Technical Design Paper by Automated Architects for the Robotics Dojo Competition 2025

Basil Karanja, Cephas Mutua, Winston Abraham, 0.G Odhiambo

Abstract—**This paper presents the design, fabrication, and deployment of the Automated Architects Robot, developed for the Robotics Dojo 2025 competition under the AFRICA-ai-JAPAN Project at JKUAT. The robot was engineered to traverse challenging terrain, including ramps, sawdust tracks, rocky sections, and obstructed paths, while performing object manipulation tasks such as load identification and targeted placement. The system integrates Simultaneous Localization and Mapping (SLAM) using RPLIDAR A1, powered by Robot Operating System (ROS), and employs a modular three-tier mechanical chassis for stability and sensor optimization. The electrical architecture emphasizes simplicity and reliability, using closed-loop motor control with encoder feedback and PID regulation. Experimental trials guided iterative improvements, notably in caster wheel design and battery performance. The design demonstrates the balance of capability versus robustness and reliability versus complexity, exemplifying collaborative engineering under time constraints. The project contributes to advancing robotics research and practical skills among engineering students in Kenya.**

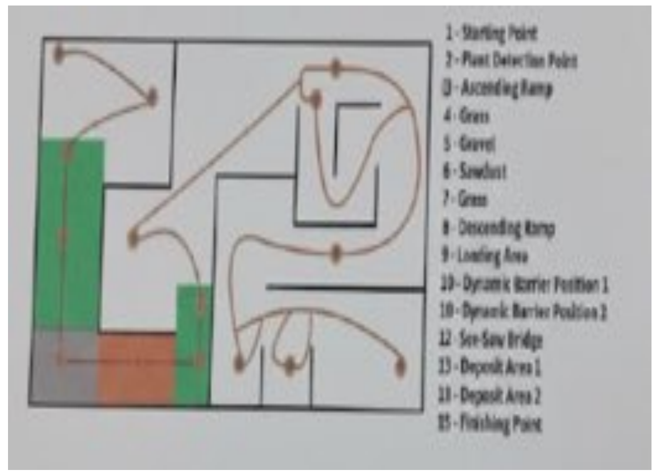
I. INTRODUCTION

The Robotics Dojo competition, organized under the AFRICA-ai-JAPAN Project by JICA at Jomo Kenyatta University of Agriculture and Technology (JKUAT), aims to strengthen robotics research capacity in Kenya. The competition provides a platform for students to design and deploy autonomous robotic systems capable of real-world navigation and manipulation.

This year's Robotic Dojo 2025 competition was structured as a multi-modal test of autonomous system capabilities, requiring successful navigation and task execution within a complex, prescribed environment.

The challenge began at the *Starting Point (1)* and demanded that robots traverse diverse terrains and obstacles. Mobility was a key requirement, as the course included sections of *Grass (4, 7)*, *Gravel (5)*, and *Sawdust (6)*, alongside demanding transitions over both an *Ascending Ramp (3)* and a *Descending Ramp (8)*. Critical operational requirements of the competition necessitated successful interaction with the environment, including locating and processing information from the *Plant Detection Point (2)*, performing a necessary payload acquisition at the *Loading Area (9)*, and executing precise object manipulation to deposit items at two distinct locations: *Deposit Area 1 (13) and Deposit Area 2 (14)*.

Furthermore, the challenge placed a high premium on robust path planning and stable control, as robots were required to navigate past two *Dynamic Barrier Positions (10, 11)* and successfully traverse the instability of the *See-Saw Bridge (12)* before reaching the final objective at the *Finishing Point (15)*. The design and implementation of a robust control and sensor fusion architecture were essential for meeting these complex operational demands.



Autonomous robots face significant challenges in localization accuracy, stability across uneven terrain, and real-time decision-making. This work addresses these challenges through a design strategy prioritizing modularity, reliability, and robustness while operating under the practical

II. DESIGN STRATEGY

The overall design strategy for the Automated Architects Robot majorly focused on realibility of the Robot. This approach intentionally navigated the inherent trade-offs between reliability versus complexity and capability versus robustness.

A. Trade-off Analysis and Prioritization

Rather than coming up with a robot with maximum system complexity, which would have increased potential points of failure within the system and demanded extensive debugging time cause of the complexity, the team prioritized the integrity of the robot performing foundational capabilities. The core team requirements—stable locomotion, accurate localization, and basic object placement—were deemed non-negotiable. Consequently, design time was allocated toward testing and validating fundamental subsystems rather than adding advanced, but potentially unstable and unused features.

This translated to the following strategic choices:

- Reliability over Complexity**:** The electrical architecture intentionally favored simplicity. Utilizing a carefully soldered perfboard to functionally mimic a PCB, despite being less sophisticated than a custom-manufactured board, dramatically reduced complexity in the wiring harness. This mitigated systemic electrical failure points and maximized the time available for essential field trials and tuning (e.g., PID controller calibration and SLAM refinement).
- Robustness over Capability**:** The robot's three-tiered mechanical chassis and custom-fabricated rubber caster wheels were chosen to maximize stability and mobility across the course's heterogeneous terrain (including the Gravel and Sawdust). This structural robustness was prioritized over the inclusion of a highly complex manipulation arm, which could have introduced significant weight, instability, and control difficulty. The resulting system possesses sufficient capability (basic loading/placement), offering a higher overall probability of mission success than a fragile, feature-rich alternative.

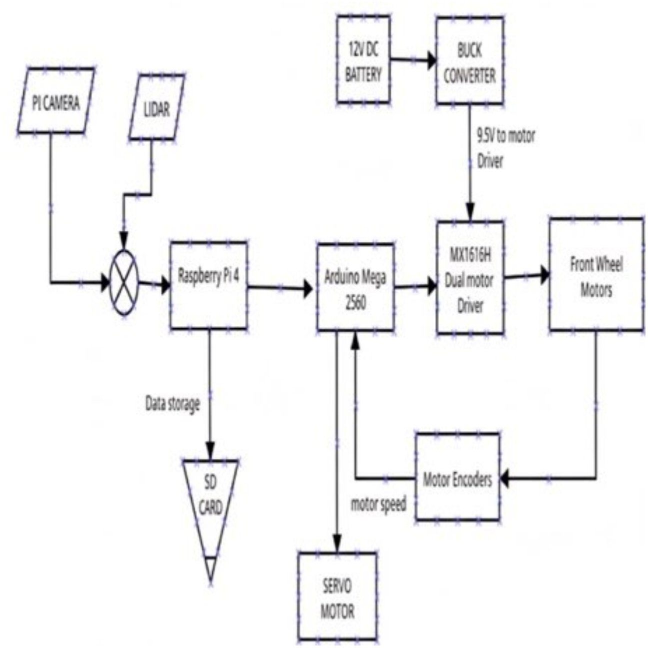
III. Vehicle Design

A. Electrical Systems Architecture

The electrical wiring implemented on this robot was designed with a focus on minimizing complexity and maximizing maintainability hence prioritizing the use of fewer wires to simplify the overall system and facilitate easier troubleshooting. Due to time constraints in the manufacturing process for a custom Printed Circuit Board (PCB), the team elected to utilize a perfboard (perforated board**)**. Soldering was carefully carried out on this perfboard to functionally mimic the connectivity and organization of a dedicated PCB, providing a reliable and robust platform for integrating the necessary electronic components under strict deadlines

Key components:

- Dual Motor Driver- mx 1616h
- Buck converter
- Arduino mega microcontroller
- 12V battery supply
- RPLIDAR A1
- Rasberry Pi 4



1. Closed Loop Motor Drive Wiring

Robotic systems require accurate and responsive motor control to ensure reliable locomotion and navigation. This project implements a closed-loop motor control architecture using encoder feedback to regulate motor speed and direction The outline of the design and implementation of a closed-loop DC motor control system was done as follows.

key components

- Dual Motor Driver – mx 1616h
- Buck Converter
- Arduino Mega Microcontroller
- 12V Battery Supply

The system uses an Arduino Mega as the main microcontroller, a mx1616h dual motor driver, and a buck converter for voltage regulation. The main objective is to achieve precise motor control by using encoder feedback.

Specifications of the components used;

Arduino Mega Micro-controller:

It serves as the central controller. It generates PWM signals for motor speed control and reads encoder data via interrupt pins. Arduino Mega was preferred in this project because of its multiple PWM pins and multiple interrupt pins which are very vital for accurate encoder pulse counting.

Mx 1616h Dual H-Bridge Motor driver:

This driver receives PWM signals from the Arduino to control motor direction and speed. Its operating voltage is 2V to 10V but for this particular design we supplied it with 9.5V for optimal performance and longevity.

Buck Converter:

The buck converter steps down the 12V battery supply to 9.5V, providing a safe and stable voltage for the motor driver. The 0.5V margin protects against overvoltage and enhances reliability.

2. Arduino Mega 2560 Pin Configuration

Motor Control (PWM OUTPUT)

Motor	Function	Arduino Pin	PWM Timer	MX1616H Pin
Right Motor	Forward	Digital 4	Timer 2	IN3
Left Motor	Forward	Digital 7	Timer 0	IN1
Right Motor	Reverse	Digital 5	Timer 2	IN4
Left Motor	Reverse	Digital 6	Timer 0	IN2

Encoder Feedback (Interrupt Inputs)

Motor	Channel	Arduino Pin	Interrupt	Signal Type
Left Motor	A	Digital 2	INT0	Quadrature A
Left Motor	B	Digital 3	INT1	Quadrature B
Right Motor	A	Digital 18	INT5	Quadrature A
Right Motor	B	Digital 19	INT4	Quadrature B

3. Power Connections

- 5V Pin → Encoder 1 VCC, Encoder 2 VCC
- GND Pin → Encoder 1 GND, Encoder 2 GND, MX1616H GND

4. Data Flow

- LIDAR Data: RPLIDAR A1 transmits mapping data to the Raspberry Pi via USB.
- Control Commands: Raspberry Pi sends motor instructions to the Arduino Mega via USB.
- Encoder Feedback: Encoders relay pulse data to the Arduino Mega, which processes and forwards relevant information to the Raspberry Pi.
- Motor Control: PWM signals from the Arduino Mega drive the MX1616H motor driver, which actuates the DC motors.

Component	Power Source	Voltage	Current
Arduino Mega	Raspberry Pi USB	5V	500mA
Raspberry Pi 4	External Power Bank	5V	2.5A – 3A
RPLIDAR A1	Raspberry Pi USB	5V	500mA
Two Encoders	Arduino Mega 5V Pin	5V	100mA
MX1616H Driver	Buck Converter Output	9.5V	2.5A

B. Control Strategy

1. Closed-Loop Feedback

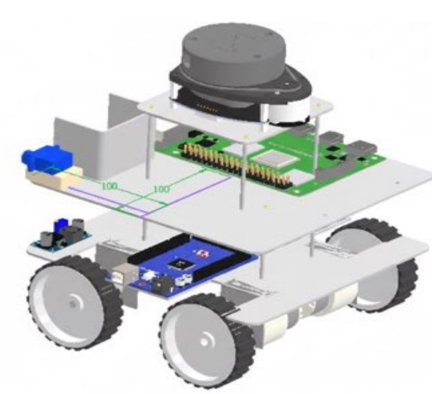
Quadrature encoders provide high-resolution feedback, enabling precise velocity and position control. Interrupt-driven signal processing ensures minimal latency and accurate pulse counting.

2. PID Regulation

A proportional-integral-derivative (PID) controller is implemented on the Arduino to adjust motor speed based on encoder feedback. This minimizes error and stabilizes motion under varying load conditions.

C. Structural Design

The mechanical structure of the robot was designed to achieve stability, maneuverability and component organization. We settled on a three-tiered chassis, a modular approach that simplifies assembly and maintenance.



1. *Tiered Chassis Design*

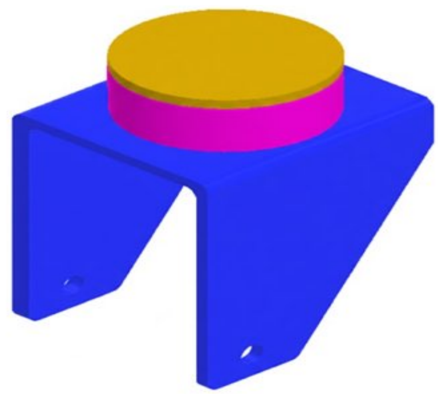
The tiers are connected by pillars with screw-in fittings, allowing for easy disassembly and reassembly. This modularity was particularly beneficial during the prototyping phase, enabling rapid modifications and access to internal components.

- *First Tier (Base)*: This is the foundational tier, designed to hold the heaviest components to maintain a low center of gravity (*CoG*). This strategic placement enhances the robot's stability, which is critical for traversing uneven terrain and preventing tipping. The first tier houses the battery holder, the Arduino Uno microcontroller, the buck converter, and the motor driver.
- *Second Tier (Mid-section)*: Positioned above the base, this tier provides a dedicated space for the Raspberry Pi single-board computer and a custom-designed loading dock. The loading dock is integrated into this tier to facilitate easy object manipulation, which is essential for the robot's operational tasks.
- *Third Tier (Top)*: The uppermost tier functions as a stand for the RPLIDAR sensor. By elevating the LiDAR, we ensured a *clear and unobstructed 360-degree view* of the environment, a crucial requirement for accurate Simultaneous Localization and Mapping ***(****SLAM*) and obstacle detection.

2. *Wheels and Drivetrain*

The robot's mobility system was engineered to handle challenging, off-road conditions, such as the sawdust and rocky terrain of the game field.

- Off-Road Drive Wheels: The robot uses *85mm-diameter* off-road wheels (*38mm thick*) for its forward motion. This configuration was chosen for a ***forward-wheel-drive*** system due to its inherent advantages:
    - Better Traction: Forward-wheel drive provides superior traction, which is essential for navigating slippery or loose surfaces like sawdust.
    - Improved Manoeuvrability: With the front wheels providing both power and steering, the robot's steering and control are highly responsive, making it easier to navigate tight spaces.
- Two Rear Custom Caster Wheel: During initial tests, it was observed that standard caster wheels made of Polypropylene (PP) struggled and often became stuck in the game field's terrain. To overcome this, we designed and fabricated a ***custom caster wheel holder*** specifically for ***6mm diameter rubber wheels***. The custom design incorporates a 12mm shaft bearing to ensure smooth, reliable rotation and prevent jamming. This custom solution was key to ensuring the robot could move freely and reliably across the entire field.



D. Software Implementation

1. *SLAM AND LiDAR Integration*

SLAM Overview

SLAM (Simultaneous Localisation and Mapping) is a computational program that allows a mobile vehicle or robot to create a map of an unknown environment while simultaneously keeping track of its own location within the map.

LiDAR Functionality

LiDAR (Light Detection and Ranging) enables SLAM by emitting rapid, high-frequency laser pulses that propagate through the air at the speed of light. When these pulses encounter an obstacle, they are reflected to the LiDAR sensor. By measuring the time interval between emission and reception—known as the time of flight—the system calculates the distance to the object using the formula:

$$D = \frac{c.t}{2}$$

where c is the speed of light and t is the time of flight.

2. Motion Estimation

Accurate motion estimation is critical for reliable autonomous navigation. In this system, odometry data is derived from encoder feedback and motor telemetry, providing real-time estimates of the robot's displacement and orientation. These estimates are fused with LiDAR-based spatial observations to enhance both localization and environmental mapping.

As the robot moves, LiDAR continuously scans the environment, generating point clouds that are compared against previously mapped features. This comparison allows the system to differentiate between newly encountered landmarks and reobserved ones, reducing cumulative drift and improving map fidelity.

By combining proprioceptive data (from encoders) with exteroceptive data (from LiDAR), the system achieves robust pose estimation even in environments with partial occlusions or dynamic obstacles. This multimodal approach is foundational to modern SLAM frameworks and significantly improves the reliability of autonomous decision-making.

3. ROS

ROS (Robot Operating System) was utilised for the development of the Robot since it provides a modular framework for distributed robotic control.

Node Implementation

Nodes are individual processes that perform a specific task. In our Robot, we have implemented the following nodes:

- LiDAR Node: Captures real-time point cloud data from the RPLIDAR A1.
- SLAM Node: Processes LiDAR data to generate a dynamic map and estimate the robot's pose using SLAM.
- Motor Control Node: Interfaces with the Arduino Mega to send PWM commands to the MX1616H motor driver.
- Encoder Feedback Node: Reads quadrature encoder signals to compute odometry, which is fused with LiDAR data for motion estimation.
- Object Manipulation Node: Controls the servo motor for load placement based on color recognition.

Communication and Coordination

- Topics: Nodes communicate using an anonymous publish-subscribe model. Data is published to a specific topic, and any node subscribed to that topic receives the data.
- Messages: The actual data structures sent over topics. They can range from simple numerical values to complex data types like point clouds and camera images.
- ROS Master (ROS 1): A central server that coordinates nodes by helping them find and connect to each other. Once connected, data flows directly between the nodes.
- Services: Used for one-time, request-response interactions, in contrast to the continuous data streams of topics. A node can provide a service that another node can call upon.

A. Experimental Results

This section should briefly describe how the team accomplishes testing (*e.g.,* unit and integration testing, simulation, etc.) and provide some notion of how much testing has occurred as of the technical design paper submission. Note that the *actual results* reported in this section will not affect the team's technical design paper score (*e.g.,* reporting a high performance will neither help nor hurt the technical design paper score). This section should also discuss any studies, calculations, or estimates that the team has performed in the areas of reliability and robustness (e.g., failure analysis, reliability modeling, structural analysis, etc.).

B. Acknowledgements (optional)

This is an optional section that teams may wish to utilize to acknowledge particular assistance, sponsors, etc.

C. References

As with any scientific publication, original ideas and content that are not generated by the paper's authors should be properly cited. While there are several reference styles, the Robotics Dojo technical design paper uses the IEEE style, which is detailed in the Appendix. This style uses the bracketed reference, which should be used in line with text as in "The work in [x] states that…" This section does not count against the page limit.

D. Appendix—Situational Awareness (optional)

The Appendix is optional and does not count against the page limit. Recall that a foundational purpose of Robotics Dojo is to strengthen and enhance the community. Therefore teams are encouraged to share their approaches to solving operational concerns relevant to the unmanned systems community. A significant challenge to adoption of unmanned systems is user trust. Human users have a need to understand what the unmanned system is doing and why; users must have confidence that the system is behaving as intended. This is particularly important as emergent behaviors become more common. Discuss how you would approach providing information to users such that they would have awareness of the unmanned systems situation, and thus confidence in the unmanned systems intentions. Although this appendix is optional, a special award could be designated for this topic.

APPENDIX

This Appendix is taken from the IEEE Transactions template on the IEEE website, and should be followed for citing references, (https://template-selector.ieee.org/secure/templateSelector/publicationType).

Basic format for books:

[1] J. K. Author, "Title of chapter in the book," in *Title of His Published Book, x*th ed. City of Publisher, Country if not

[2] USA: Abbrev. of Publisher, year, ch. *x*, sec. *x*, pp. *xxx–xxx.*

Examples:

[3] G. O. Young, "Synthetic structure of industrial plastics," in *Plastics,* 2nd ed., vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.

[4] W.-K. Chen, *Linear Networks and Systems.* Belmont, CA: Wadsworth, 1993, pp. 123–135.

Basic format for periodicals:

[5] J. K. Author, "Name of paper," *Abbrev. Title of Periodical*, vol. *x,* no. *x,* pp. *xxx-xxx,* Abbrev. Month, year.

Examples:

[6] J. U. Duncombe, "Infrared navigation—Part I: An assessment of feasibility," IEEE *Trans. Electron Devices*, vol. ED-11, no. 1, pp. 34–39, Jan. 1959.

[7] E. P. Wigner, "Theory of traveling-wave optical laser," *Phys. Rev.*, vol. 134, pp. A635–A646, Dec. 1965.

[8] E. H. Miller, "A note on reflector arrays," *IEEE Trans. Antennas Propagat.*, to be published.

Basic format for reports:

[9] J. K. Author, "Title of report," Abbrev. Name of Co., City of Co., Abbrev. State, Rep. *xxx*, year.

Examples:

[10] E. E. Reber, R. L. Michell, and C. J. Carter, "Oxygen absorption in the earth's atmosphere," Aerospace Corp., Los Angeles, CA, Tech. Rep. TR-0200 (4230-46)-3, Nov. 1988.

[11] J. H. Davis and J. R. Cogdell, "Calibration program for the 16-foot antenna," Elect. Eng. Res. Lab., Univ. Texas, Austin, Tech. Memo. NGL-006-69-3, Nov. 15, 1987.

Basic format for handbooks:

[12] Name of Manual/Handbook, *x* ed., Abbrev. Name of Co., City of Co., Abbrev. State, year, pp. *xxx-xxx.*

Examples:

[13] *Transmission Systems for Communications*, 3rd ed., Western Electric Co., Winston-Salem, NC, 1985, pp. 44–60.

[14] *Motorola Semiconductor Data Manual*, Motorola Semiconductor Products Inc., Phoenix, AZ, 1989.

Basic format for books (when available online):

[15] Author. (year, month day). *Title.* (edition) [Type of medium]. *volume (issue).* Available: site/path/file

Example:

[16] J. Jones. (1991, May 10). *Networks.* (2nd ed.) [Online]. Available: http://www.atm.com

Basic format for journals (when available online):

[17] Author. (year, month). Title. *Journal.* [Type of medium]. *volume (issue),* pages. Available: site/path/file

Example:

[18] R. J. Vidmar. (1992, Aug.). On the use of atmospheric plasmas as electromagnetic reflectors. *IEEE Trans. Plasma Sci.* [Online]. *21(3),* pp. 876–880. Available: http://www.halcyon.com/pub/journals/21ps03-vidmar

Basic format for papers presented at conferences (when available online):

[19] Author. (year, month). Title. Presented at Conference title. [Type of Medium]. Available: site/path/file

Example:

[20] PROCESS Corp., MA. Intranets: Internet technologies deployed behind the firewall for corporate productivity. Presented at INET96 Annual Meeting. [Online]. Available: http://home.process.com/Intranets/wp2.htp

Basic format for reports and handbooks (when available online):

[21] Author. (year, month). Title. Company. City, State or Country. [Type of Medium].Available: site/path/file

Example:

[22] S. L. Talleen. (1996, Apr.). The Intranet Archi-tecture: Managing information in the new paradigm. Amdahl Corp., CA. [Online]. Available: http://www.amdahl.com/doc/products/bsg/intra/infra/html

Basic format for computer programs and electronic documents (when available online): ISO recommends that capitalization follow the accepted practice for the language or script in which the information is given.

Example:

[23] A. Harriman. (1993, June). Compendium of genealogical software. *Humanist.* [Online]. Available e-mail: HUMANIST@NYVM.ORG Message: get GENEALOGY REPORT

Basic format for patents (when available online):

[24] Name of the invention, by inventor's name. (year, month day). *Patent Number* [Type of medium]. Available: site/path/file

Example:

[25] Musical toothbrush with adjustable neck and mirror, by L.M.R. Brooks. (1992, May 19). *Patent D 326 189* [Online]. Available: NEXIS Library: LEXPAT File: DESIGN

Basic format for conference proceedings (published):

[26] J. K. Author, "Title of paper," in *Abbreviated Name of Conf.*, City of Conf., Abbrev. State (if given), year, pp. *xxxxxx.*

Example:

[27] D. B. Payne and J. R. Stern, "Wavelength-switched pas- sively coupled single-mode optical network," in *Proc. IOOC-ECOC,* 1985, pp. 585–590.

Example for papers presented at conferences (unpublished):

[28] D. Ebehard and E. Voges, "Digital single sideband detection for interferometric sensors," presented at the 2nd Int. Conf. Optical Fiber Sensors, Stuttgart, Germany, Jan. 2-5, 1984.

Basic format for patents:

[29] J. K. Author, "Title of patent," U.S. Patent *x xxx xxx*, Abbrev. Month, day, year.

Example:

[30] G. Brandli and M. Dick, "Alternating current fed power supply," U.S. Patent 4 084 217, Nov. 4, 1978.

Basic format for theses (M.S.) and dissertations (Ph.D.):

[31] J. K. Author, "Title of thesis," M.S. thesis, Abbrev. Dept., Abbrev. Univ., City of Univ., Abbrev. State, year.

[32] J. K. Author, "Title of dissertation," Ph.D. dissertation, Abbrev. Dept., Abbrev. Univ., City of Univ., Abbrev. State, year.

Examples:

[33] J. O. Williams, "Narrow-band analyzer," Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, 1993.

[34] N. Kawasaki, "Parametric study of thermal and chemical nonequilibrium nozzle flow," M.S. thesis, Dept. Electron. Eng., Osaka Univ., Osaka, Japan, 1993.

Basic format for the most common types of unpublished references:

[35] J. K. Author, private communication, Abbrev. Month, year.
[36] J. K. Author, "Title of paper," unpublished.
[37] J. K. Author, "Title of paper," to be published.

Examples:

[38] A. Harrison, private communication, May 1995.
[39] B. Smith, "An approach to graphs of linear forms," unpublished.
[40] A. Brahms, "Representation error for real numbers in binary computer arithmetic," IEEE Computer Group Repository, Paper R-67-85.

Basic format for standards:

[41] *Title of Standard*, Standard number, date.

Examples:

[42] IEEE Criteria for Class IE Electric Systems, IEEE Standard 308, 1969.
[43] Letter Symbols for Quantities, ANSI Standard Y10.5-1968.