuration provides omnidirectional mobility and stable chassis foundation for all integrated components



Robotic Arm Assembly

4 precision servo motors enable multi-axis articulation and accurate end-effector positioning during boring operations



Boring End-Effector

High-torque brushless motor drives the cutting mechanism for material excavation and tunnel formation



Control System

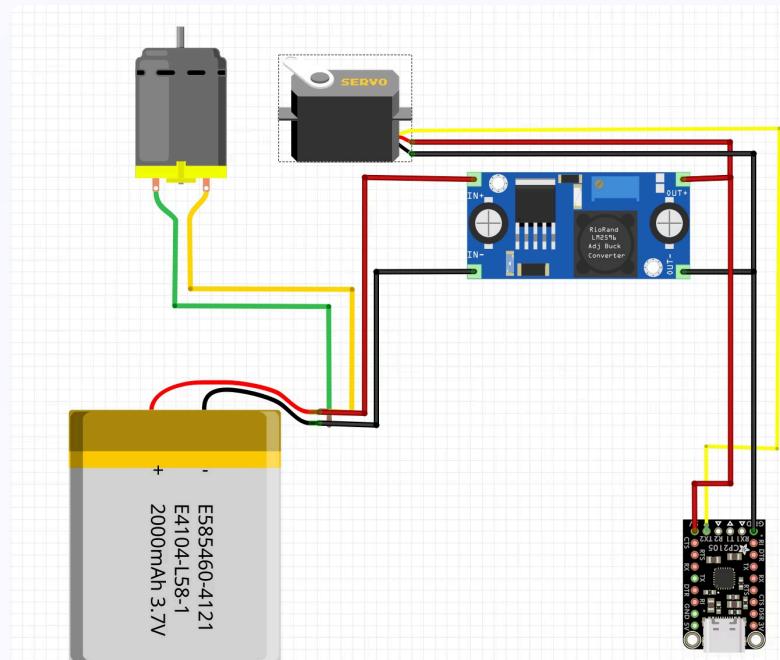
ESP32 microcontroller serves as central processor with custom shield for actuator communication and sensor integration

Electrical Architecture & Power Distribution

O1

Primary Power Source

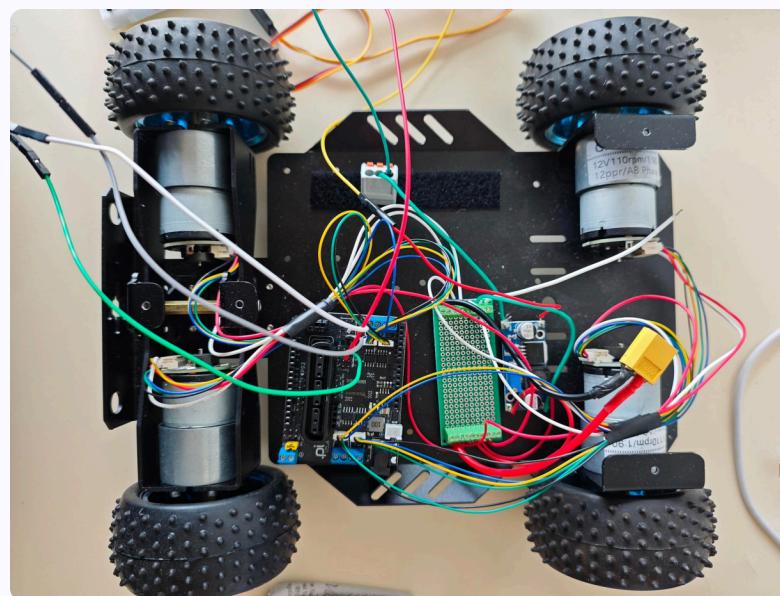
12V battery pack delivers main system power for high-current actuators and motors



O2

Voltage Regulation

Buck converter steps down voltage to power ESP32 controller and servo motors at appropriate levels

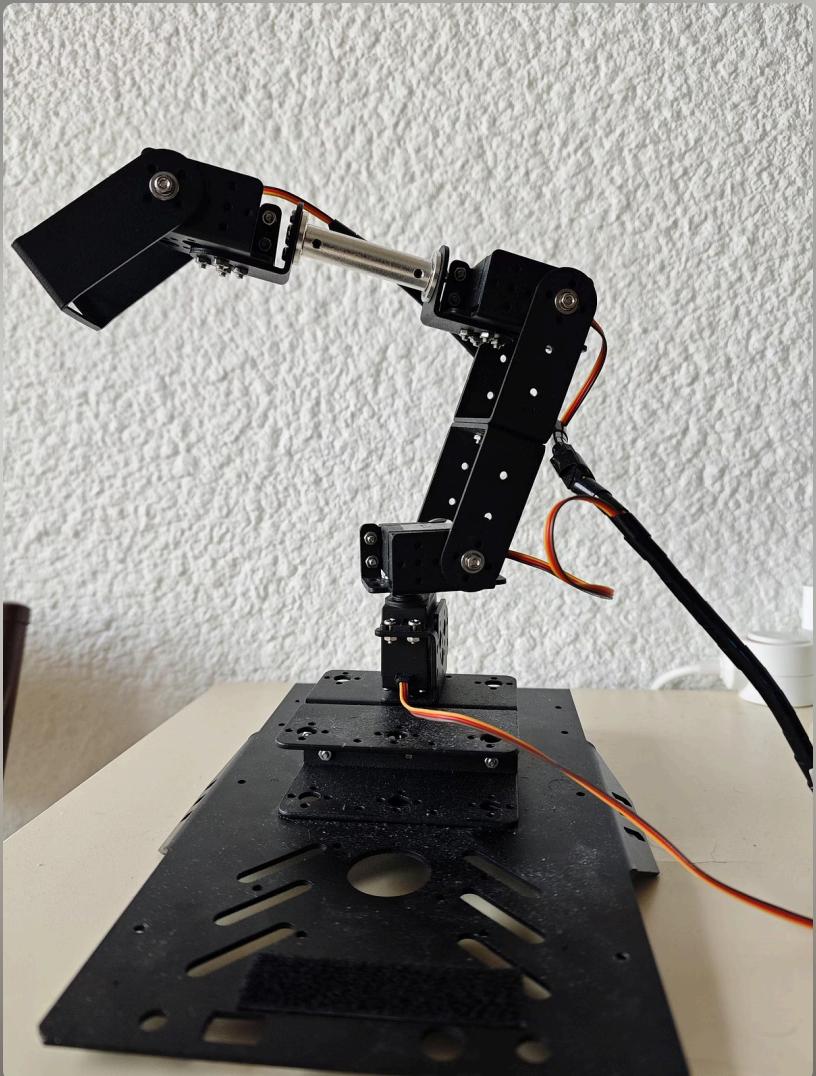


Detailed Fritzing diagram and corresponding real-life connections illustrate the power distribution and signal routing architecture.



Kalman Filter Implementation

Advanced state estimation technique fuses encoder data from servo motors to accurately predict robotic arm position in real-time



1

Sensor Inputs

Encoder angle measurements from 4 servo motors provide joint position data with inherent noise and uncertainty

2

Mathematical Model

State-space representation captures arm kinematics and dynamics for prediction and correction phases

3

Estimation Output

Optimal position estimate achieved through recursive Bayesian filtering of measurement and prediction data

Non-linear Mathematical Model

Non-linear Model

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau$$

M

Inertia matrix, $\in \mathbb{R}^{3 \times 3}$

C

Coriolis and centrifugal effects
matrix, $\in \mathbb{R}^{3 \times 3}$

g

Gravitational force vector, $\in \mathbb{R}^3$

τ

Generalized force (input) vector, $\in \mathbb{R}^3$

q

Joints position vector, $\in \mathbb{R}^3$

This represents the complete non-linear dynamics of the robotic arm system.

Linearization Around Equilibrium Point

Linearization

To simplify the non-linear mathematical model for control design and analysis, linearization is performed around a specific equilibrium point. For our robotic arm, this equilibrium point is assumed to be [0, 0, 0].

Key assumptions for linearization include:

- Trigonometric Simplifications: $\cos(0) = 1, \sin(0) = 0$
- Coriolis & Centrifugal Forces: $C(q, \dot{q})\ddot{q} = 0$
- Gravitational Force: $g(q) = g_0$
- Inertia Matrix: $M(q) = M_0$

With these key assumptions, the simplified linearized equation is obtained:

$$M_0 \cdot \ddot{q} + g_0 = \tau$$

This linearization simplifies the non-linear model for control design and analysis, making the system dynamics more manageable for feedback control system implementation.

State-Space Dynamic System Representation

Dynamic Model of the System

The linearized model of our robotic arm system can be transformed into a standard state-space representation, which is essential for advanced control design and Kalman filter implementation.

$$\dot{X}(t) = F \cdot X(t) + G \cdot u(t) + w(t)$$

The state vector $X(t)$ is a 6-dimensional vector containing the joint positions and velocities:

$$X(t) = [q_1, q_2, q_3, \dot{q}_1, \dot{q}_2, \dot{q}_3]^T$$

F matrix (6×6):

$$F = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}$$

G matrix (6×3):

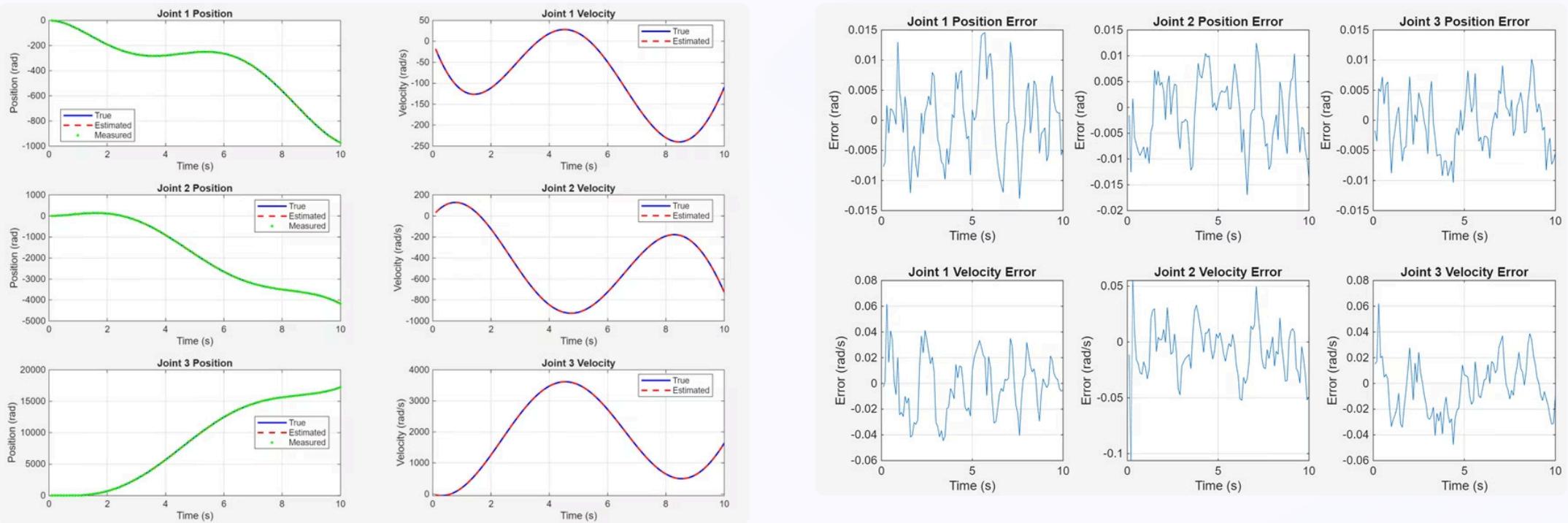
$$G = \begin{bmatrix} 0_{3 \times 3} \\ M_0^{-1} \end{bmatrix}$$

w vector (6×1 process noise vector):

$$w = \begin{bmatrix} 0_{3 \times 3} \\ M_0^{-1} \times g_0 \end{bmatrix}$$

This state-space form provides a concise and mathematically tractable model that is perfectly suited for integration with our Kalman filter implementation, allowing for robust state estimation and control of the robotic arm.

Kalman Filter Results



The first plot shows the estimation and tracking results for the arm's joints position and velocity, demonstrating the Kalman filter's ability to track the system state. The second plot displays the position and velocity errors, which remain bounded throughout the simulation, validating the filter's performance.

Physical parameters were derived from published literature on similar robotic models, as direct measurement was not feasible. The simulation employed a sinusoidal input signal: $U = [\sin(0.5t), \cos(0.3t), \sin(0.7t)]$ with a period $T = 0.1s$, providing comprehensive validation across multiple joint dynamics.

Control Software & Demonstration

Demo Implementation Status

Basic control software successfully demonstrates coordinated movement of both the mobile base and robotic arm for proof-of-concept validation.

The demonstration program showcases fundamental locomotion capabilities and arm articulation through simple command sequences executed by the ESP32 controller.

- ❑ **Development Note:** Software refinement was deprioritized to allocate more time to hardware integration and Kalman filter optimization.



Technical Obstacles & Solutions



Power Insufficiency

High amperage demands from boring motor exceeded initial battery specifications, requiring power system redesign and upgraded capacity



Motor Controller Issues

Original brushless boring motor required unavailable ESC



Component Failure

One servo motor burned out due to excessive continuous operation during testing



Controller Programming & Tuning

Customized Wemos D1 R32 (ESP32-based) microcontroller with integrated power-distribution shield required extensive firmware development and parameter tuning for real-time sensor fusion and motor control synchronization



Project Outcomes & Timeline

3

60%

14

Core Disciplines

MATLAB modeling, sensor fusion algorithms, electrical integration

Time on Circuits

Electrical troubleshooting and power tuning

Work Sessions

Dedicated development periods over 4 months

Development Timeline

Sessions	Period	Activities
1-3	September	Project ideation, task planning, component sourcing and procurement
4-7	Oct - Mid Nov	Kalman filter design, MATLAB modeling, algorithm tuning, results validation
8-11	Mid Nov - Dec	Hardware assembly, soldering, circuit testing, motor selection, power management, mechanical integration
12-14	Final Period	Demo software development, presentation preparation, technical report compilation

Key Learnings: Mastered state estimation theory, developed practical sensor fusion skills, and gained deep understanding of power electronics-critical foundations for advanced robotics applications.

Annexes: Kalman Filter Parameter Computation

Kalman Filter Parameter Computation

$$\begin{aligned}
 m_{11} &= p_1 + 2p_2 \cos(q_2) + 2p_3 \cos(q_3) + 2p_4 \cos(q_2 + q_3); \\
 m_{12} &= p_5 + p_2 \cos(q_2) + 2p_3 \cos(q_3) + p_4 \cos(q_2 + q_3); \\
 m_{13} &= p_6 + p_3 \cos(q_3) + p_4 \cos(q_2 + q_3); \\
 m_{21} &= m_{12}; \\
 m_{22} &= p_5 + 2p_3 \cos(q_3); \\
 m_{23} &= p_6 + p_3 \cos(q_3); \\
 m_{31} &= m_{13}; \\
 m_{32} &= m_{23}; \\
 m_{33} &= p_6.
 \end{aligned}$$

$M(q)$

$$\begin{aligned}
 c_{11} &= -p_2 \sin(q_2) \dot{q}_2 - p_3 \sin(q_3) \dot{q}_3 - p_4 \sin(q_2 + q_3) (\dot{q}_2 + \dot{q}_3); \\
 c_{12} &= -p_2 \sin(q_2) (\dot{q}_1 + \dot{q}_2) - p_3 \sin(q_3) \dot{q}_3 - p_4 \sin(q_2 + q_3) (\dot{q}_1 + \dot{q}_2 + \dot{q}_3); \\
 c_{13} &= -p_3 \sin(q_3) (\dot{q}_1 + \dot{q}_2 + \dot{q}_3) - p_4 \sin(q_2 + q_3) (\dot{q}_1 + \dot{q}_2 + \dot{q}_3); \\
 c_{21} &= p_2 \sin(q_2) \dot{q}_1 - p_3 \sin(q_3) \dot{q}_3 + p_4 \sin(q_2 + q_3) (\dot{q}_1); \\
 c_{22} &= -p_3 \sin(q_3) \dot{q}_3; \\
 c_{23} &= -p_3 \sin(q_3) (\dot{q}_1 + \dot{q}_2 + \dot{q}_3); \\
 c_{31} &= p_3 \sin(q_3) (\dot{q}_1 + \dot{q}_2) + p_4 \sin(q_2 + q_3) \dot{q}_1; \\
 c_{32} &= p_3 \sin(q_3) (\dot{q}_1 + \dot{q}_2); \\
 c_{33} &= 0.
 \end{aligned}$$

$C(q, \dot{q})$

$$\begin{aligned}
 g_1 &= p_7 \cos(q_1) + p_8 \cos(q_1 + q_2) + p_9 \cos(q_1 + q_2 + q_3); \\
 g_2 &= p_8 \cos(q_1 + q_2) + p_9 \cos(q_1 + q_2 + q_3); \\
 g_3 &= p_9 \cos(q_1 + q_2 + q_3).
 \end{aligned}$$

$g(q)$

These detailed mathematical formulas are essential for implementing the Kalman filter with accurate physical parameters, ensuring precise state estimation for the robotic arm system.