PAPER • OPEN ACCESS

Automated Quadruped Robot Simulation using Internet of Things and MATLAB

To cite this article: Richik Ray *et al* 2021 *J. Phys.: Conf. Ser.* **2115** 012022

Automated Quadruped Robot Simulation using Internet of Things and MATLAB

Richik Ray[1*], Rishita Shanker[1], Harsh Gupta[1], Mohit Sharan[1], Swagatika Mohanty[1]

Vellore Institute of Technology, Chennai Campus, Tamil Nadu, India.

*richik.ray2018@vitstudent.ac.in

Abstract. In this paper, a MATLAB Simulink model of a Quadruped Robot is presented alongside its remote, control and monitor user interface that has been developed by using the fundamentals of Internet of Things on a Node-Red Flow and the FRED-Cloud Server. Robotics and Automation over the recent years have developed exponentially and hence have been a key factor in the rise of Industry 4.0 which has usurped manual supervision and operation in industrial and manufacturing processes around the globe. The design and creation of technologically advanced robots integrated with computer-based software for their automation has not only successfully made the tasks facile to manage within short spans of time, but also has increased the efficiency notably. The stability and mobility of quadruped robots is considered to be ideal on differing terrains with minimal subtle changes, thereby making it an asset. Internet of Things on the other hand, has paved its way over the control of robots as well, with its unparalleled benefits. This paper is focused on the design and execution of the Quadruped model which includes the observation of the various significant graphs achieved post simulation with respect to electrical values such as power and current consumption, and a visual animation of the robot running in the workspace. Furthermore, a single platform is developed and displayed that allows a user to log in for security purposes and thereby, operate and monitor the functions and conditions of the bot easily, ranging from remote visual support, directional integrity, damage control and more, without the need of multiple platforms to carry out varying tasks with respect to control.

1. Introduction

Robots have been fundamental in carrying out operations for humans since ages in industries for manufacturing, storing, and similar purposes, while at homes, have been efficient for conducting household chores. Being a programmable machine, the tasks performed by a robot can be encoded and controlled autonomously by a microcontroller board. Quadruped robots as discussed in this paper, have 2 sets of front and rear limbs, 4 in total, and the design and mobility are developed from an in-depth analysis of the gait movement of 4-legged animals. In comparison to wheeled robots, quadrupeds are preferred due to higher functionality when it comes to rougher terrains and climbing the stairs of a building [1]. To achieve optimum conditions for the design, multiple variables are entered and tests are run to ensure various factors that affect the overall mechanism of the robot, hence it is a wise technique to run simulations for an ideal model.

The simulations allow the developers to study the various graphical representations and understand the necessary values for various factors so that the final prototype has minimum changes due to practical reasons. These virtual experiments analyse the locus of the centre of gravity with respect to its gait, thereby being the basis of the overall movement of the bot [2]. In the case of quadrupeds that are more advanced, the gait design ensures straight path movement while curved paths are handled using motion sensors for any obstacles and utilising the longitudinal axis to ensure differential strokes in each of the limbs [3]. Furthermore, quadrupeds are be classified on the basis of their leg structure in two categories, precisely the Mammal and the Sprawling type. The mechanical structure can be designed to suit the motion of the robot in real-time, ranging from normal straight paths to climbing walls [4]. High-rise building construction procedures over time have incorporated automation and robotics in densely populated countries, such as China. Robots are operated to carry out the three-part methodology in such cases, i.e.; earth and foundation work, superstructure erection and façade installation [5]. IoT over time, has been integrated with smart technology, machine learning algorithms, etc., to develop complex and intelligent systems that when incorporated with actuators and sensors, are highly efficient for data processing and instantaneous transfer. This is observed in various healthcare and industrial scenarios, where equipment is controlled and data is monitored based on IoT [6]. Standard protocols like TCP/IP are widely used in machine-to-machine communication, and is being applied to several fields such as agriculture, military, robotics, etc. This leads to a plethora of technologies like Cloud Computing, Artificial Intelligence, and IoT to form IoRT, i.e., Internet of Robotic Things. Humanoid, Mobile, Swarm and Flying bots fall in this category as applications of IoRT, finding diverse objectives in the fields mentioned previously [7].

New methodologies to control the functions carried out by a bot are developed constantly, which is inclusive of speech-recognition software that analyses voice commands and interprets the same as procedures to be carried out. Besides nodes for voice recognition in Node-Red, Python has modules that can recognize around 119 languages without any contingencies. This methodology comes in handy when equipped with ML algorithms. Robots can effectively be configured with ultrasonic sensors that detect any obstacles on their path and can be programmed to traverse accordingly, yet with the assistance of a Wi-Fi module or a microcontroller, such as Raspberry Pi 3, the user can directly voice his/her commands into the interface and via IoT principles, the data is received and hence the bot can be controlled [8]. Robots integrated with IoT can vary in their operations and designs, suited to fit their functions and necessities. IoT has multiple levels or layers in each of its protocols that allow bots to be modelled for their differing processes. InterBot 1.0 was an example of the same, as discussed in the reference mentioned. It was meant for environment monitoring via IoT and was efficient in terms of temperature, humidity, gas-sensing, etc. and ha a long-range communication system dependent on a 2.4 Ghz 6-channel remote and a Bluetooth Module [9].

Besides construction procedures, the automotive industry incorporates automation by collaborating intelligent robots with machines and humans for smart manufacturing processes. Not only ranging to fully electric vehicles, leading companies like Tesla have smart devices and upgrades to the Model S and X that allow autopilot on freeways, and the data that is fed to humanoids to teach driving is altered with voice-control and similar necessary changes to do so [10].Quadruped robots date back to the TITAN series in the 1980s, developed by the Hirose-Yoneda Laboratory, followed by BigDog designed by Boston Dynamics in 2005, to the latter’s successor, LS3 (Legged Squad Support System), that was used for military purposes, developed as a DARPA endeavour. Studies are conducted on similar legged robots including different approaches, such as whole-body nonlinear model predictive control for rigid body systems. Dynamic models are developed and tested to find optimal conditions and settings for movement, transferability and robustness [11]. The MIT Mini Cheetah is an example when it comes to the demonstration of controller capabilities to exploit angular momentum of the body for disturbance recovery on two feet, which can be applied and extended to other quadrupeds. Variational-based linearization approaches in such situations where friction limitations need to be handed are efficient [12].

MATLAB can be integrated with microcontrollers such as Arduino, to build a control and visualization environment with the board acting as the final controller. With proper mechanical structuring, internal and external circuit design, dynamic and kinetic analysis, etc., a robot building procedure has multiple steps until a finished product is achieved [13].MATLAB allows a user to create a simulation file for any robotic structure he/she wishes to design, and in this simulation, a similar file is created that shows the dynamic motion of a 4-legged running robot along with its various graphs for overall and intricate analysis. SimMechanics on MATLAB combined with Newton-Euler formulation helps determine joint torque values, to give an example. This is done in this paper using SSC Explore [14].To design wireless networks for control, modules like Xbee can be used and combined with the robot via a graphical user interface, similar to the one discussed in this paper. In a prototype, all results and findings achieved by the bot can be accessed easily through this low-cost communication system [15]. This simulation and its results will help us analyze the working of the robot and give us a better understanding of its various applications. Following this, a proposal for the integration of this system with a control and communication design developed on the fundamentals of Internet of Things is presented. This control system shall include an administrative voice control system for the bot, and also provide the user a digital environment for monitoring the health of the robot with an alert generation setting as per the current operation status of the bot, amongst other security protocols and measures. Till date, multiple platforms are required to handle each of the aforementioned operations separately, although using the flow model discussed in the following section, the control and monitor interface is sufficient to handle these processes single-handedly, thereby providing an ease of access and eliminating the previous requirement. Furthermore, the user login details are saved and are accessible via the FRED server which extracts the user details and maintains a log, hence bringing a higher sense of security with the bot.

2. Methodology and Design

There are multiple subsystems that are integrated together to build the Simulink model for the 4-legged robot in MATLAB. Each of these systems are crucial for the final design to function and give the required graphs successfully.

2.1 Power Supply Subsystem

One of the major subsystems that includes the battery and load conditions. The plot current code needs to be formulated for the complete functioning of this given subsystem. This also includes further breakdowns of battery and load current sub-sub-systems that include current sensors and PMCs. The values from the current and voltage sensor are multiplied further to give the value for power consumed and the graph is visualized via a scope as shown in Figure 1.

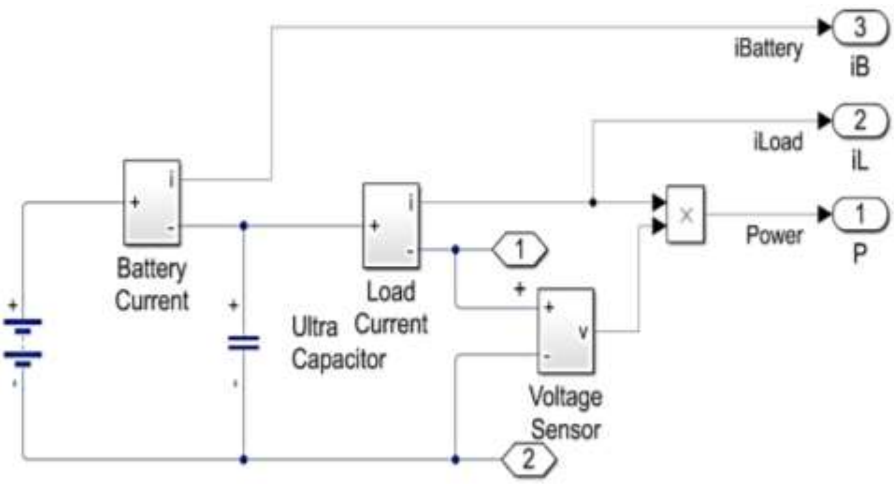

Figure 1. Power Supply

2.2 Gait Phase Design

The Gait Phase design is one of the most important subsystems in the Simulink model. This subsystem is responsible to ensure the motion of the limbs and the mainframe body of the robot with time. When the simulation is run, it is the gait phase design that handles the movement of each joint and link in the robot that overall determine its motion. The movement is efficiently timed in a practical manner to ensure smooth movement of the robot in the given direction. It has multiple phases determining the motion of the robot, subdivided into 8 phases of movement in time. This determines how each limb pushes the ground and folds before reconnecting to the floor in the exact same order or fashion as that of a 4-legged animal. Figure 2 displays the same.

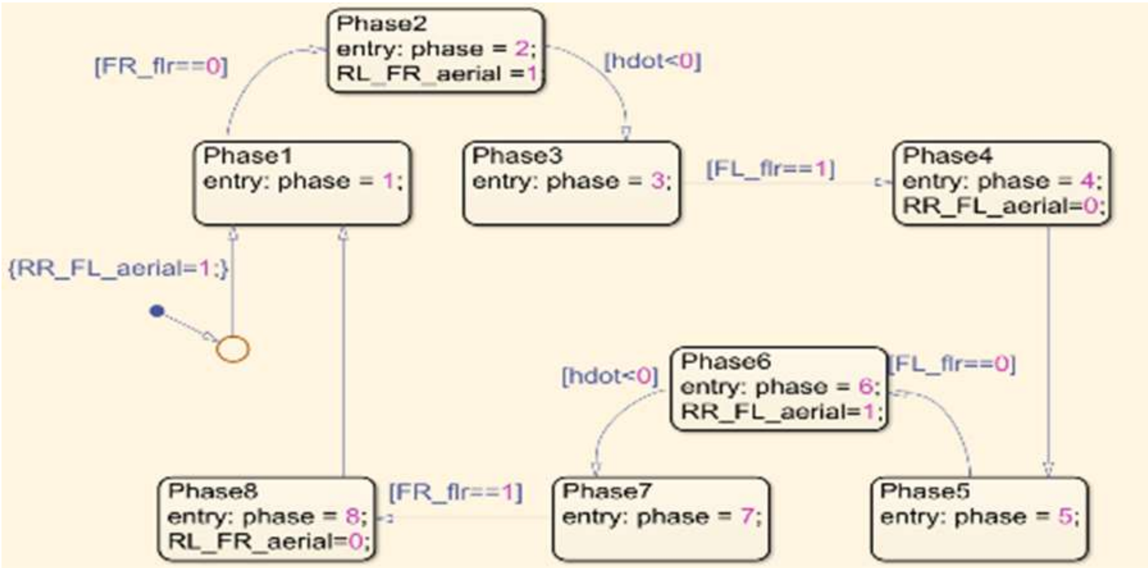

Figure 2. Gait Phase

2.3 Floor and Mesh Design

This includes forming the frame and the platform upon which the running simulation shall be projected. Includes components from the World Frame package in MATLAB Simulink that allows us to do the task mentioned above. Includes a PMC port to attach the subsystem with the final design, with a floor and mesh frame. The floor is a solid plane with the following properties and parameter as shown for the robot to run smoothly as visualized in Figure 3.

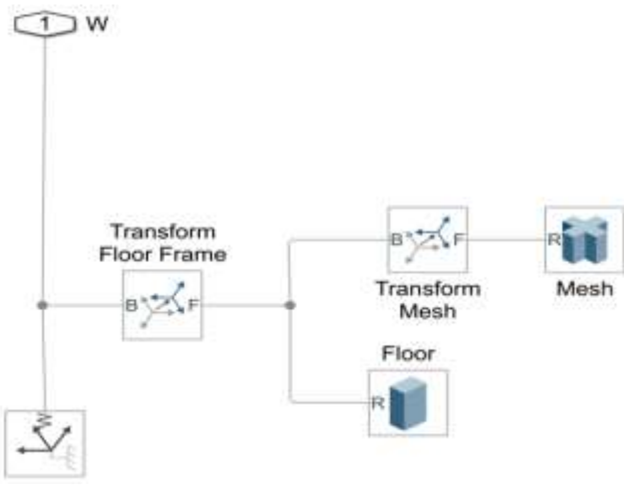

Figure 3. Floor and Mesh Frame

2.4 Limb Design

Another major subsystem that is based on manual calculations for selection of parameters for the design of the framework of the robot. This subsystem includes the main body, all 4 legs (rear and front, each for Left and Right), the tail and the nose of the robot. Combined further with the gait subsystem, the complete simulation will start running. The limb parameters are to be integrated via a MATLAB script file in this subsystem, with the gait phase as well. This gait phase and limb design integration finalizes the movement of the animated file that is visualized upon executing the simulation. Figure 4 displays the Limb Design.

All the subsystems mentioned are further integrated individually and then the final Simulink model is developed which is displayed in Figure 5. This mechatronic running robot model allows users to run simulations which help in the real-time design of the prototype. This prototype can be automated by utilizing IoT fundamentals and hence the control and monitor systems are developed.

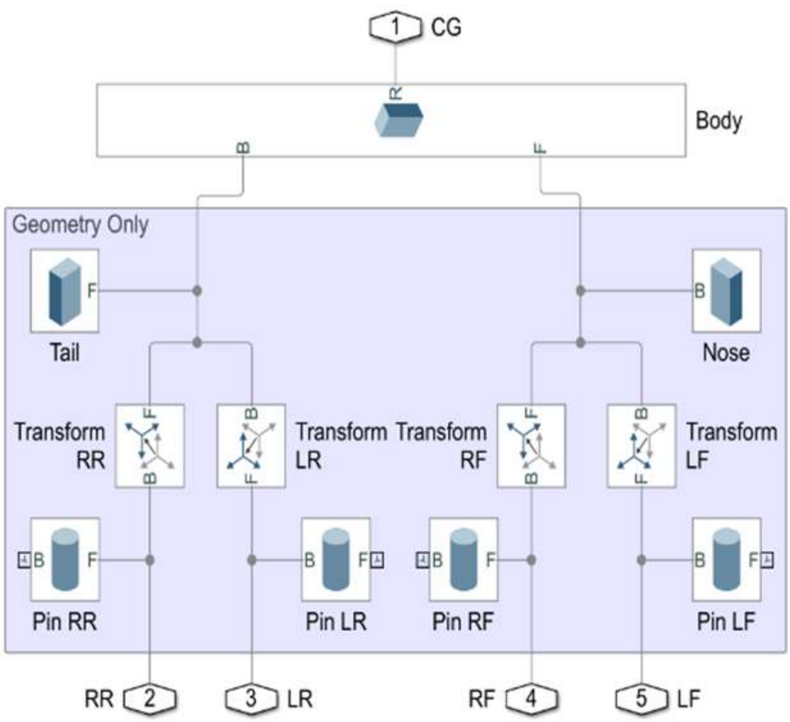

Figure 4. Limb Design

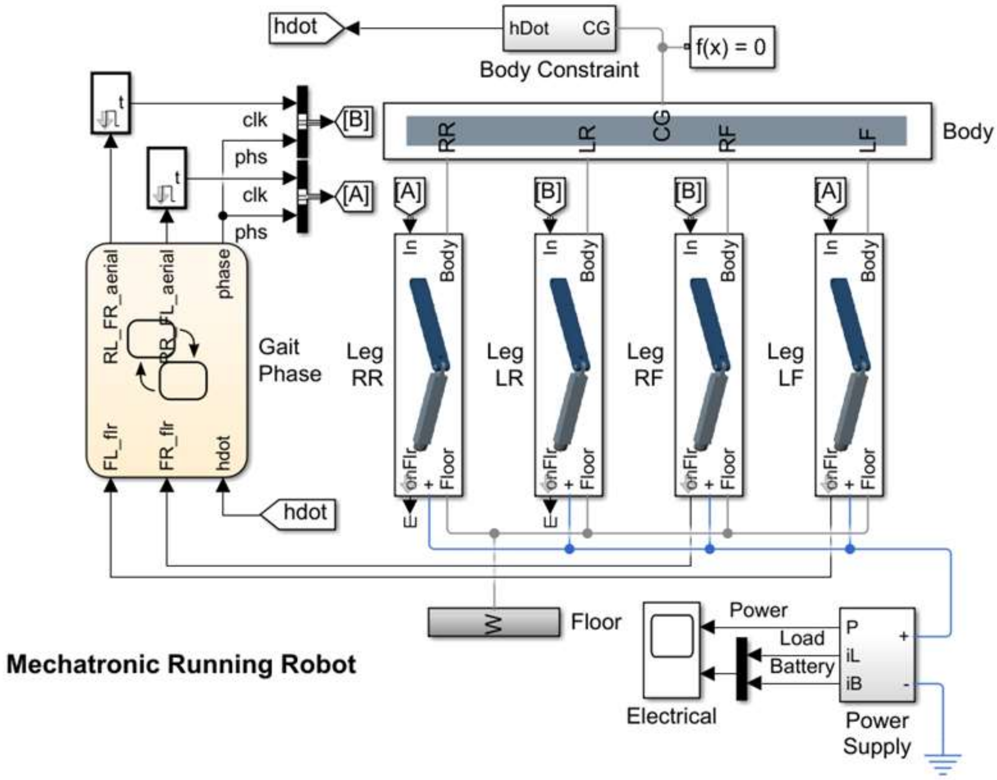

Figure 5. Mechatronic Quadruped Robot Simulink Design

Next, there is the automation segment that is based on IoT fundamentals and is designed using Node-Red and FRED. This allows a user to register himself/herself on a Smart User Interface that allows overall control and monitoring of the robot on a single platform. Upon real-time prototype integration, this will be highly efficient and at the same time simple to operate. Nodes implemented are mentioned in Table 1.

Table 1. Nodes Implemented

Serial Number	Node
1	Inject
2	Debug
3	HTTP in
4	HTTP Response
5	Function
6	Template
7	Comment
8	Webcam
9	Text
10	Chart
11	Button
12	Form
13	Notification
14	Gauge

Hence, the control and communication systems are built using these nodes to create flows. The control and monitoring set-up has the following set of functionalities that overall form the User Interface Dashboard for a user upon registration.

2.5 BOT Eye:

BOT eye feature requires access of the front facing camera. This is used to maintain a surveillance using the BOT eye. For the purpose of simulation, the Webcam node is used for this application. The node requires the user to allow access to his/her webcam on the UI and when incorporated with the quadruped, obstacles and the path ahead will be visible directly on the device used to access the UI.

2.6 Operational Voice Control:

This allows verbal commands given by the user to the BOT for conducting various tasks. With voice-control operation, maneuvering the bot remotely is a lot easier. The user needs to press a button to activate the function, and commands spoken are directly analyzed and transmitted to the bot as instructions.

2.7 Check for damage on BOT:

This feature allows to check the health status of the BOT, which is of prime importance to avoid malfunctioning of the BOT. This will provide the status feed of the BOT and accordingly action can be taken by the user.

2.8 Report Damage Form:

This form is provided to get it filled in case when BOT is under damage. Every BOT is provided with unique ID, this ID will be filled along with various other details, like date, phone number, mail ID and also the damage type in description. This form on submission will give the malfunctioning report to the concerned authority and thus action can be taken soon.

2.9 Speed Check:

Speed Check feature will help in maintaining a speed of the robot. This feature is useful to avoid over speeding as that might result in BOT going out of control. With the assistance of the Bot eye, this allows the user to ensure a particular speed is maintained since the terrains and environment might be varying at different points in time.

2.10 Battery Percentage:

BOTs are electrically charged. Being driven by an energy source, it becomes at most important to know how long the battery will last. Using this feature, the battery percentage can be checked any time and thus the BOT can be charged accordingly.

2.11 Compass Direction:

This feature is used to get the direction of the BOT. The BOT can be easily located using the compass direction. This makes use of the direction sensor which is incorporated inside the BOT.

Furthermore, the integrity of the bot is also well maintained due to this secure registration-based process that requires each user to log into the UI every time he/she wishes to operate the bot. For communication system, a Registration Form is created on Node-Red. For this, code is written in HTML and CSS. This page is the login page and is basically the authentication before giving access to the user. Once the details are filled on the registration page, the user will be directed to the User Interface which will give access to the user. These details filled by the user are on the client-side server. Now, using JSON code, the phone number of the user, which is unique for all, along with their name will be sent on the Cloud (Server End-FRED).

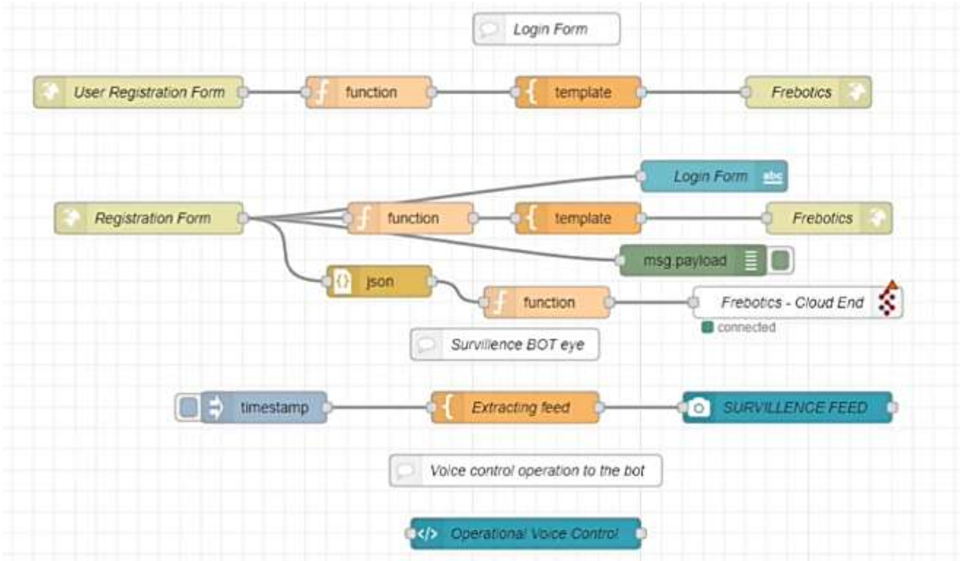

Figure 6. Login, Bot-Eye, and Voice Control flow

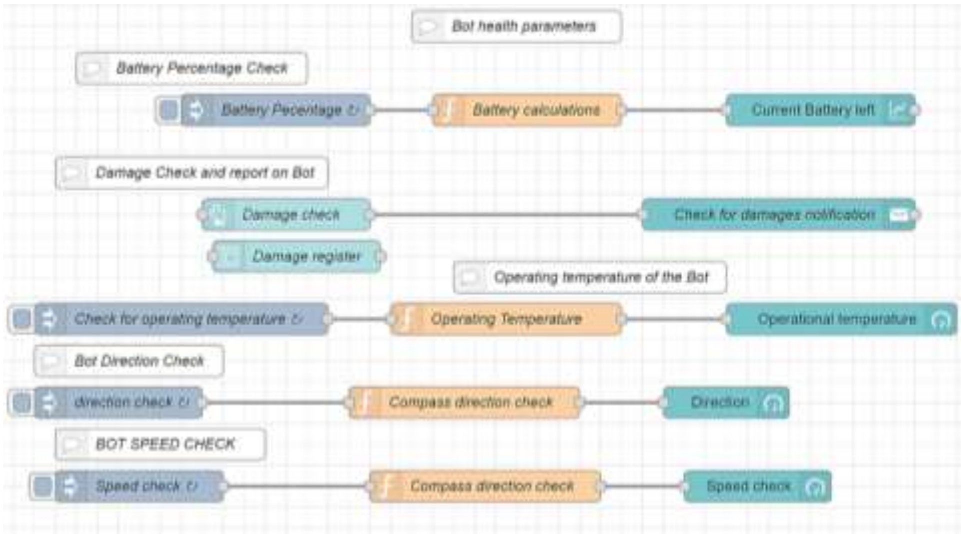

Figure 7. Bot health, operating temperature, battery, damage, direction and speed check flow

3. Results and Discussion

Upon execution in MATLAB, the Simulink model undergoes a complete simulation in integration with constraint codes and the overall design, thereby formulating graphs that allow the developers to attain and study necessary details before moving on to building the prototype. Scopes used in the Simulink model let the user access these graphs and further, SSC explore visualizes intricate graphs from different segments, including each of the limbs with their respective actuators, power supply, body constraints, etc. Besides these graphical outputs, the file after execution displays a 3-dimensional model of the robot that visualizes its design and its respective motion in an animated format. This display can be controlled by looping the traversed path and the speed can be altered as well. Output is shown in Figure 8.

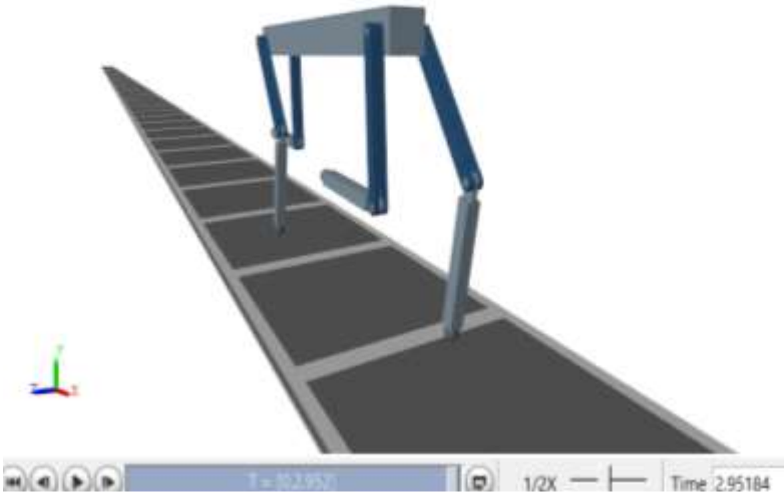

Figure 8. Quadruped Robot Simulation

Figure 9 displays the electrical reference scope for the design. Being an ideal simulation model, the input current and the relation with power is directly proportional in this simulation, hence giving symmetric plots for power consumption and currents. It can be therefore observed that both the lines for load are hence in sync, that is the black plot. Load current is taken as the input while the power generated depends completely on the former plot. The faintred line portrays the battery plot, which over the entire simulation time remains quite constant and does not cross a certain threshold throughout the whole range of motion.

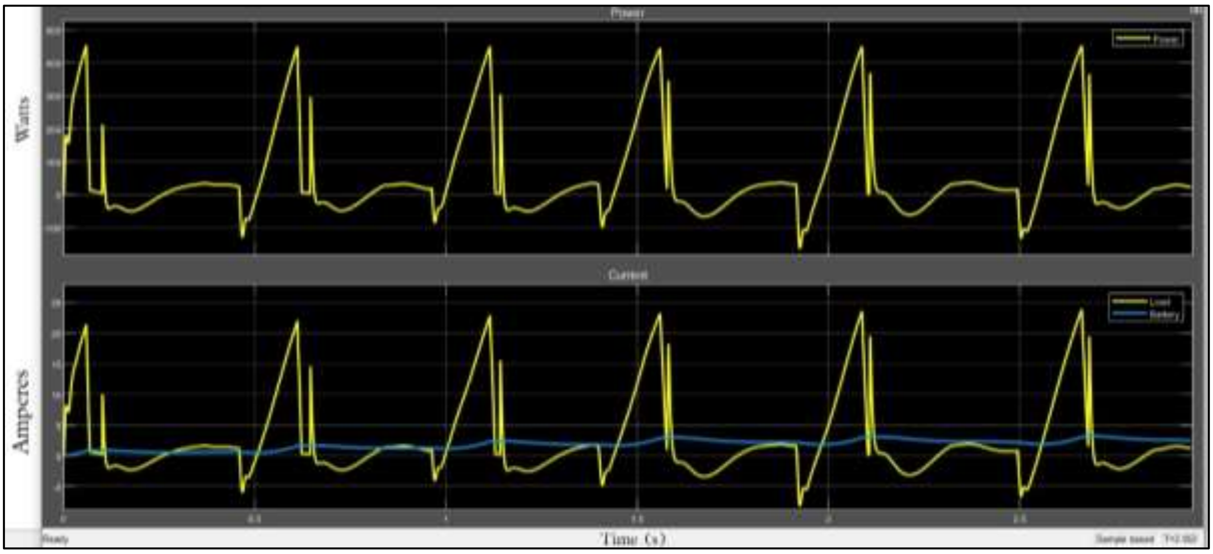

Figure 9. Electrical Reference Scope

Furthermore, SSC explore as mentioned can visualize intricate graphs. Figure 10 shows an example of the Torque verses Time graph for the Left Front limb. Here, since the simulation is ideal and running in a controlled, symmetric environment, no irrationalities are observed in the plot which fairly remains constant with the limbs kicking the ground to push the whole body forward.

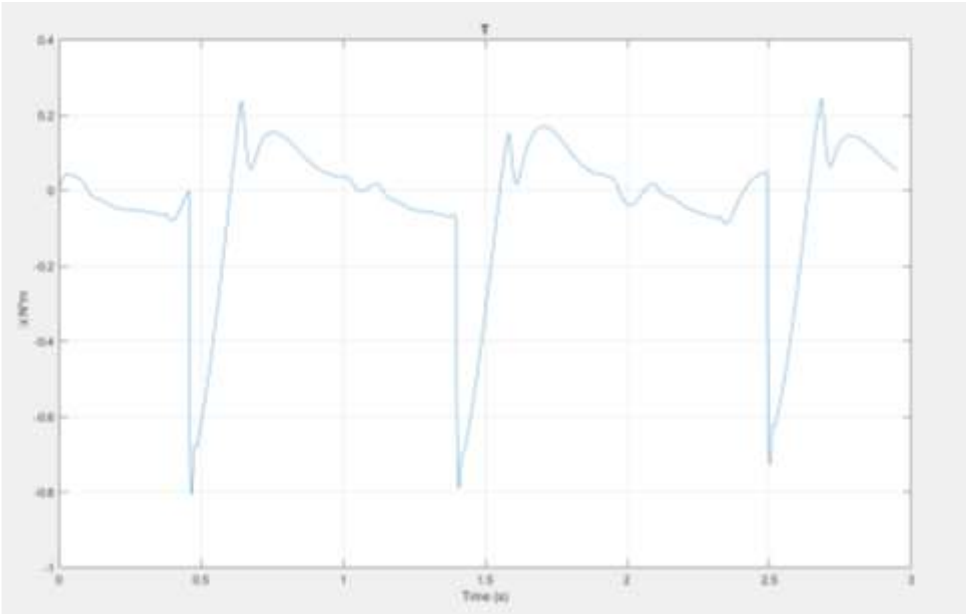

Figure 10. Torque v/s Time using SSC Explore

In the automation segment, the Node-Red and FRED simulations are executed while being in sync that allows the user to access the UI after logging in via the login form page. The registration form is shown in Figure 11 that takes in the user details and further displays the name and mobile number on the FRED-Cloud Service through JavaScript in the debug window. All details upon logging in are stored in the Node-Red debug window as well.

In the Dashboard UI in Figure 12, the user can access the features of the bot and hence control and monitor it in a facile manner as discussed in the previous sections. The figure displays the webcam output and the voice-recognition system identifying the instruction. Values generated for the rest of the gauges and the graphical output for current is randomly generated for the purpose of the trial. Damages can be checked by clicking on the option for "Check for Damages if Any" and if any contingencies are found, the same can be reported using the form below. This is reflected on the remote server as well.

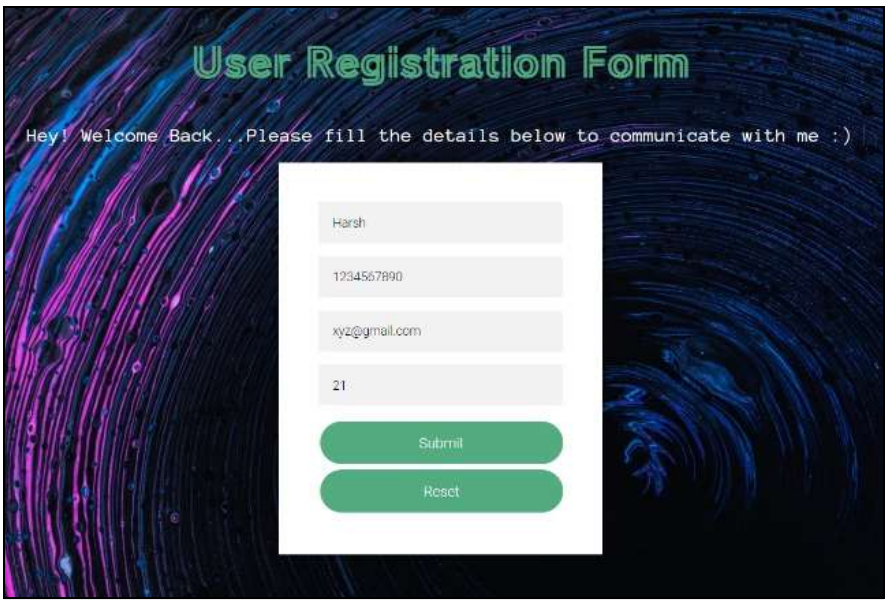

Figure 11. User Registration Form

Figure 12. Control and Monitor User Interface

4. Conclusion

The design and execution for a 4-legged autonomous Robot with Voice control upon integration was developed and simulation results were observed. This shows that a voice control embedded system would help in establishing a better control and command over the quadruped-bot and hence help in enhancing the output of the application from it. Benefits include faster data input operation, reduced number of operative errors thereby saving time, and also improvement of human inter-operability. The speed is variable and also can be controlled via the voice control system, ranging from 0.2 m/s to a top speed of 1.5 m/s, averaging at 1 m/s. With new and efficient robots being manufactured for wide ranging purposes every day, this paper brings in a system that allows the user to operate the robot easily with voice administration, and also lets him/her to gain access to the digital environment that gives crucial intel about the robot's functioning and operations, all on a single smart user interface. Power consumption in the simulation shows it peaks at about 450 W and current flow tops off 22.5 Amperes during the whole cycle. The applications of the design developed ranges from recording and collecting data from uninhabitable zones, to construction processes, homely chores and defense-based projects for the military.

New advancements in the field of robotics and automation are groundbreaking, with Google developing worker-robots which are intended to have their separate unique personalities, downloaded from a cloud-based system. Universal Robots have created the UR3, which is capable to build its own replacement parts and can carry out functions such as gluing, soldering, etc. Pepper on the other hand, is a humanoid developed for therapeutic purposes. It adapts to the mood of the humans around it by perceiving their emotions from their speech and behavior, and responds accordingly. In conclusion, the quadruped designed and discussed in this paper has its own set of conventional pros that are semantic to 4-legged robots, for example stability and flexibility. The design in its ideal form gives graphical evidence that accounts for minimal loss of power during any operations. Furthermore, the user interface created using Node-Red flows allows universal and complete control of the bot from a single platform, which is a rare scenario in the robots mentioned previously in this section, giving rise to ease of control and monitor. Being based on IoT, the machine-to-machine procedures and communication protocols are made simple and quick, leading to less delay in the instruction and response period. The user interface behind the modern robots discussed is based on each of the primary constituents of Industry 4.0, hence IoT is a substantial tool for automation, and hence it is evident that the future of robotics and automation depends largely on it.

References

[1] Study the Leg Mechanism of a Four-Legged Field Robot, Mohammad Harun-Or-Rashid, Mostafijur Rahman, Sabrina Rashid, Ahsanullah University of Science and Technology, Dhaka, Bangladesh
[2] A Study on Locomotion of Quadruped Robot, T.T.K DAT, PhucThien Tran, Faculty of Mechanical Engineering, Ho Chi Minh City University of Technology, Vietnam
[3] A four-legged walking robot with obstacle overcoming capabilities, Antonio Bento Filho, Gianluca Miglio, Paulo Faria S Amaral, Human System Interactions (HSI), 2010 3rd Conference.
[4] Mechanical Design of a Quadruped Robot for Horizontal Ground to Vertical Wall Movement, Abd Alsalam Sh. I. Alsalameh, Shamsudin H.M. Amin, RosbiMamat, UniversitiTeknologi Malaysia
[5] Cai, Shiyao; Ma, Zhiliang; Skibniewski, Miroslaw; Guo, Jianfeng; Yun, Langsheng. ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction; Waterloo Vol. 35: 1-8. Waterloo: IAARC Publications. (2018)
[6] Qingsong Ai, Wei Meng, FaycalBensaali, XiaojunZhai, Lu Liu, Nasser Alaraje, Editorial for FGCS special issue: Intelligent IoT systems for healthcare and rehabilitation, Future Generation Computer Systems, Volume 125, 2021, Pages 770-773, ISSN 0167-739X, https://doi.org/10.1016/j.future.2021.07.029.
[7] R. S. Batth, A. Nayyar and A. Nagpal, "Internet of Robotic Things: Driving Intelligent Robotics of Future - Concept, Architecture, Applications and Technologies," 2018 4th International Conference on Computing Sciences (ICCS), 2018, pp. 151-160, doi: 10.1109/ICCS.2018.00033.
[8] S. A. Rahat, A. Imteaj and T. Rahman, "An IoT based Interactive Speech Recognizable Robot with Distance control using Raspberry Pi," 2018 International Conference on Innovations in Science, Engineering and Technology (ICISET), 2018, pp. 480-485, doi: 10.1109/ICISET.2018.8745656.
[9] Nayyar A., Puri V., Nguyen N.G., Le D.N. (2018) Smart Surveillance Robot for Real-Time Monitoring and Control System in Environment and Industrial Applications. In: Bhateja V., Nguyen B., Nguyen N., Satapathy S., Le DN. (eds) Information Systems Design and Intelligent Applications. Advances in Intelligent Systems and Computing, vol 672. Springer, Singapore. https://doi.org/10.1007/978-981-10-7512-4_23
[10] Benotsmane R., Dudás L., Kovács G. (2021) Survey on New Trends of Robotic Tools in the Automotive Industry. In: Jármai K., Voith K. (eds) Vehicle and Automotive Engineering 3. VAE 2020. Lecture Notes in Mechanical Engineering. Springer, Singapore. https://doi.org/10.1007/978-981-15-9529-5_38
[11] M. Neunert et al., "Whole-Body Nonlinear Model Predictive Control Through Contacts for Quadrupeds," in IEEE Robotics and Automation Letters, vol. 3, no. 3, pp. 1458-1465, July 2018, doi: 10.1109/LRA.2018.2800124.
[12] M. Chignoli and P. M. Wensing, "Variational-Based Optimal Control of Underactuated Balancing for Dynamic Quadrupeds," in IEEE Access, vol. 8, pp. 49785-49797, 2020, doi: 10.1109/ACCESS.2020.2980446.
[13] A. M. Al-Busaidi, "Development of an educational environment for online control of a biped robot using MATLAB and Arduino," 2012 9th France-Japan & 7th Europe-Asia Congress on Mechatronics (MECATRONICS) / 13th Int'l Workshop on Research and Education in Mechatronics (REM), 2012, pp. 337-344, doi: 10.1109/MECATRONICS.2012.6451030.
[14] Beaber, SI, Zaghloul, AS, Kamel, MA, & Hussein, WM. "Dynamic Modeling and Control of the Hexapod Robot Using Matlab SimMechanics." Proceedings of the ASME 2018 International Mechanical Engineering Congress and Exposition. Volume 4A: Dynamics, Vibration, and Control. Pittsburgh, Pennsylvania, USA. November 9–15, 2018. V04AT06A036. ASME.
[15] Mart Tamre, Robert Hudjakov, Dmitry Shvarts, Ahti Polder, MaidoHiiemaa, Mart Juurma (2018) Implementation of Integrated Wireless Network and MATLAB System to Control Autonomous Mobile Robot, Vol. 1, No. 1 (2018): International Journal of Innovative Technology and Interdisciplinary Sciences - Volume 1, Issue 1