IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 April 2025, accepted 12 April 2025, date of publication 22 April 2025, date of current version 2 May 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3563235

APPLIED RESEARCH

V-Model Based Integration of Mechatronics and Power Electronics for a Solar-Powered Wire-Traversing Robot

EFRAIN MENDEZ-FLORES[1], (Member, IEEE), DIANA MORALES[2], AND MAGNUS EGERSTEDT[2], (Fellow, IEEE)

[1]Department of Civil and Environmental Engineering, University of California, Irvine, CA 92697, USA
[2]Department of Electrical Engineering and Computer Science, University of California, Irvine, CA 92697, USA

Corresponding author: Efrain Mendez-Flores (efrainmf@uci.edu)

This work was supported by the Crystal Cove Conservancy.

ABSTRACT This work focuses on the design of a solar-powered wire-traversing robot for environmental monitoring in remote areas, where solar power harvesting ensures continuous operation without frequent external charging. This paper presents an integrated power electronics design, emphasizing system-level considerations for efficient energy management and enhanced battery life, extending operational time and reliability. Experimental results showed the RaccoonBot could dynamically locate maximum solar exposure with 0.134[mm] resolution, maintain up to 7.5[W] of charging power, and consume just 0.025[A] in standby. These features, combined with the bio-inspired solar-tracking features and fail-safe design, enable over a week of autonomous operation, extending runtime from 5 hours to weeks. The findings validate the V-model's effectiveness in creating efficient, reliable systems, demonstrated by the RaccoonBot for environmental monitoring.

INDEX TERMS Robotics, power electronics, bio-inspired, solar panels, system design, V-model, persistent environmental monitoring, mechatronic systems, wire-traversing, solar-tracking systems.

I. INTRODUCTION

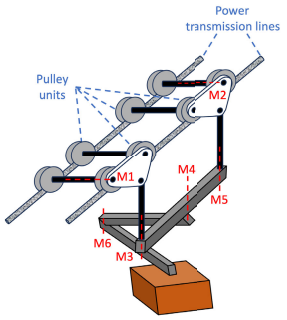
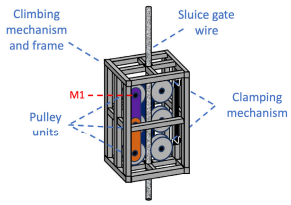
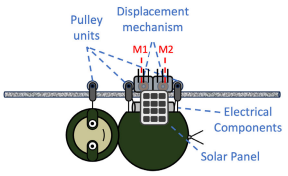
Renewable Energy Sources (RES) have a critical role to play in advancing environmental sustainability, economic growth, technological innovation, and worldwide energy accessibility. Among them, Photovoltaic (PV) systems achieved a record annual increase of 270 [TWh] in 2023 compared to the power supplied in 2022, which according to the International Energy Agency (IEA) [1], represents around 1,300 [TWh] of the worldwide energy supply, which emphasizes the importance of solar systems for endless applications.

Moreover, [2] showed how the growing interest in PV systems has led to major improvements in manufacturing technologies of solar panels and a reduction in implementation costs, while [3] illustrates how that development has enabled the integration of PV sources with advanced robotics, which has ushered a new era of more sustainable and efficient solutions; including, solar-powered UAV's [4], smart farming

FIGURE 1. RaccoonBot deployed along a wire between two trees.

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad Alshabi.

TABLE 1. Summary of State-of-the-art Wire-traversing robots divided by application.

Wire-Traversing Robot Type	Description	Frame example
Power transmission lines inspection robots	In the case of the inspection of power transmission lines, the robot addressed in [9] has served as a cornerstone for different wire-traversing robot styles; such as, [10]–[12]. These robots have the main goal of providing reliable platforms for inspection and maintenance of power transmission lines, where [13] validates that some of their common features are: fail-safe mechanical designs to overcome unexpected windy conditions, mechanisms capable of obstacle crossing (tension clamps, vibration dampers, suspension clams, insulators, etc. [14]), and autonomous or remote-controlled navigation systems. This is why, these robots usually need multiple actuators, to provide robustness to the navigation along the wires even when the robot is designed for a single power transmission line (e.g., [15]–[17]).	
Wire climbing robots	Unlike live line inspection robots, wire climbing robots are not designed to travel over the wire, these robots are mostly intended to climb wires vertically as explored in [18]. Common applications for wire-climbing robots are: inspection and maintenance of hanger cables in suspension bridges (e.g., [19]–[21]), and the regular wire maintenance in sluice gates ([18], [22]). These designs usually have a clamping mechanism, allowing the robot to be locked to the wire such that the actuators enable it to climb or descend, as explained in [18]. The example in this section is inspired by the design from [18], which uses a set of springs as a clamping mechanism to regulate the pressure to keep the robot in the wire.	
Environmental monitoring robots	For environmental monitoring applications robots need to be able to gather data around an area of interest; and when persistent long-lasting deployments are required, robots should also have low energy-consumption to ensure longer samples [23]. Thus, [23], [24] introduce two different robots that use wires as tethers together with propellers to enable the ability to move in a conical 3D space among tree canopies. Even though those robots are good examples of how tether robots improve environmental monitoring, when long-time deployments are required (for weeks or even months of operation) wire-traversing robots need to be integrated with solar panels to achieve persistence over long periods of time, as validated by the SlothBot introduced in [25]. The example in this section is inspired by the SlothBot from [25], which is an energy-efficient solar-powered robot, with two motors above the wire for the displacement.	

robots [5] & [6], NASA lunar exploration rovers [7], and environmental monitoring robots [8]. This paper is focused in the RaccoonBot introduced in [8] (showed in Fig. 1), which is a novel environmental monitoring robot.

Among these innovations, solar-powered wire-traversing robots have enabled the inspection, maintenance, and monitoring capabilities in hard-to-reach places and under complicated weather scenarios [26]. Deepening into wire-traversing robots, applications that have lead to the development of this category are summarized in Table 1. In all the example figures from Table 1, M stands for all the mechanical actuators used for the given example.

As implied from the examples from Table 1, the application usually leads to the features to be developed from the perspective of displacement mechanisms, electrical design, and in some cases the integration with RES. Additionally, [27] highlights that the environmental data to be acquired or monitored, has critical relevance in the choice of features for an effective locomotion and electrical design. Therefore, this paper focuses on the RaccoonBot (introduced in [8]), a wire-traversing robot for environmental monitoring in remote or difficult-to-access locations; however, as this robotic platform has the objective of achieving long-term environmental monitoring, harvesting solar power is crucial to ensure continuous operations without the need for frequent external charging (as analyzed for mobile robotics in [3]).

A similarity between the RaccoonBot and the SlothBot [28], is that these types of robots require a pre-mounted wired structure, limiting the application spectrum to scenarios where the sampled area is well-defined without room for samples in random locations [29]. Still, those features also enable reliable environmental monitoring tools with reduced energy consumption, since most of the operations are constant during the long-term operations.

The solar-tracking and wire-traversing features of the RaccoonBot were validated in [8], nevertheless the electromechanical integration to enable the robot's deployment was not fully addressed, since the main intention of [8] was to validate the mechanical features of the robot through multiple tests designed to evaluate its capabilities for persistent environmental monitoring.

Unlike [8], this work presents all the design-level considerations to bridge the gap between design, and implementation stages for environmental robots deployments. Additionally, this work also evolves the solar-tracking algorithm introduced in [8], where this second version of the algorithm also considers the implementation of a Real-Time Clock (RTC) for longer deployments to provide a reliable re-initialization signal, which enables the robot to take better decisions, wether to enter deep-sleep mode or to wake-up and continue its solar-tracking endeavor.

When considering solar powered wire-traversing robots, or even proper PV integration to multiple applications, there is a research gap between the control circuits, systems, and techniques dedicated to maximizing the electrical power produced by a PV source [30]. Consequently, this work emphasizes the system-level considerations crucial for efficient energy management and enhanced battery life, to extend the robot's operational time and reliability based on the V-model methodology applied for the electrical system design.

To develop a wire-traversing robot capable of providing long-term solar-powered environmental monitoring, the V-model is particularly well-suited due to its structured and highly iterative nature (as analyzed in [31]). When compared to other development methodologies, such as Agile [32] or Waterfall [33], the V-model offers distinct advantages that align with the specific needs to iteratively develop the electrical system of the robot, especially in terms of reliability, system integration, and long-term performance.

The V-model is built around a systematic breakdown of the development process, which makes it ideal for complex systems that require high levels of coordination between subsystems (as validated in [34]). Particularly for the RaccoonBot, the development must integrate solar energy harvesting, displacement mechanisms, and data collection systems, and this methodology ensures that every subsystem is carefully defined, designed, and tested before full integration.

When compared against more flexible methods like the Agile methodology, where system components may evolve throughout development, the V-model provides a clear frame where design and validation happen in parallel. This structure, according to [35], is crucial for a project like RaccoonBot, since it minimizes the risk of unexpected issues during the integration phase, ensuring that the robot's solar-tracking, wire-traversing, and energy-aware systems work cohesively from the beginning.

On the other hand, when compared to the Waterfall methodology that also follows a sequential approach, the V-model has a more reliable approach since it incorporates *verification and validation* at each stage, which is one of the most desirable features for robot developers (as studied in [36]). The Waterfall methodology typically reserves the testing stages for the end of the project, while the V-model ensures that every phase, from the high-level system design to detailed system design, undergoes rigorous testing as the robot is developed [35], [36]. Continuous testing and validation have greater relevance for long-term deployments, where failures in the field can be difficult to address. For instance, ensuring that the energy management system can reliably harvest and store solar power under various environmental conditions is both a final step and an iterative refinement process, throughout the development process.

To summarize the compared features of those three methodologies from the perspective of the RaccoonBot design, Fig. 2 shows a qualitative comparison of the models (summarized from [35], [36], [37], [38], [39]), highlighting in green the most desirable features for the robot's development.

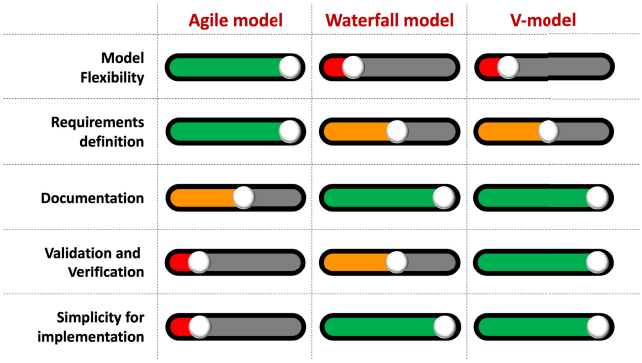


FIGURE 2. Development models compared from the perspective of what is needed for the RaccoonBot integration.

Additionally, [35] shows that the features of the Agile model make the methodology complex to implement, and constant verification can be difficult to achieve due to the agile dynamics of the model, which makes it suitable for applications that require flexibility and short-term developments. That is what it makes this approach less ideal for the RaccoonBot's development. Moreover, it is also true (as presented in [36], [39]) that the V-model is almost a modified version of the Waterfall model, which makes them share the fact that dealing with changes among stages can be complicated. Still, the V-model suits this application better because of the addition of constant validation and verification features, which simplifies the iterative process for the RaccoonBot's mechatronic development.

An example of how the features from the V-model make it a better approach for the RaccoonBot's development, is found under the knowledge that the robot's ability to harvest solar energy and manage the battery's charge cycle

directly impacts its capacity for travel along the wire and to fulfill its environmental monitoring tasks. These relationships can be anticipated, designed for, and tested in the early stages of development using the V-model, reducing the risk of late-stage failures that may be more difficult to fix in adaptive or incremental methodologies, as validated through [34].

This work contributes to the state-of-the-art by filling the gap between the design and the systems integration, through an adapted version of the V-model design methodology followed to bring to life a solar-tracking wire-traversing autonomous robot that features energy-aware mobility; which results in a reliable fail-safe robot capable of being deployed for persistent environmental monitoring for long periods of time (beyond a single battery charge).

Previous work on environmental monitoring robotics has explored various approaches to achieving energy-efficient and autonomous operation. Our prior contribution, RaccoonBot [8], introduced a solar-tracking mobility strategy, enabling the robot to dynamically adjust its position for optimal energy harvesting. However, [8] focused on the mechanical design and movement capabilities, validating the robot's wire-traversing ability and an efficient solar-tracking behavior. While this approach improved energy efficiency from a mechanical standpoint, it did not incorporate a comprehensive electrical system integration or an advanced energy management strategy, meaning the robot remained constantly active, consuming energy even when movement was unnecessary. In contrast, this work introduces a Torpor-inspired energy management approach, allowing the robot to dynamically enter a low-power sleep state when solar energy is insufficient, significantly extending its operational endurance.

Beyond robotics, our previous work has also contributed to advancements in power electronics, energy harvesting systems, and optimization techniques for electrical systems [40], [41], [42]. These studies focused on maximizing solar

TABLE 2. Symbols and acronyms nomenclature Table.

Subsystem	Symbol / Acronym	Definition	Units
Electrical system design	PV	Photovoltaic	-
	I	Output current of the solar panel.	[A]
	I_{ph}	Current generated by the solar cell (single-diode model).	[A]
	I_d	Current consumed through the diode (single-diode model).	[A]
	I_p	Current across the parallel parasitic resistor (single-diode model).	[A]
	I_0	Diode's reverse saturation current (single-diode model).	[A]
	I_{sc}	Short-circuit current (single-diode model).	[A]
	V	Output voltage of the solar panel.	[V]
	V_{oc}	Open-circuit voltage (single-diode model).	[V]
	G	Solar irradiation.	$[W/m^2]$
	T	Temperature.	[°C]
	T_c	Temperature of the solar cell (single-diode model).	[°C]
	R_s	Series parasitic resistance of the PV panel (single-diode model).	[Ω]
	R_p	Parallel parasitic resistance of the PV panel (single-diode model).	[Ω]
	μ_{sc}	Temperature coefficient at short circuit current (single-diode model).	[%/°C]
	ε_G	Material's band-gap energy (single-diode model).	[eV]
	STC	Standard Test Conditions (G = 1000 $[W/m^2]$, T = 25[°C]).	-
	N_s	Number of solar cells connected in series.	-
	A	Ideality factor of the diode (single-diode model).	-
	k	Boltzmann constant ($1.381 \cdot 10^{-23}$[J/K]) (single-diode model).	[J/K]
	q	Electron charge factor ($1.602 \cdot 10^{-19}$[C]) (single-diode model).	[C]
	V_T	Thermal voltage (single-diode model).	[V]
	MPPT	Maximum Power Point Tracking.	-
	RTC	Real-Time Clock.	-
	MCU	Microcontroller Unit.	-
Mechanical system design	τ	Torque induced by the motor.	$[N \cdot m]$
	N	Normal force acting on the robot's wheels.	[N]
	W	Weight of the robot.	[N]
	F_R	Friction force.	[N]
	F_M	Motor's linear resultant force.	[N]
	d	Distance traveled.	m
	d_{enc}	Distance traveled in encoder counts.	counts
	CPR_{enc}	Encoder counts per revolution.	counts
	CPWR	Counts per wheel revolution.	counts
	Rev_{req}	Required revolutions to travel d.	-
	PPR	Pulses per revolution.	pulses
Solar-tracking algorithm	Δx	Step size for solar tracking movement.	[m]
	I	Inertia coefficient for solar-tracking algorithm.	-
	ζ	Damping ratio coefficient in the algorithm.	-
	G_{Best}	Best global solar power position found by the robot.	[W]
	x_{Best}	Position on the wire corresponding to G_{Best}.	[m]
	x_0	Robot's Initial position.	[m]
	it_{sim}	Simulated solar tracking iterations.	-
	it_{exp}	Experimental solar tracking iterations.	-

power utilization and improving energy conversion efficiency in autonomous platforms. However, while they provided critical insights into solar energy optimization, they were not directly integrated with a robotic system for real-time adaptive energy management. This work bridges that gap by combining V-model-based mechatronic integration with an advanced Torpor-inspired energy management approach. By dynamically transitioning into a low-power sleep state, the robot mimics the biological behavior of raccoons, optimizing energy consumption and extending operational endurance beyond a single battery cycle.

By merging solar-tracking mobility, structured mechatronic integration, and efficient power electronics design, this research advances the field of solar-powered environmental monitoring robotics and establishes a foundation for the development of long-term autonomous systems in off-grid conditions. Table 2 presents a summary of the symbols and acronyms nomenclature used along this paper.

Building on this foundation, this paper further expands its contributions by applying a V-Model-based approach to systematically integrate mechatronic systems and power electronics, ensuring a structured and efficient development process. The proposed system, inspired by RaccoonBot [8], prioritizes energy efficiency and sustainability, key for long-term deployment in remote and off-grid locations. Additionally, this work introduces an innovative integration of power electronics for solar energy management, optimizing the robot's performance, endurance, awareness and adaptability in real-world autonomous monitoring tasks. These contributions highlight an interdisciplinary approach that enhances the reliability and effectiveness of environmental monitoring robotics.

Furthermore, this work advances the state-of-the-art in bio-inspired robotics through the development of a Torpor-inspired energy management algorithm, detailed in Section V-B. This algorithm is directly inspired by the natural behavior of raccoons, which enter a Torpor state to reduce their metabolism and body temperature in response to environmental conditions to conserve energy. Similarly, the proposed algorithm dynamically adjusts the robot's energy consumption based on environmental factors, ensuring sustainable, long-term operation. By leveraging this biologically inspired approach, the system achieves a balance between autonomous functionality and energy efficiency, making it well-suited for extended environmental monitoring applications.

This paper is organized as follows: Section II introduces the adapted V-model methodology to design the electrical system of the RaccoonBot. Meanwhile, Section III, presents an overview of the electro-mechanical features of the robot together with the tasks it is designed for. Section IV explores the model used to characterize and integrate the solar panel to the robot. Then, Section V introduces the solar tracking algorithm that enables the energy-aware mobility features in the robot. The results of the validation tests are discussed in Section VI, and the conclusions of this work in Section VII.

II. V-MODEL IN ELECTRICAL SYSTEMS ENGINEERING

The V-model for Systems Development was originally introduced in [43], where the main concepts of the original methodology was designed as a frame for Software Development process, which according to [44], has served as a reliable tool for the safe development, verification, validation, and introduction of different types of technical systems.

An application example of the V-model is addressed in [45], where it served as cornerstone for a software and hardware integration developed to enable a framework for model-driven software-systems; which, according to [45], can support the development of resilient cyber-physical systems in a Smart City context applied to limited traffic zones. Moreover, [44] expands the V-model to fulfill the requirements for the development of complex AI systems. Both applications converge at the importance of having a frame that allows the verification and validation of elements to be integrated to reduce risks during a system development; since, according to [46], the V-model can be divided into two set development stages: the project definition and the project integration, as presented in Fig. 3 and as addressed in [47].

Also from Fig. 3, it can be seen how the first stages of the V-model (left hand side of the V structure) are mostly related to the requirements definition, development planning, and documentation that lead to the project's implementation. On the other hand, the second group of stages (right hand side of the V structure) are where the integration occurs, since the V-model paves the way to a final product through the assemble and performance validation stages.

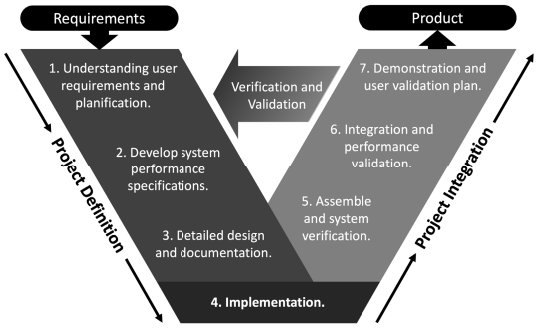


FIGURE 3. V-model for Systems Development structure.

To address the needs of the development process for a robotic platform some adjustments need to be made to the V-model structure in Fig. 3; as explored in [48], robotic systems need a development process that moves from a high-level system design into a detailed system design for its further implementation (as highlighted in [49], [50]). An example that enhances the suitability for robotic systems is addressed in [48], where the concept of a "robotic baby" is introduced and defined as the system-level requirements, through the V-model development process, to get a minimal viable prototype designed, educated and validated to be capable of learning and evolve.

Furthermore, [51] uses the V-model methodology to carry out a modular mechatronic infrastructure, particularly for robotic planetary exploration among a field operation scenario; which, under the Autonomous Robotic Networks to Help Modern Societies (ARCHES) project, validated the importance of the verification and validation process as a critical asset to ensure that requirements are aligned to the target's application in robotic applications. In a similar way, [50] shows how, from the perspective of a robotic platform designed for sanitizing purposes, the V-model enhances the way robotic platforms can be developed in a systematic path to reduce performance uncertainty in the final product.

Summarizing the concepts from the robotics development process from [47], [48], [49], [50], and [51], Fig.4 shows the V-model structure adapted to the RaccoonBot's development process, where the input of the development process are the required robot's solar-tracking, wire-traversing, and energy-aware features, and the expected output is the RaccoonBot itself. This V-model adaptation, also drags inspiration from the described V-model for electrical systems addressed by SIEMENS in [52] for industrial applications.

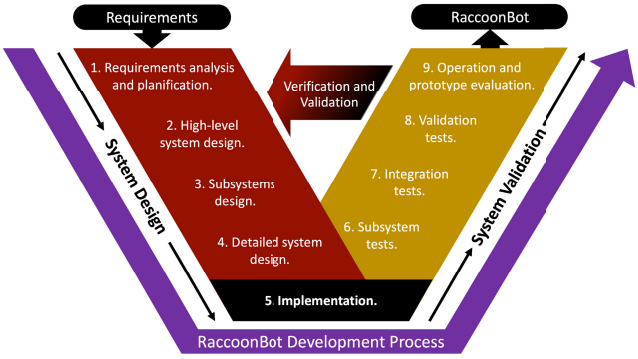


FIGURE 4. V-model adapted to the RaccoonBot development process.

The following section explains in greater detail how the V-model from Fig.4, is embraced for the development of each system required for the RaccoonBot's integration and further performance evaluation, as this is one of the contributions of this work. Additionally this work also analyzes, from a system-level, the requirements for a proper PV integration through the V-model methodology; since, as explained in [30], there has been a research gap between the control circuits, the systems that integrate the energy harvesting process, and techniques dedicated to maximize the electrical power produced by a PV source, which is crucial from the perspective of a robot that critically needs solar power to extend its operating time to achieve energetic persistency over time.

From the perspective of the adapted V-model described in Fig.4, Section III will introduce the RaccoonBot from the perspective of the *2. High-level system design*, *3. Subsystems design* and *Detailed system design* stages, meanwhile Section IV will dive into the subsystems integration to the PV source from the perspective of the *Subsystem tests* and *Integration tests* stages, and Section V highlights the energy-aware features of the robot through the *Validation tests* stage. Additionally, Section VI presents the *Operation and prototype evaluation* stage through the results presentation.

III. RaccoonBot

The RaccoonBot is a mechatronic system that needs to be designed as a reliable robotic platform for persistent environmental monitoring. As a result, the first step is to apply the adapted V-model to understand the requirements to feature a proper development. Among the general requirements, the RaccoonBot's design process has the following objectives:

- Create a robotic platform capable of acquiring environmental data even in hard-to-reach scenarios between trees (wire-traversing robot).
- The robot must withstand the weather conditions it may face (tough windy or rainy conditions endemic to the California application site).
- The Robotic platform must be bio-inspired to enhance environment preservation and to be minimally invasive for the environment in which it moves.
- For persistence over time, the robot needs to perform continuous operations without constant need for external charging, which is why it needs to be solar-powered.
- To increase the probability of the robot getting access to direct photovoltaic power, energy-aware mobility is a desired feature enabled by a solar-tracking algorithm.

With those requirements in mind, the V-model methodology addresses the need to move forward in the development process to a High-level system design, to start shaping the features to be implemented.

A. HIGH-LEVEL SYSTEM DESIGN

Fig.5 shows the main systems that constitute the high-level system design, which are the general features to fulfill application requirements.

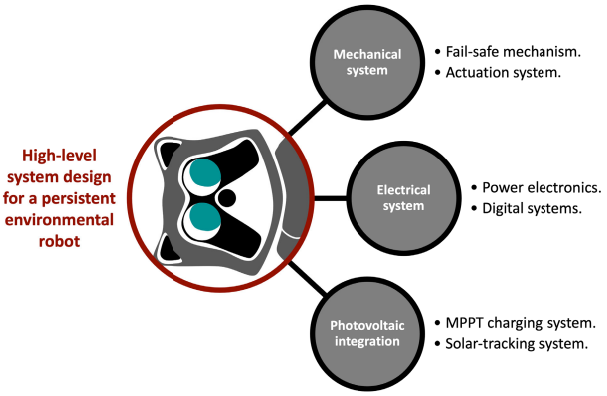


FIGURE 5. Main high-level features for the RaccoonBot.

The mechanical system addressed in Fig.5 is the part of the robot in charge of keeping it safe even against rough weather conditions, while still ensuring a smooth and reliable displacement through the actuation system.

In addition, the electrical system is designed to accommodate the power electronics required to drive the actuator and the power conversion required to regulate the battery's power into all the digital components, such as the microcontroller unit (MCU) and the sensors. Also from Fig.5, it can be seen that the photovoltaic integration stage is the part of the robot development process in charge of the energy harvested, which is needed to charge the battery and to enable the energy-aware mobility features, and to be able to position the robot where it can find direct access to solar power.

B. MECHANICAL SYSTEM DESIGN

The RaccoonBot's mechanical system is designed to meet specific technical requirements that ensure reliable operation in outdoor environments. The structure must support a total payload of up to 15 [kg](the RaccoonBot weights around 5 [kg]), which includes the power electronics components, actuation system, and battery pack, while maintaining a lightweight design to optimize energy efficiency. Additionally, the traction mechanism must provide sufficient friction and grip on the wire to prevent slippage and enable smooth wire-traversing under varying environmental conditions.

The system also requires a high-torque motor capable of delivering at least 0.15 [N·m] (RaccoonBot's motor can deliver up to 0.8 [N·m] before the 28:1 ratio gearbox) to overcome mechanical resistance during movement. Consequently, the mechanical frame (RaccoonBot's body) is designed to ensure durability and resistance to environmental factors; such as, temperature fluctuations and humidity, ensuring long-term deployment in remote locations. Finally, for the RaccoonBot's development, modularity and versatility are key considerations, to enable easy assembly, maintenance, and potential system upgrades.

Moving forward to the subsystem design stage and to a more detailed description of the mechanical stage, the designed mechanism satisfies the reliability requirements against wire perturbations to ensure a fail-safe design. Moreover, the mechanism enables the robot to safely remain on the wire even in the event of an electrical failure.

Consequently, Fig. 6 shows the mechanism in charge of the actuation system inside the RaccoonBot, where a single actuator is needed to provide the torque (τ) to enable the linear displacement of the robot. As additionally highlighted in Fig. 6, the designed structure for this application is a worm with a two gears mechanism, where the torque induced by the actuator is increased by the speed reduction caused by the gears in a 1:28 ratio.

The worm-gears mechanism designed for this application was selected for the mechanical advantages inherent in this type of gearbox; such as a large reduction ratio and non-backdrivability (as explored in [53]). This large reduction ratio allows the mechanism to better exploit the torque generated by the actuator, which enables a reliable platform that can travel smoothly along the wire even against different inclinations due to the wire tension.

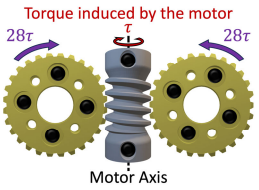


FIGURE 6. Worm with two gears mechanism designed as the actuation system.

As shown in [53], the non-backdrivability feature of worm-gear mechanisms means that they are able to safely hold on to a position in the wire without consuming additional energy. In other words, since the mechanism is non-reversible, the gears will only move due to torque applied by the actuator (through the motor axis in Fig. 6), but when there is non torque applied by the actuator, the mechanism self-locks and enables the robot to safely keep its position even in the event of a possible electrical failure (or shortage).

Both the non-backdrivability and the self-locking features are critical for persistent environmental monitoring applications, since the robot needs to remain safe at all time, even against unpredictable disturbances that the RaccoonBot may face due to rough weather conditions, objects colliding with the robot, and electrical failures among others. With those considerations in mind, Fig. 7 presents the free body representation of the mechanism inside the robot.

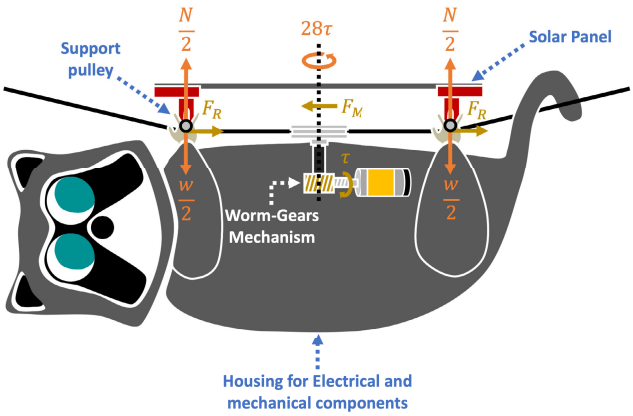


FIGURE 7. Free body representation of the mechanical features of the RaccoonBot.

As depicted in Fig. 7, the RaccoonBot's motion relies on the combination of support pulleys and the worm-gears mechanism between them, which is housed inside the RaccoonBot's body. Although the pulleys are responsible for all the Y-axis forces, as a safety measure the wheels in charge of the X-axis displacement are designed in such a way, that they can support the whole robot's weight in case of pulley failure. Addressing the main forces interacting among the free body diagram from Fig. 7, Table 3 shows the overview of the variables that represent them.

TABLE 3. Mechanical system variables for the RaccoonBot's mechanism.

Variable	Description
N	Normal force.
W	Robot's weight.
F_R	Friction force
τ	Torque induced by the Motor.
F_M	Motor's linear resultant force.

The free body representation shows how the pulleys arrangement enable force-decomposition, since the pulleys placed as the RaccoonBot's hands handle the forces in the Y-axis related to the weight, which creates a tense flat path with the wire that allows the actuation system to be only affected by the horizontal forces; which, in turn, improve the effects of the motor's torque. Thus, as addressed by Fig. 6, the conversion ratio of the torque (τ) is 28:1 due to the gearbox reduction.

To understand how the electrical design inside the RaccoonBot operates to drive the actuator inside the robot, the following subsection presents the details of the electronics subsystem.

C. ELECTRICAL SYSTEM DESIGN

For the development of the electrical system inside the RaccoonBot, the main requirements to address are the constant need for a reliable and efficient circuit suitable to achieve electrical autonomy, to be integrated with the photovoltaic subsystem required to enhance the battery's charge time. Consequently, to provide a clear track to understand the electrical system designed for the RaccoonBot, Table 4 presents the main electrical specifications of the design, meanwhile Fig. 8 introduces the blocks-diagram with the most relevant components of the electrical structure embedded inside the RaccoonBot.

TABLE 4. Electrical system technical specifications.

Parameter	Specification	Unit
Solar Panel Power at MPP	9	[W]
Solar Panel Voltage at MPP	17.34	[V]
Solar Panel Current at MPP	0.570	[V]
PCB Operating voltage rating	3.4 - 12	[V]
Battery Capacity	5000	[mAh]
Rated Battery Voltage	7.4	[V]
DC Motor Rated Voltage	12	[V]
DC Motor Rated Current	0.25	[A]
MCU Operating Voltage	5	[V]

In Fig. 8, the first part of the circuit is highlighted as the *Energy storage stage*, where the power from the PV system is regulated through a DC-DC converter featured as a Maximum Power Point Tracking (MPPT) battery charger. The MPPT circuit takes the unregulated voltage and current from the PV system, to maximize the energy harvested to charge the battery. The battery powers the electrical circuit inside the RaccoonBot; however, the power electronics that drive the motor, the sensors, and the digital components are connected to different power supplies.

We start with a step-by-step description of why multiple DC-DC converters are needed as Power Management Integrated Circuits (PMICs) for this application. The first PMIC is the DC-DC Buck Boost converter attached to the MCU, this circuit is in charge of regulating the power from the battery into a digital voltage to power the ATmega32U4. Which, when analyzing the application needs through the V-model requirements analysis stage, it is clear that maximizing the operating range in which the MCU can operate, is critical to ensure safe operations for longer periods of time; and according to [54], the way to go is using a buck-boost converter since it allows stepping down the battery voltage

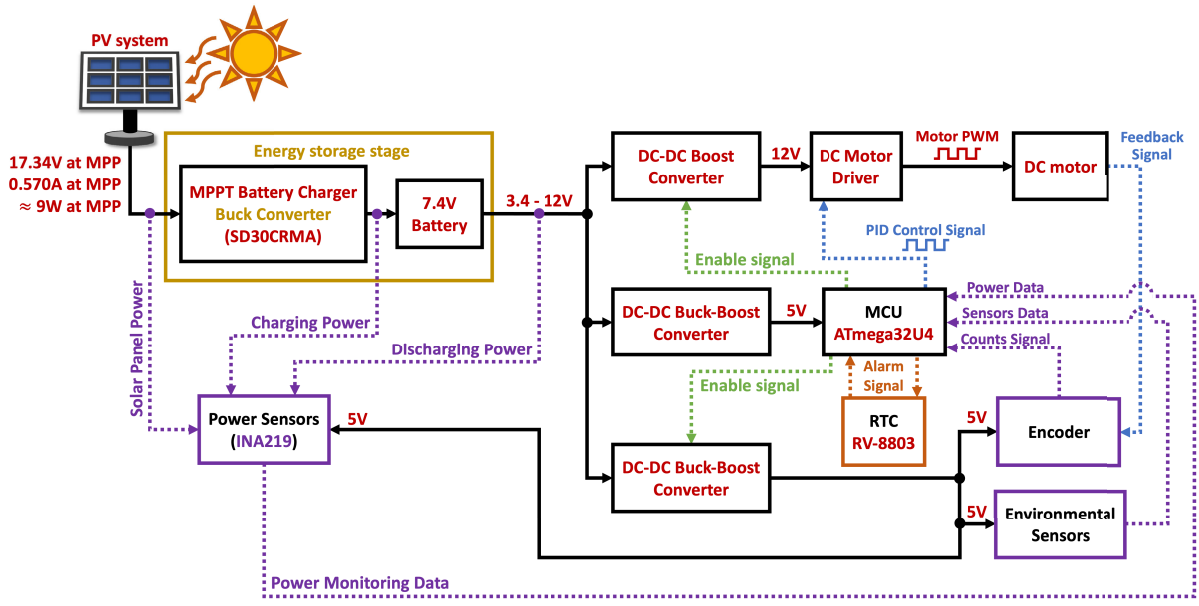


FIGURE 8. Electrical design embedded inside the RaccoonBot. The doted lines represent the data acquired from all the sensors in the circuit, and all the continuous lines represent all the electrical signals.

into a digital voltage when it is fully charged, but enables operating boosting the output voltage when the battery discharges bellow the 5[V] range, which extends the working time of the RaccoonBot.

This circuit design has two buck-boost converter and both have 5[V] as output. The main reason is the requirements developed at the beginning of the V-model, the robot needs to achieve persistency over time; to fulfill that requirement the robot is designed to enter into a ''sleep-mode'' when non-displacement is needed. When the RaccoonBot is in sleep-mode, to minimize the power consumption of the circuit, the MCU disables the power supplies from all the sensors and the power electronics stage from the motor driver.

That is why Fig. 8, shows two enable signals coming from the MCU to drive two power supplies in the circuit. The first one, to enable the PMIC that powers all the sensors and the digital components inside the RaccoonBot; meanwhile the other one, is designed to enable the boost converter that powers the DC motor driver stage. Moreover, the block described as RTC is a Real-Time-Clock module, which has an external CR2032 battery that powers the IC to ensure energetic autonomy for over a year.

The RTC module is adapted for this application so that, when the robot finishes its solar-tracking and sensing duty, it can configure an alarm so that the MCU can sleep until an external interrupt triggered by the RTC's alarm wakes-it-up. Fig. 9, summarizes the states that the robot operates in as related to the RTC module, where the middle dark blue states are the ones where the robot turns-off the ICs that it is not using and enters the MCU's sleep-mode.

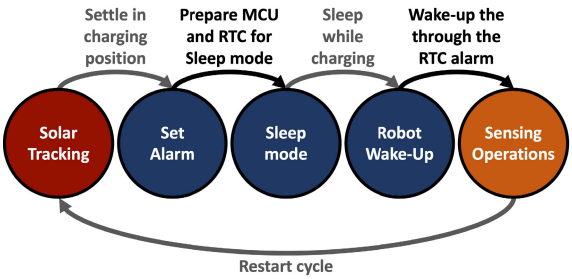


FIGURE 9. High-level states diagram related to the RTC operation inside the RaccoonBot.

This integration with the RTC module allows minimizing the overall power consumption, since the only components connected in active mode to the battery will be the ones required for the robot to wake-up, allowing a reduction of more than 75% to the power consumption when the robot is charging the battery or, when it is not performing environmental sensing activities and there is no solar power available.

For the PMICs in this application DC-DC converters are used, since according to [54] it is well known that switching power supplies are suitable for applications where efficiency is critical (such as persistent environmental monitoring). Therefore, Fig. 8 also shows how a boost converter is used to step-up the nominal 7.4[V] from the battery to the required 12[V] for the motor driver (H-bridge). All the selected converters for this electrical design report around 90% of efficiency, which enables minimizing energy losses due to parasitic elements.

Additionally, the PMICs arrangement addressed in Fig. 8 is the key design-wise difference when compared to the approach presented in [8]. Hence, Findings in [8], enabled validating that to push even further the RaccoonBot's operation time, using separated DC-DC converters provide a solution that allow the RaccoonBot to turn-off all the components that where not used at each stage.

This electrical circuit, together with the mechanical design from the previous subsection, build the mechatronic system required to perform the environmental monitoring tasks that the robot was designed for. Yet, to fulfill the persistency over time requirements of this application, the integration with RES is needed and explored with greater detail in the following section.

IV. PHOTOVOLTAIC INTEGRATION

When talking about photovoltaic systems, [30], [54] show how the proper integration between power converters, solar panels and optimization algorithms is needed to maximize the energy harvested through time. Hence, to understand the integration from a system design-level for the RaccoonBot, this section fills the gap of the development process carried out by the V-model adapted structure.

A. SUBSYSTEM MODELING

To understand the dynamic electrical features of the system, and to enable a proper solar panel selection for this application and a reliable solar-tracking algorithm, the mathematical model needs to be addressed to fulfill the V-model's *Detailed system design* phase.

As PV systems are capable of converting sunlight into electrical power, [30] explains that there are different approaches to understand how solar irradiation together with climatic conditions (Temperature) are translated as current, and subsequently voltage when the impedance at the solar panel terminals is modulated. As explored in [40] and validated through [30], the Single-Diode Model (SDM) enables simulating the PV system dynamic features under different solar irradiation and temperature scenarios. A key feature of this model, is that it allows understanding how the voltage and current profiles can be used as design features for a proper solar panel selection. Thus, the equivalent circuit of the SDM solar panel representation is presented in Fig. 10.

Even though the SDM represents a good balance between accuracy and simplicity (as detailed in [55]), this model finds its main limitations under low irradiance conditions for thin-film-based solar cells (as studied in [30]) or when modeling losses in the depletion region (as explored by [56]). Still, since this application requires modeling the dynamic features of the solar panel to be integrated around Standard

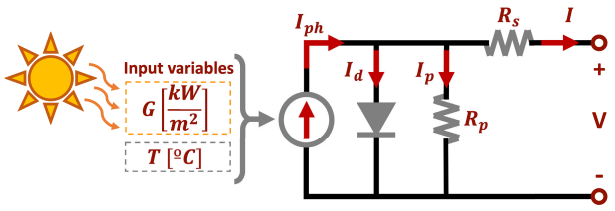


FIGURE 10. Electric diagram of the single-diode with ohmic losses equivalent model for PV systems.

Test Conditions (STC), [30] highlights that the SDM is effective to enable the I-V (Current-Voltage) and P-V (Power-Voltage) profiles characterization of the solar cell.

Additionally, from Fig. 10, it can be analyzed how solar irradiation (G) and temperature (T) make the solar panel work as a current source that supplies the output current I. Therefore, by applying Kirchhoff's Current Law (KCL) to the circuit, I can be defined as represented by Eq. (1).

$$I = I_{ph} - I_d - I_p, \tag{1}$$

where I_{ph} is the current generated by the solar cell, meanwhile I_d is the current consumed through the diode, and I_p the current along the parallel parasitic resistor.

Consequently, [40] shows that I_{ph} can be rewritten to consider the effect of the solar irradiation G and the PV panel's temperature T as input variables, which can be explained as shown by Eq. (2).

$$I_{ph} = \frac{G}{G_{STC}} \cdot [I_{ph_{STC}} + \mu_{sc}(T - T_{STC})], \tag{2}$$

where μ_{sc} is the temperature coefficient at short circuit current. Moreover, the repeated variables with the *STC* subindex refer to those variables but under Standard Test Conditions, which means: $G = 1000$ [W/m^2] and $T = 25$ [° C].

Going deeper into the diode's current I_d term from Eq. (1), Eq. (3) represents the mathematical expression considering the input variables (G and T).

$$I_d = I_0 \exp\left(\frac{V + I \cdot R_s}{a} - 1\right), \tag{3}$$

where I_0 is the diode's reverse saturation current, V is the voltage supplied by the solar panel across its output terminals, and R_s is the series parasitic resistance from Fig. 10.

Also from Eq. (3), a represents the term know as modified ideality factor, which [30] explains that serves the model as a change of variable term addressed by Eq. (4).

$$a = \frac{N_S \cdot A \cdot k \cdot T_c}{q} = N_S \cdot A \cdot V_T, \tag{4}$$

where V_T represents the thermal voltage, N_s is the number of solar cells connected in series inside the module, A is the ideality factor, k is the Boltzmann constant ($1.381 \cdot 10^{-23}$[J/K]), T_c is the cell's temperature, and q is the electron's charge factor ($1.602 \cdot 10^{-19}$[C]).

Going back to Eq. (3), the saturation current I_0 can be defined as shown by Eq. (5).

$$I_0 = \left[I_{sc_{STC}} \exp\left(\frac{-V_{oc}}{a}\right)\right]\left(\frac{T_c}{T_{c_{STC}}}\right)^3 \exp\left[\left(\frac{q\varepsilon_G}{AK}\right)\left(\frac{1}{T_{c_{STC}}} - \frac{1}{T_c}\right)\right], \tag{5}$$

where I_{sc} is the short-circuit current, V_{oc} is the open circuit voltage, and ε_G is the material's band-gap energy.

Nevertheless, there is still a term remaining from Eq. (1) to be addressed to complete the system's model, which is the current leaked across the parallel resistor I_p, which can be explained through Eq. (6).

$$I_p = \frac{V + R_s \cdot I}{R_p}. \tag{6}$$

Fig. 11 shows a breakdown of how Equations (1)–(6) interact with each other to mathematically describe the I-V output of the PV system.

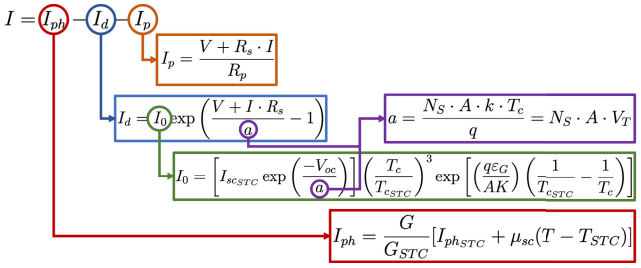


FIGURE 11. Overview of the Equations that describe the SDM.

Moreover, in order to facilitate the analysis of Fig. 11 breakdown, a summary of the variables needed for the Single-Diode representation is presented in Table 5.

To understand how this model fits into the subsystem integration, Fig. 12 shows the current against voltage (I-V) and the power against voltage (P-V) profiles of the solar panel integrated on the RaccoonBot (as validated by the model implementation from [57]). The I-V profiles

TABLE 5. Single-Diode model variables summary.

Variable	Description
I	Output current.
I_{ph}	Current generated by the solar cell.
I_d	Diode current.
I_p	Current across the parallel resistor.
I_0	Diode's reverse saturation current.
I_{sc}	Short-circuit current.
R_s	Series parasitic resistance.
R_p	Parallel parasitic resistance.
V_{oc}	Open circuit voltage.
V_T	Thermal voltage.
N_s	Cells connected in Series.
T_c	Cell's temperature.
A	Ideality factor.
k	Boltzmann constant.
V	Output voltage.
q	Electron's charge factor.
G	Solar Irradiation.
T	Temperature.
μ_{sc}	Temperature coefficient at short circuit.
ε_G	Material's band-gap energy.
STC	Standard Test Conditions

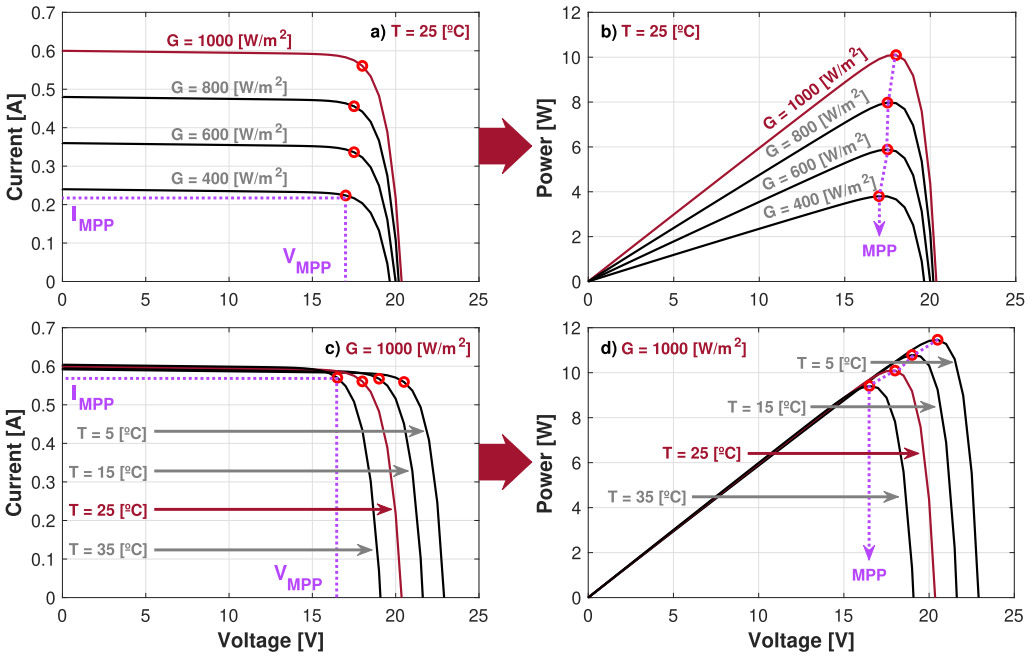


FIGURE 12. Current and Power against Voltage (I-V and P-V) Profiles of the solar panel integrated in the RaccoonBot, showing the how the Maximum Power Point (red circles in each profile) changes under different irradiation (a,b) and temperature (c,d) changes.

(Fig. 12a) and c)) are achieved through a voltage sweep, in order to estimate the resultant current at each given voltage value; on the other hand, the P-V curves (Fig. 12b) and d)) are delivered by the relation given by $P = V \cdot I$.

Also in Fig. 12, the highlighted in dark red profiles are the power and current curves of the solar panel under STC ($G = 1000$ [W/m^2] and $T = 25$ [° C]), which helps analyzing the dynamic behavior of the PV source when the irradiation or the temperature input values change. This validates if the dynamic features fulfill the requirements, since this solar panel is intended to be used as the charging source for the battery through the Buck DC-DC converter that performs the MPPT charging functions (as shown in Fig. 8). Additionally, the red circles represent the MPP in each curve, which also validates the need for power regulation in order to track the MPP to harvest more power from the PV source.

To ensure the reproducibility of the parameters characterization of the solar panel integrated on the RaccoonBot, the simulation parameters where directly taken from the specifications provided by the manufacturer's datasheet addressed in [58]; where, the solar panel for this application is a Monocrystalline 9[W] 18[V] ETFE Solar Panel from Voltaic Systems®, with part number P108. The main solar panel parameters taken from [58] used for this characterization are summarized in Table 6.

Therefore, Fig. 12c) and d) show the expected behavior of the solar panel when direct sunlight hits the solar panel under different temperature scenarios, which is a temperature range endemic from the application site (California, USA). Consequently, this validates the suitability of the PV panel to perform as the solar charger for the battery inside the robot.

TABLE 6. RaccoonBot's solar panel parameters for characterization.

Variable	Description	Value
R_s	Series parasitic resistance.	0.2[Ω]
R_p	Parallel parasitic resistance.	2000[Ω]
N_s	Cells connected in Series.	32
T_{STC}	Temperature at STC.	25[°C]
A	Ideality factor.	0.95
k	Boltzmann constant.	$1.381 \cdot 10^{-23}$[J/K]
q	Electron's charge factor.	$1.602 \cdot 10^{-19}$[C]
μ_{sc}	Temperature coefficient.	0.4[%/°C]
ε_G	Material's band-gap energy.	1.12[eV]
$I_{sc_{STC}}$	Short-circuit current.	0.6[A]
$V_{oc_{STC}}$	Open-circuit voltage.	20.45[V]

Moving forward to the integration of the solar panel with the robot, the following subsection presents the summary of the integration tests, performed as part of the *Integration Tests* stage of the V-model process from Fig. 4.

B. INTEGRATION TESTS

To validate a proper system integration between the power electronics, the digital elements and the solar panel inside the RaccoonBot, experimental data was acquired as part of

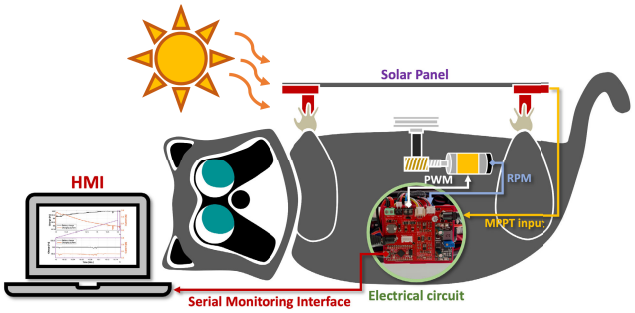


FIGURE 13. General structure for the integration tests.

the validation set of stages addressed in Fig. 4. To that end, Fig. 13 presents the general structure to acquire the data of the integrated system in this section, where HMI stands for Human-Machine-Interface which is the serial interface designed for monitoring purposes.

Consequently, Fig. 14 shows a 30[sec] sample of the data acquired during the integration tests performed to validate the implementation. Fig. 14a shows the power profiles acquired from the solar panel, the charging power and the power consumed by the robot's circuit when the RaccoonBot stops moving on the wire but with all the subsystems on.

In the power profiles from Fig. 14a, the battery discharging profile shows to be around a constant 0.025[A]±0.003[A], where the behavior reflects a clear stable power consumption when all the electrical subsystems are on. On the other hand, Fig. 14a shows how the power from the solar panel and the charging power (power consumed by the battery) is slowly decreasing, which is an expected dynamic through the fact that when batteries charge they start drowning less power from the power supply when approaching their full charge state.

To understand if the integration tests were successful or not, it is critical to address the environmental conditions under which the tests took place, to compare results with the expected behavior. Therefore, through data from [59] it can be estimated that the average solar irradiation at the moment of the tests was around 673.1$[W/m^2]$ at 15[°C], which according to the solar panel power profiles from Fig. 12, means that the expected maximum achievable power from the solar panel can be up to 7[W] under the given conditions. Hence, the power profile from the solar panel addressed in Fig. 14a clearly validates the functionality of the MPPT to achieve the MPP under the testing conditions.

Then a new question arises, how to validate that the power acquired is still the MPP available on the given conditions?, Since solar irradiation and the battery charge can both affect the energy harvested. To solve that issue, [60] brakes-down how the current and voltage behave under different input variations, where results show that to validate that the MPP is found the best solution is look at the voltage profiles, since the V_{MPP} (voltage at MPP) has almost a steady behavior through time when the MPP is reached, unlike the I_{MPP} profile that usually follows the power profile dynamics.

Therefore, Fig. 14b and Fig. 14c show the voltage and current profiles sampled from Fig. 14a decomposition, where it is validated the expected dynamics of V_{MPP} and I_{MPP}. From Fig. 14c, it can be clearly seen how the current drown by the battery is decreasing over time even if the discharging current remains the same; and, since the solar irradiation and temperature conditions remained the same during the test, it can be deduced that the only variable changing is the actual charge from the battery, which validates that the integration of the solar panel with the power electronics is capable of charging the battery as expected.

Additionally from Fig. 14, when zooming into the battery voltages addressed in Fig. 14b, it is validated how the voltage

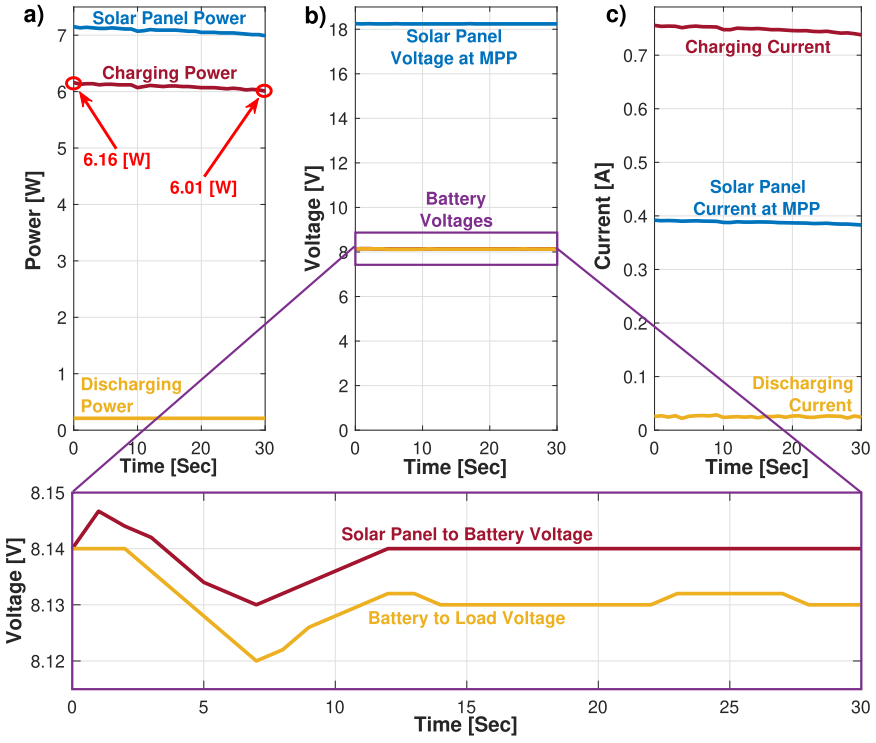


FIGURE 14. Experimental a) Power, b) Voltage, and c) Current profiles from the integration tests, taken from the solar panel and battery charging/discharging signals.

measured in the solar panel to battery node follows the same path as the voltage signal in the battery to load node. Nevertheless, the voltage drop between both profiles is due to the Schottky diode connected as the blocking diode to protect the MPPT circuit when there is no sun power, where its only objective is to block reverse energy flow from the battery into the PV array and the MPPT circuit. Hence, the dynamic features of the voltage are not affected but a small voltage drop is induced.

As a result, with all the the electromechanical subsystems in place for the RaccoonBot's implementation, the following section introduces the solar-tracking algorithm designed to enable the energy-aware mobility features of the robot. Where the algorithm's design process will be addressed as the subsystem integrated through the V-model structure, starting from the conceptualization of the algorithm, following by the simulated solar-tracking dynamics of the robot, and finally leading to the validation stage in the final product.

V. SOLAR-TRACKING ALGORITHM

As presented in Fig. 12, Temperature (T) and Solar irradiation (G) variations along a regular day make the Maximum Power Point (MPP) to have unpredictable dynamics. This is why, from the power electronics perspective, [30] explains how the issue is dealt by MPPT algorithms implemented through DC-DC converters (as we do in the RaccoonBot electrical frame from Fig. 8).

As can be analyzed from the power profiles from Fig. 12b) and d), changes in solar irradiation cause more aggressive variations in the MPP available under those environmental conditions. This also means that when there is no direct contact of the solar panel with sunlight, the available power decreases due to the lack of solar irradiation.

Consequently, the work in [61] shows how the orientation and position of the solar panels have a critical role in the energy harvesting process; which is why together with MPPT, for some scenarios where PV sources don't have direct contact to direct solar power, Solar Tracking Systems (STS) have gained greater relevance. Therefore, this section explores where the RaccoonBot fits among the STS solutions, and how the tailored solar-tracking algorithm is integrated into the development of the robot.

A. SOLAR TRACKING SYSTEMS

Usually a STS seeks to maximize the energy harvested from the PV panel by optimizing the angle at which the panel receives solar irradiation (known as angle of incidence); where the objective is to minimize the angle of incidence by modulating the orientation of the solar panels. Moreover, according to [62], STS can be understood mostly through two types of tracking systems: Single-Axis STS (SASTS) and Dual-Axis STS (DASTS).

In the case of SASTS (e.g., [63], [64]), the robotic system is designed to work by tilting the solar panel on a one axis, which is usually fixed aligned with the north or south direction, in order to rotate from east to west to track the sun from the moment it rises to sunset. To illustrate the dynamics of a SASTS, Fig. 15a, highlights the rotating axis and the path to be optimized.

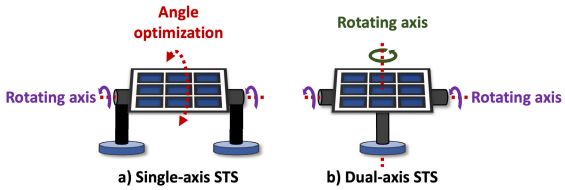


FIGURE 15. General structure of Solar Tracking Systems.

On the other hand, in DASTS (e.g., [65], [66]) the robotic system allow solar panels to rotate on two axis, usually aligned both with the north-to-south and the east-to-west axis. The main difference of these systems when compared to the SASTS, is that the dual-axis system enables tracking the seasonal variations in the height of the sunlight incidence, in addition to the regular dawn to dusk daily dynamics. To address this behavior, Fig. 15b shows the typical rotating axis in a DASTS.

But, what happens when the issue to be addressed is not only the sun's position during the day but also overcoming the obstacles (such as trees) between the sun and the solar panel in the robot? Then, fixed STS are not suitable for the RaccoonBot due to the partially shading conditions caused to both the tree shades and the sun's dynamics during the day. That is why the RaccoonBot's development also contributes to the state-of-the-art mobile STS, where the robot needs to be able to move along the wire to find a position with direct access to PV power (as shown in Fig. 16).

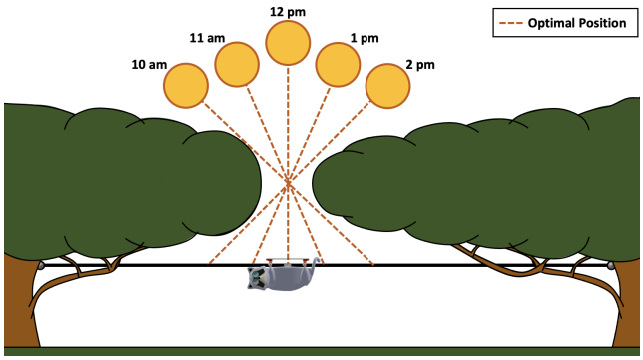


FIGURE 16. Exemplification of the RaccoonBot's need for a mobile Solar Tracking System.

Also in Fig. 16, it is illustrated why the robot needs to be able to move to be able to harvest solar power in the giving wire-traversing application, which is a requirement defined in the first stage of the V-development (*Requirements analysis and planification*) process and designed in greater details as part of stage 4: *Detailed system design*. To explain the moving resolution of the solar-tracking system implemented in the RaccoonBot, Fig. 17 presents the free body diagram of the robot, but this time through the perspective of the elements that interact for the estimation of the tracking resolution;

which are: the wheels radius, worm-gears mechanism and the encoder at the back of the motor.

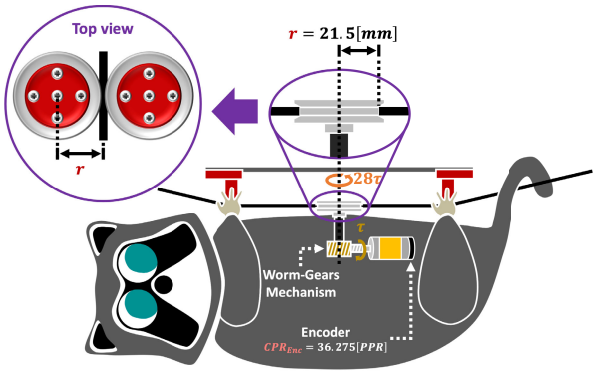


FIGURE 17. Diagram of the variables in the RaccoonBot required to estimate the solar tracking resolution.

Starting from the DC motor selected for this design (model 5203-2402-0005 from ServoCity®addressed in [67]), the attached quadrature encoder addressed in Fig. 17 has a reported interpolated resolution of 145.1 PPR [67]; yet, since for the robotic integration requires the native counts for the signal processing through the MCU, the native resolution of the motor inside the RaccoonBot is 36.275 PPR. But, since the MCU can only read 1 or 0-logic states, then the reference value taken for the encoder is 36 PPR.

Nevertheless, since the motor shaft passes through the worm-gears reduction, the mechanism's PPR resolution can be defined as shown in Eq. (7).

$$CPWR = 28 \cdot (CPR_{Enc}) = 1008[PPR], \tag{7}$$

where, $CPWR$ represents the Counts Per Wheel Revolution, and CPR_{Enc} are the Counts Per Revolution from the encoder. Now, since the radius of the RacoonBot's wheels are $r = 21.5[mm]$, the distance that the robot travels in a revolution can be defined through the perimeter of the wheels as presented by Eq. (8).

$$d = 2\pi r. \tag{8}$$

If Eq. (8) is rewritten from the perspective of the required revolutions to travel a given distance d, the result shown in Eq. (9).

$$Rev_{req} = \frac{d}{2\pi r}, \tag{9}$$

where Rev_{req} are the required wheel revolutions to travel a distance d. Nevertheless, Eq. (10) shows how the revolutions can be used to define the traveled distance using the encoder counts as measuring unit.

$$d_{enc} = (Rev) \cdot (CPWR), \tag{10}$$

where, d_{enc} is the total counts the robot has traveled through a certain number of wheel revolutions (Rev). Consequently, the way to estimate the required counts in the robot to travel a given distance d, can be explained through both Eq. (9) and Eq. (10), which results in Eq. (11).

$$x_c = (CPWR) \cdot \left(\frac{d}{2\pi r}\right), \tag{11}$$

where x_c is the total counts the robot needs to travel to cover the desired distance d, which is the equation used for the position control of the robot.

$$RB_{res} = \frac{2\pi r x_c}{CPWR} = \frac{2\pi(21.5[mm])(1)}{1008} = 0.134[mm], \tag{12}$$

where RB_{res} represents the RaccoonBot's resolution as a linear STS, which is translated as the distance the robot will traveled before receiving the first pulse from the encoder in the motor. Consequently, RB_{res} is the precision in which the RaccoonBot can track the sun through the Solar-tracking algorithm. The following subsection introduces the algorithm behind the solar-tracking features of the robot, that enable the energy-aware mobility dynamics of the robot.

B. RACCOONBOT'S SOLAR-TRACKING ALGORITHM

From the V-model adapted through the RaccoonBot development process, the stages related to the *System validation* phases (from Fig. 4) are addressed through the implementation of the solar-tracking algorithm, since this current development requires all previous subsystems to be working in order to be integrated into the robot.

As the RaccoonBot needs to track sunlight along the wire, Fig. 18 presents the general structure of the mobile STS algorithm developed for wire-traversing robots, which can also be suitable for robots that seek to find an optimal position along a horizontal line.

Starting with the peripherals, sensors, and algorithm variables initialization, the flowchart from Fig. 18 shows how the algorithm at the beginning updates the actual time (T_a) variable through the RTC (connected as addressed in Fig. 8). This is a critical feature, as it allows the system to gain awareness. After updating T_a, the robot needs to find out if the current time is inside the working ours, which is when Eq. (13) is fulfilled.

$$T_w \leq T_a \leq T_s, \tag{13}$$

where T_w is the waking-up time and T_s the sleeping time. When the robot finds through the communication with the RTC (RV-8803), that the current time is between the waking and the sleeping hours, the robot will run the solar-tracking algorithm, meanwhile if the actual time is out of that range, the robot will enter a "Torpor" phase. This feature is bio-inspired from the dormant state called "Torpor" which is a state where, according to [68], Raccoons slow their metabolism and their body temperature in response to adverse environmental conditions, and inside their dens they fall asleep wrapped around their tails to keep them warm. Unlike an hibernation stage, [68] explains that this state can as short as an overnight phase.

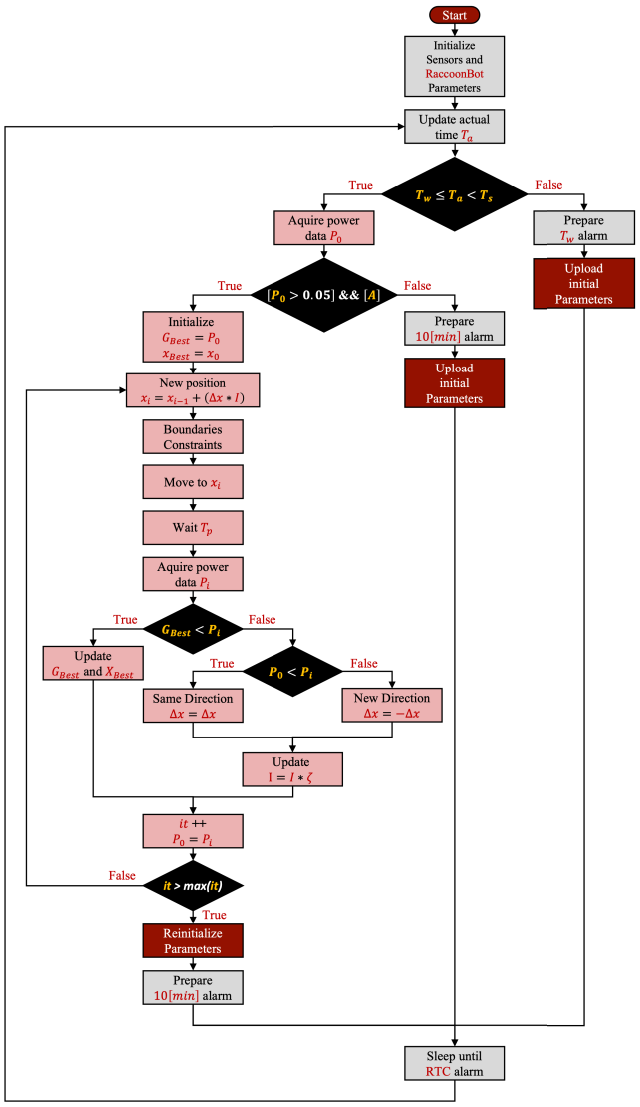


FIGURE 18. Flowchart diagram of the RacoonBot's STS algorithm.

In the case of RaccoonBot, this translates as the microcontroller entering a deep-sleep mode, which disables the DC-DC converters related to nonessential elements, where only the DC-DC converter connected to the MCU (as shown in Fig. 8) remains on; keeping the robot alive with reduced power consumption during the sleep-mode. This enables a slower discharge rate of the battery during sleep time. The addition of the "Torpor" phase enables a significant part of RacoonBot's endurance. On the other hand, when Eq. (13) is fulfilled the robot wakes-up and decides if the solar tracking algorithm will be carried out, enabling adaptability and awareness of its real-world environment.

To make that decision, the RaccoonBot uses an INA219 power sensor to acquire an initial power measurement (P_0), where the first condition seeks to find out through the power signals from the solar panel if there is available sun power, achieved by the knowledge that the INA219 gives a value of $P_0 \geq 0.05\ [W]$ under sunny conditions when the robot is under tree shades, since low solar irradiation is still able to pass through tree leaves. Consequently, the threshold for the solar-tracking algorithm is taken as $P_0 \geq 0.05\ [W]$.

The second condition to be fulfilled to enter the solar tracking algorithm is represented by the variable A that can be addressed through the conditions from Eq. (14).

$$A = (P_0 < (G_{Best} - G_{Best} \cdot 0.2)) \vee (G_{Best} == 0), \quad (14)$$

where G_{Best} represents the global best measurement of the power from the solar panel, which was taken when the robot was located at the x_{Best} position in the wire.

Dwelling further into the meaning of the conditions of Eq. (14), the condition will be fulfilled when the actual acquired power (P_0) is 20% less than the G_{Best} power, or when the initial conditions are uploaded ($G_{Best} = 0$), to ensure that the solar tracking algorithm will be carried out at least one time.

Once the RaccoonBot decides its time to perform the tracking algorithm, the initial values in the given position are initialized as the first G_{Best} and x_{Best} values. Consequently, the next step is to estimate the next position that the robot is going to move to at the iteration i, which is estimated through Eq. (15).

$$x_i = x_{i-1} + (\Delta x \cdot I), \quad (15)$$

where x_i represents the robot's new position, x_{i-1} is the actual one, meanwhile Δx is the step size that the RaccoonBot will take, and I is the inertia coefficient that allows the algorithm to converge through time.

Following the x_i estimation, a boundaries constraints stage is applied to ensure that the robot remains in-between the limits of the wire (boundaries constraints block from Fig. 18), this constraining process can be understood through the flowchart presented in Fig. 19.

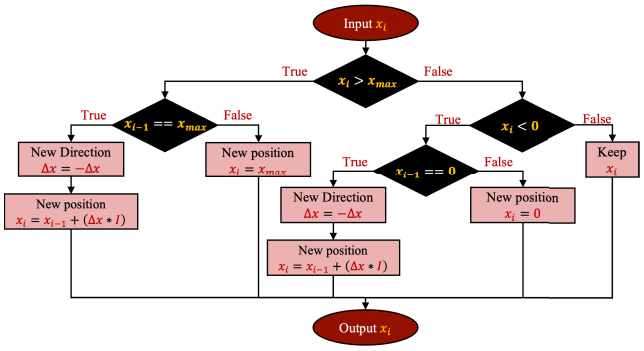


FIGURE 19. Flowchart diagram representation of the boundaries constraints for x_i.

After estimating and constraining the new position for the robot, the RaccoonBot travels along the wire to that desired position. Subsequently, the RaccoonBot waits a period of time T_p as a standard procedure for devices that involve

MPPT circuits; since, according [30], [69], MPPT algorithms can take 50 [mSec] to settle at the MPP, and the idea is to acquire the new P_i when the signal is established in the maximum power point available at that current position to compare them to the previous ones. Thus, T_p is assigned be ≥ 50 [mSec].

Following the waiting period, the acquired P_i and G_{Best} are compared and ranked to decide if the robot found a new G_{Best} and hence a new x_{Best}. On the other hand, if there is no new G_{Best}, the robot decides through the algorithm the direction for the following step based on the vector generated by P_0 and P_i as shown in Fig. 18.

Fig. 18 shows that after defining the new direction, the inertia coefficient I is updated through Eq. (16), which foreshadows how the algorithm starts to converge towards the final position.

$$I = I \cdot \zeta, \tag{16}$$

where ζ represents is the dampening coefficient, which is the rate at which the step size will decrease. Finally, the P_0 value is updated and if the maximum iterations are not reached yet, then the algorithm runs once again.

In the case that the maximum iterations have been reached, the algorithm is reinitialized, a 10[min] alarm is set and the robot enters its ''Torpor'' stage, where it sleeps until the RTC wakes the MCU through an interrupt signal triggered by the RTC's alarm. An important difference between the ''*Reinitialize Parameters*'' block and the ''*Initial Parameters*'' reinitialization block, is the step size initialization (Δx); since under initial conditions, the step size is bigger to cover a wider area. But when the algorithm uses the ''*Reinitialize Parameters*'' block, it is under the assumption that it is closer to the sun; therefore Δx is reinitialized with a smaller size to provide a finer and faster search with less power consumption.

To summarize the main solar tracking features of the algorithm, Fig. 20 shows the RaccoonBot's behavior in a sequence of figures addressing how the mechatronic system converges into the G_{Best}.

Fig. 20a shows the initial position of the RaccoonBot, where it is designed to start on the left side of the wire and to have an initial positive Δx assigned to the right (moving to the tail side of the robot). Therefore, with the initial conditions, the robot takes its first step towards the x_1 position in the wire. In that position, the robot defines whether it found a better position in terms of the energy harvested, or if it is not ranked better when compared to x_0.

After the RaccoonBot takes the decision on what direction to take in the following iteration, the RaccoonBot takes a damped step towards x_2. If the RaccoonBot finds that direction led to a new best position energy wise, then the robot will continue to explore in that direction and move in that same direction for its new position x_3. Yet, Fig. 20b shows that when the step to be taken leads to a position that is outside the boundaries, then the robot gets a new cropped position to move on to, through the process described in Fig. 19.

By comparing the step taken from x_2 to x_3 (Fig. 20b) against the step from x_3 to x_4 (Fig. 20c), it is clear that the step taken to x_4 is greater than the one taken to x_3 but still smaller than the one taken to the x_2 direction. That is because the rate on which the subsequent steps gradually decrease is not affected by the boundaries the system takes to keep the robot safely inside the searching space, nevertheless after the boundaries are applied, the robot's Δx sign changes to ensure that the robot looks for a new G_B outside the reached boundary position.

In the case of Fig. 20c and Fig. 20d, it can be seen how the dampening ratio ζ affects the inertia coefficient I to allow the robot to start converging as the iterations go. Therefore, after the RaccoonBot continues its solar-tracking path, it starts

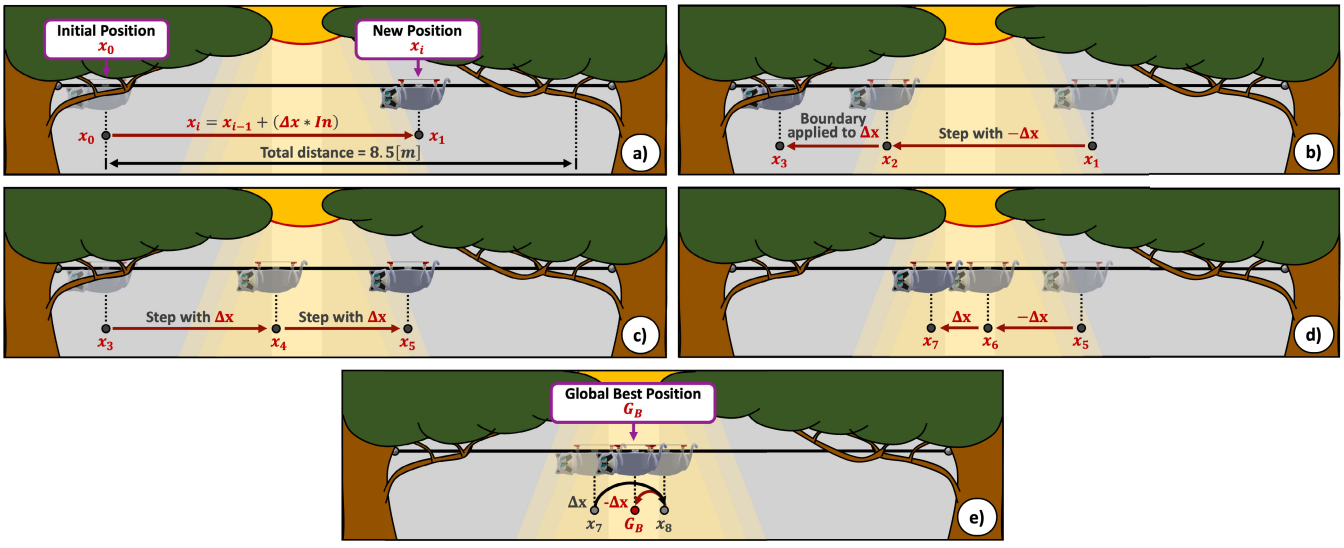


FIGURE 20. RaccoonBot's behavior during the solar-tracking algorithm.

converging into its final position by moving around the point in the wire where the robot finds more solar power through time, which leads to the robot resting in the position where it can harvest the most solar power to charge the battery. Altogether, the RacoonBot's awareness, adaptability and endurance against different environmental conditions, were taken into consideration through the algorithm's bio-inspired features, which reinforces our robot's reliability.

Now, since [47] explains how the general structure of a mechatronic system developed through the V-model requires to include a simulated performance evaluation to move on into the system's experimental validation, the following subsection presents the simulated performance of the system integrated to the solar-tracking algorithm, which is later validated through the results in this paper.

C. VALIDATION TESTS

To fulfill the simulated validation required by the development process, Fig. 21 presents the first environment created to analyze the expected behavior of the RaccoonBot, which also serves to properly calibrate the solar tracking algorithm according to the implementation scenario.

The wire position plot from Fig. 21 represents the RaccoonBot tracking the sun in a wire which is 8[m] long, where the sun position shows where the robot should expect to harvest more solar power. On the other hand, the power profile from Fig. 21 is simulated to be as a Gauss membership function as presented by Eq. (17), in accordance to the expected irradiation performance visualized in Fig. 20.

$$f(x) = e^{\frac{-(x-c)^2}{2\sigma^2}}, \tag{17}$$

where σ is the standard deviation taken as 0.5[m], c is the function's mean (expected position of the MPP) taken as 3[m] for this simulation.

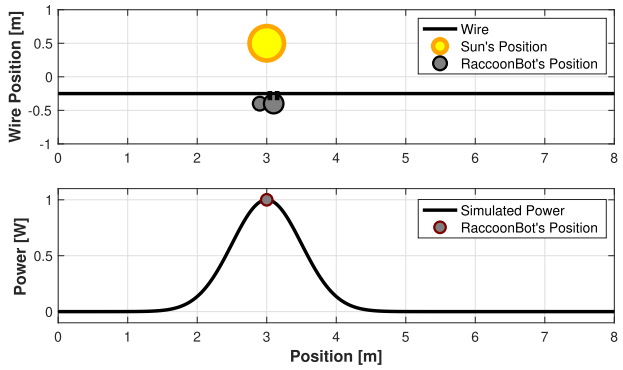


FIGURE 21. Simulated environment for the solar tracking features of the RaccoonBot.

Among the properties of membership functions that make them suitable for this simulated validation, is that membership functions take into consideration normalized values between 0 and 1. This simplifies the validation process, by identifying the point where the robot can harvest the most power while tracking the sun, independently from the actual wattage or irradiation values it might represent. This approach is particularly suited for the simulation, as its primary goal is to validate the performance and dynamics of the solar tracking algorithm.

To that end, Fig. 22 the simulated dynamics of the solar tracking subsystem embedded in the RaccoonBot. First, the actual power profile in Fig. 22 represents the power sensed by the robot at each iteration, where it is reflected how the inertia coefficient enable the robot to converge through time in the position where the robot is able to harvest more solar power.

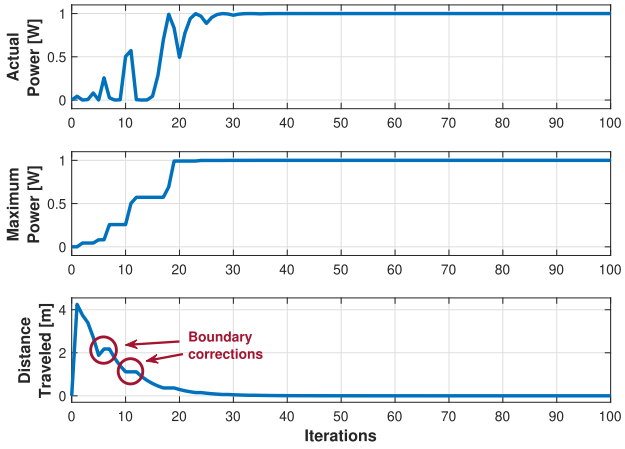


FIGURE 22. Performance profiles from the simulated validation tests.

The maximum power curve from Fig. 22, shows the convergence plot of the global maximum found over time, which enables validating the algorithm's searching process as a single solar tracking agent. When compared to the actual power plot, it can be seen how the robot "bounces" around the position where it can harvest more solar power and the algorithm makes it converge through time.

Additionally from Fig. 22, the distance traveled profile highlights the convergence plot of the robot's displacement at each step during each iteration, where it can be seen how the first Δx was taken as 4.25[m], which constantly decreases due to the dampening coefficient coded in the algorithm. Yet, the distance traveled convergence plot also shows two segments where the decreasing rate was not constant, that modification occurred due to the robot reaching the boundaries and thereby to the correcting decisions taken as addressed in Fig. 19. To deepen into the effect of the boundaries correction process, Fig. 23 shows two more simulated results to explore in a better way the solar tracking features of the RaccoonBot starting from $x_0 = 0$.

On the one hand, Fig. 23a shows a simulated power profile, where multiple local maximums are placed near the global best position, in order to evaluate the algorithm's ability to avoid getting stuck into a local solution. Below the power profile, the performance evaluation curves are presented as done so previously for 22. The test from Fig. 23a, allows

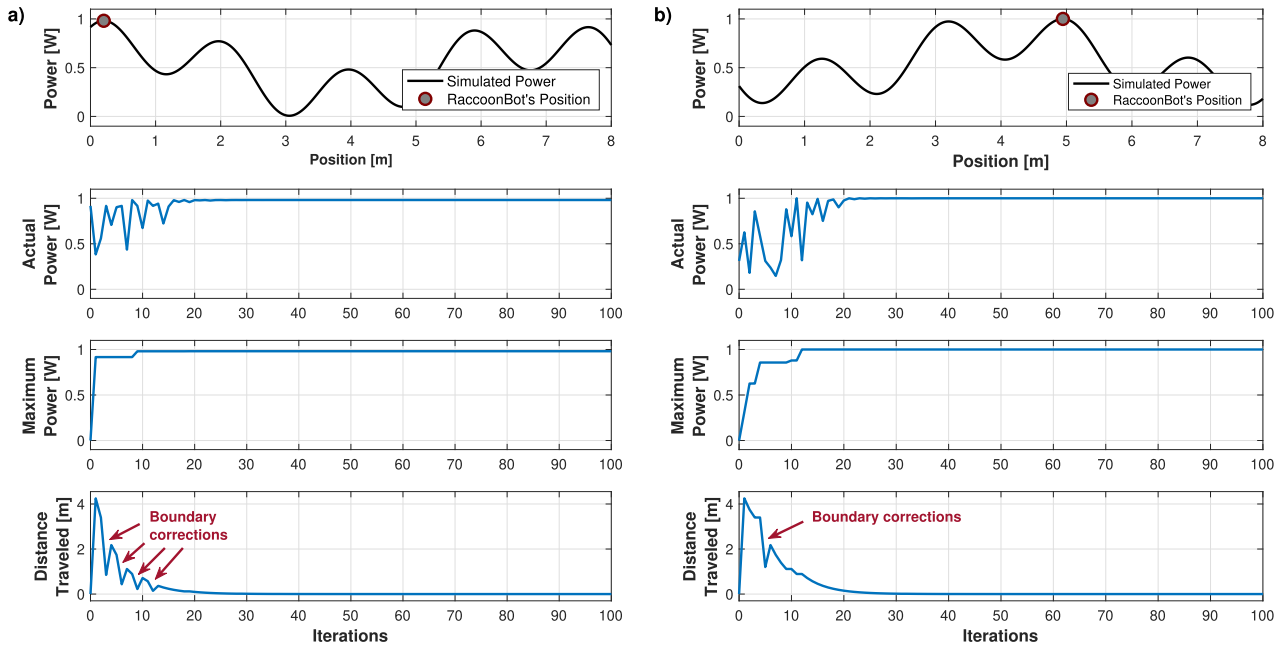


FIGURE 23. Algorithm validation tests against multiple harvesting profiles, simulating when there are multiple local maximums around the global best.

analysis of how the boundary correction affects the harvesting process when the best solution is near one end of the wire. It can be observed that, the maximum power curve of Fig. 23a shows the convergence plot of the global maximum found over time, enabling the algorithm to validate the searching process as a single solar tracking agent. When compared to the actual power plot, it can be seen how the robot "bounces" around the position where it can harvest the most solar power and the algorithm converges.

To address the benchmark equation related to this test, Eq. (18) shows the mathematical expression used to simulate the power profile in this experiment.

$$f(x) = \frac{\sin(x+1) + \sin((3.33)\cdot x + 1) + 1.8}{3.8}. \quad (18)$$

In this particular case, it can be seen how even when the robot started the first iteration by taking a step across the half of the wire ($\Delta x = 4.25[m]$), the robot managed to come again to the end of the wire to converge into the Global best solution. Additionally, it can be noticed that because the optimal position is near the starting point, convergence is quickly reached.

On the other hand, Fig. 23b presents the other set of plots related to a scenario where the best position is nearby the middle of the wire. For this case study, Eq. (19) shows the equation that describes the power profile used for this simulated environment.

$$f(x) = \frac{\sin(x+10) + \sin((3.33)\cdot x + 10) + 2}{4}. \quad (19)$$

It can be seen from this test how the boundary corrections decreased to only one when compared to the previous tests, since in this case the robot spends more time in the middle of the wire. Here, the maximum power curve of Fig. 23b shows the convergence plot of the global maximum takes longer than Fig. 23a, this is because the optimal position here is further away from the initial position at $x_0 = 0$. When compared to the actual power plot, it can be seen how the robot "bounces" around the position where it can harvest the most solar power and converges through time, as expected.

It is important to highlight that both Fig. 23a and Fig. 23b, are exaggerated scenarios designed to take the algorithm validation tests to the limit, since in the actual experimental setup where the RaccoonBot will be placed, the expected profile has greater affinity to the power profile from Fig. 21.

Since this tests where specially design to evaluate the performance of the algorithm, in addition to find the proper tunning for the final experimental validation of the RaccoonBot's implementation, Table 7 shows the parameters used for the simulated results in this section, which are the same parameters tuned for the experimental results introduced in the following section.

TABLE 7. STS algorithm calibration parameters.

Variable	Description	Parameter
Δx_i	Initial step size	4.25[m]
Δx_f	Finer initial step size	1.5[m]
$\max(x)$	Boundary constraint.	8.3[m]
I	Initial inertia coefficient.	1
ζ	Initial dampening ratio.	0.8
x_0	Initial position.	0[m]
it_{sim}	Simulated solar tracking iterations.	100
it_{exp}	Experimental solar tracking iterations.	50

It is important to highlight that, the difference between the final implementation of the RaccoonBot and the simulated validation tests relies in the step sizes, since in all the simulation tests the RaccoonBot start from $x_0 = 0$ with an initial step size of $\Delta x = 4.25[m]$. Meanwhile in the experimental setup, the RaccoonBot is left there for several days to validate its reliability, and since Fig. 18 shows that the complete structure of the algorithm takes into consideration several sleeping and waking-up operations, where during "working hours" the robot only uses $\Delta x = 4.25[m]$ at the beginning of each day, and uses $\Delta x = 1[m]$ every time the robot wakes up after the first solar-tracking operation.

Another important difference between the simulated and the real environment, can be found in the number of solar tracking iterations the robot takes to finally rest in the charging position; since Table 7 introduces that for the experimental setup the robot will only take 50 iterations for its tracking operations, mostly decided due to the performed validation tests, where Figs. 22 and 23 show that 50 iterations are enough to fulfill the robot's task.

Moving on to the final outcome developed through the V-model design methodology adapted for this environmental application, the following section focuses in the RaccoonBot's itself, from the perspective of the final product developed and also from the validation point of view; where according to [47], is a critical measure to evaluate the success of a mechatronic system design through V-model structures.

VI. RESULTS

Summarizing the results carried out by the V-model development process, adapted to the research and development of environmental solar-tracking robots, Fig. 1 shows the RaccoonBot as it was deployed to gather the data addressed in this section.

As this paper contributes to the system integration state-of-the-art, through the adapted V-model design methodology to bridge mechatronic systems and power electronics to develop a robotic system for environmental monitoring, the first result to address in this section is the adaptation of the V-model for environmental monitoring robotics itself. To simplify understanding the development process described in Fig. 4, and how it is adapted throughout this paper, Fig. 24 shows the V-model with the figures from this paper embedded in each stage as they relate to each stage of the RaccoonBot development and deployment.

From Fig. 24, it can be seen how the implementation is the actual base of the V structure. Consequently, the left side of the diagram represent all the sequence that throughout this paper the requirements were translated into the system design phases; meanwhile, the right side shows how through the implementation stage, the development process enters an iterative validation process towards the final implementation of the RaccoonBot in its actual environment (final product deployment).

Fig. 24 also presents how the final stage is the expected demonstration that the prototype operates as it should, and the

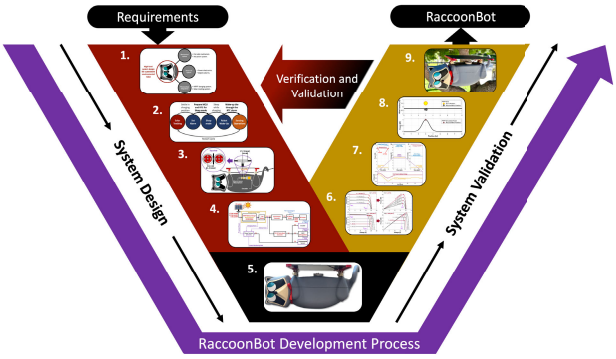


FIGURE 24. V-model adapted to the RaccoonBot's resultant prototype.

RaccoonBot is the outcome at the end of the development process. Therefore, since stage 9 is the *Operation and Prototype Evaluation* phase described in Fig. 4, the following results are focused in the operation and performance evaluation of the robotic system. Hence, Fig. 25 shows the interior of the RaccoonBot in the experimental setup, in addition to the connection to the monitoring computer to acquire the data to be analyzed.

FIGURE 25. RaccoonBot during the data acquisition experimental setup.

For the data acquired through the RaccoonBot, a USART interface is used to communicate the HMI (Human-Machine Interface) with the MCU inside the RaccoonBot, as presented in Fig. 25. In that setup, it is also shown the initial position of the RaccoonBot before each test, always defining $x_0 = 0[m]$ in one end of the wire where there's no direct access to solar power.

In order to evaluate the performance of the RaccoonBot, Fig. 26 shows the power profiles acquired through the experimental setup at 2 pm during different days of the week, in order to evaluate under similar conditions the consistency in the operation of the RaccoonBot.

Additionally from Fig. 26, the mean values are highlighted to address the average best position in which the robot harvested the most solar power, and the average maximum

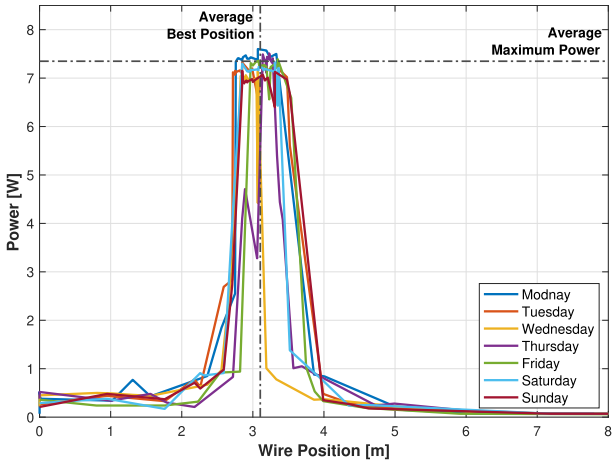


FIGURE 26. Acquired power profiles along a week.

power achieved by the solar power in that position on the wire.

Among the power profiles, the consistency of the dynamic features of the RaccoonBot can be graphically analyzed, since even when the weather conditions were similar but not exactly the same during the different tests, the robot still managed to find the best position on the wire to still be able to harvest solar power to charge its battery.

The power profiles show how each new day of the week, RaccoonBot starts at position $x_0 = 0$ and finds the best position for maximum average power. Graphically Fig. 26 reveals that on some days, the maximum power can be higher (Monday), and on other days the maximum power can be found at a different position (Sunday). The results of Fig. 26 showcase RacoonBot's adaptability and awareness, enabling its endurance.

To evaluate even further the RaccoonBot's operation and performance throughout the different deployment tests, Table 8 shows a summary of the quantified results from the daily tests, which provides a clear perspective that the robot autonomously found the best position to harvest the most solar power each day.

TABLE 8. Summary of the RaccoonBot's quantified results.

	Maximum Power [W]	Best position [m]
Monday	7.60	3.07
Tuesday	7.32	2.85
Wednesday	7.14	3.07
Thursday	7.51	3.23
Friday	7.39	3.35
Saturday	7.33	3.31
Sunday	7.15	2.85
Mean	7.3486	3.1043
Standard deviation	0.1708	0.2042

From Table 8, it is clear that after settling down in the best position, the solar panel in the given configuration was able to supply every day at least over 7[W], to charge the battery; which enables successfully validating the integration through the design methodology. Yet, to validate the dynamic behavior of the integrated solar-tracking system, a comparison against the simulated behavior is needed.

Following the simulated environment from Fig. 21, the simulated and experimental behavior can be compared under the same circumstances. To achieve a fair comparison, the mean values can be used to generate the simulated profile through the average maximum power and the average best position it was found on; which is why those values are estimated and addressed first in Fig. 26, and later presented in Table 8.

Therefore, by substituting the estimated mean (3.1043[m]) and standard deviation (0.3416[m]) into the membership function from Eq. (17) to fit the power curves, the dynamics of the robot can be compared to the simulated scenario. If the power profiles from Fig. 26 are normalized and then the 7.60[W] is used as a reference for the maximum achievable power, a simulated profile can be obtained. Fig. 27 shows this simulated power profile compared against the normalized power profiles from the experimental tests and the optimal position to obtain maximum power. From the perspective of the normalized power profiles, and since the normalization process was done through the maximum power achieved on Monday, the resultant power curve can be taken to compare the dynamic optimization features of the algorithm embedded in the RaccoonBot against its expected behavior.

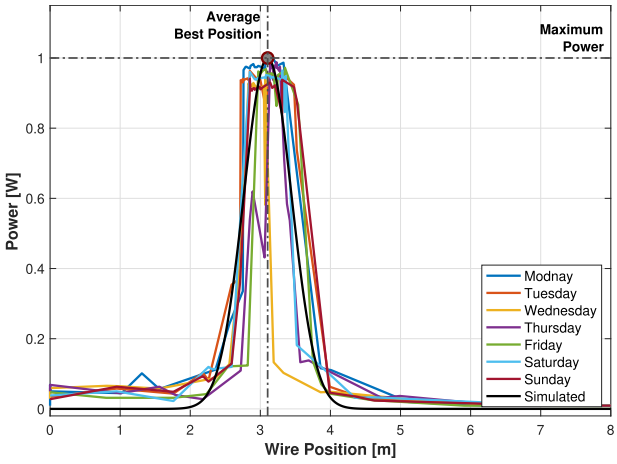


FIGURE 27. Normalized power profiles compared to the simulated scenario.

From the perspective of the normalized power profiles, and since the normalization process was done through the maximum power achieved on Monday, that power curve can be taken to compare the dynamic optimization features of the algorithm embedded in the RaccoonBot, against its expected behavior. To that end, Fig. 28 shows the performance profiles from the simulated test compared against Monday's solar-tracking test.

In Fig. 28, the actual power plots are related to the power measured at each iteration, meanwhile the maximum power plots show the data from the global best saved value

through iterations. In both cases (actual and maximum power profiles), it is clear that the dynamic features of the simulated and the experimental tests are similar in their behavior, although they might momentarily differ in their magnitudes during some iterations.

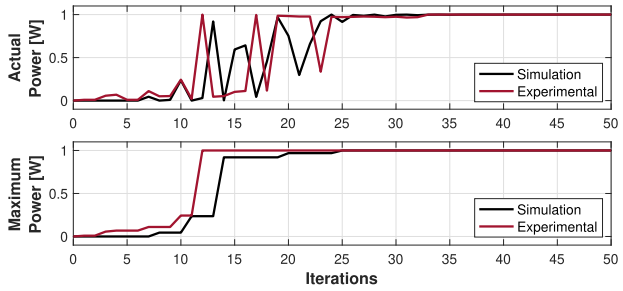


FIGURE 28. Performance profiles from the simulated test against the experimental test.

Analyzing the actual power plots in Fig. 28, it can be seen that before iteration 15 the optimization paths followed by the simulation and the RaccoonBot itself, are almost the same. Yet, mostly after iteration 15 and until iteration 19 the algorithm shows its greater differences between the expected and the real behaviors, since in the experimental setup the trees create a non homogeneous shade in the wire, which makes it receive more or less solar power when compared to the regular surface from the Gauss membership function from the simulation.

The non-homogeneous shades from the trees also enhances the robustness evaluation from the algorithm, since between iterations 20 and 25 the algorithm manages to still converge in the best position, in a similar way that the simulated test predicted.

Since the maximum power profiles from Fig. 28 are the convergence plots from the algorithm in both the experimental and simulated tests, these curves show in a better way the similarity between the expected and the real performance, which validates (as required by the V-model) the affinity between the designed algorithm and its implementation. But, to understand how the robot's positioning after the solar-tracking looks like in the experimental setup, Fig. 29 shows the RaccoonBot after Monday's session.

In the front view perspective from Fig. 29, the experimental setup can be seen with greater detail, where the irregular shades from the trees under the charging position can be visualized. Yet, to clearly see from the robot's solar panel perspective how the algorithm leads to a position where the robot can charge its battery, the bottom view from Fig. 29 presents a perspective where it can be seen how the robot is aligned with the sun after the solar-tracking process.

Addressing the final position of the robot from the bottom view perspective, that scenario makes it clear why the solar-tracking algorithm is needed, and why the convergency properties of the algorithm are so important for this implementation; mostly due the fact that, the shades from the trees make it nearly impossible for the robot to achieve

FIGURE 29. Front and bottom views from the RaccoonBot after the solar-tracking positioning.

a constant charging rate if the robot doesn't follow the sun through time.

To finally validate the robotic system integration and its capacity to fully charge the battery in the given configuration, Fig. 30 shows the charging profiles of the battery inside the RaccoonBot. This data, was acquired under the condition that the only power supply connected to the battery was the solar panel in a realistic scenario. For this test, the battery was initially discharged (1 pm) and the test ended when the battery got fully charged (4:30 pm).

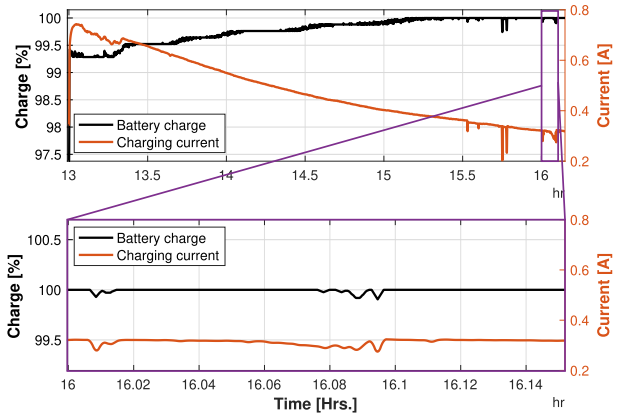


FIGURE 30. RaccoonBot's battery charging profiles.

Fig. 30 presents the current supplied after the MPPT circuit by the solar panel, meanwhile the battery charge profile is taken as the voltage at the terminals of the battery, which allows understanding and validating the energy transferred from the PV source to charge the RaccoonBot's battery.

The charging process follows a multi-stage behavior, where the battery initially charges at a higher current due to a lower state of charge, following a constant current (CC) phase. As the charge level increases, the charging current gradually decreases, transitioning into a constant voltage (CV) phase, where the battery voltage stabilizes near its maximum value.

The zoomed-in section in Fig. 30 highlights the point at which the charging current reaches a steady level, confirming that the battery is approaching its full charge state. This behavior aligns with expected battery charging characteristics, where a battery draws more current when deeply discharged and gradually reduces current intake as it nears full capacity. Furthermore, the MPPT controller ensures efficient solar power utilization, dynamically adjusting to changes in solar irradiance and maximizing energy transfer throughout the charging process.

This final test shows a proper integration between the renewable energy source and the robotic system, which fully validates the RaccoonBot's development process, since the final product results as a reliable robotic platform for environmental monitoring. Thereby, these results validate that the contributed adaptation of the V-model development process for environmental robotics was successfully applied for the RaccoonBot case study.

VII. CONCLUSION

This work introduces the RaccoonBot, a solar-powered wire-traversing robot designed for persistent environmental monitoring, with key innovations in power electronics, mechatronic design, and energy management. By leveraging the V-model methodology, the development process ensured seamless integration of a fail-safe mechanical design, efficient power electronics, and innovative solar energy management, resulting in a fail-safe robot capable of operating in remote, difficult-to-access locations.

The robot's worm-gear mechanism achieved a 28:1 torque amplification and non-backdrivability, ensuring reliable traversal and stationary stability even against adverse weather conditions. Its electrical system, with over 90% power conversion efficiency and a robust MPPT-based charging system, enabled sustained operations by efficiently harvesting solar energy.

Experimental results confirmed the RaccoonBot's ability to dynamically locate and converge on positions along the wire with maximum solar exposure, achieving a tracking resolution of 0.134 [mm]. The system maintained a stable charging power of approximately 6.01 [W] under real-world conditions while consuming just 0.025 [A] in standby mode. These features, together with the energy-aware solar-tracking algorithm, enable weeks-long autonomous operation without external intervention. That end, was sustained by the fact that without the solar integration, the Robot could only last around 5 hours of operation, which was extended to over a week (without completely draining the battery) after the PV integration.

The RaccoonBot exemplifies the synergy between sustainable energy and robotics, offering a blueprint for the future of autonomous systems in environmental conservation. By addressing critical challenges in energy-aware mobility and robust system design, this work sets the stage for scalable and versatile applications, advancing the role of solar-powered robotics in creating a greener, smarter world.

REFERENCES

[1] IEA. (2023). *Tracking Clean Energy Progress 2023*. [Online]. Available: https://www.iea.org/reports/tracking-clean-energy-progress-2023

[2] J. Dhilipan, N. Vijayalakshmi, D. B. Shanmugam, R. J. Ganesh, S. Kodeeswaran, and S. Muralidharan, "Performance and efficiency of different types of solar cell material—A review," *Mater. Today Proc.*, vol. 66, pp. 1295–1302, May 2022.

[3] A. A. Hassan, M. El Habrouk, and S. Deghedie, "Renewable energy for robots and robots for renewable energy—A review," *Robotica*, vol. 38, no. 9, pp. 1576–1604, Sep. 2020.

[4] S. Morton, R. D'Sa, and N. Papanikolopoulos, "Solar powered UAV: Design and experiments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 2460–2466.

[5] A. Bender, B. Whelan, and S. Sukkarieh, "A high-resolution, multimodal data set for agricultural robotics: A ladybird's-eye view of brassica," *J. Field Robot.*, vol. 37, no. 1, pp. 73–96, Jan. 2020.

[6] A. A. Chand, K. A. Prasad, E. Mar, S. Dakai, K. A. Mamun, F. R. Islam, U. Mehta, and N. M. Kumar, "Design and analysis of photovoltaic powered battery-operated computer vision-based multi-purpose smart farming robot," *Agronomy*, vol. 11, no. 3, p. 530, Mar. 2021.

[7] A. Colaprete, "Volatiles investigating polar exploration rover (VIPER)," NASA, Washington, DC, USA, Tech. Rep. 20210015009, Apr. 2021.

[8] E. Mendez-Flores, A. Pourshahidi, and M. Egerstedt, "Raccoonbot: An autonomous wire-traversing solar-tracking robot for persistent environmental monitoring," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, 2025.

[9] N. Pouliot and S. Montambault, "Geometric design of the LineScout, a teleoperated robot for power line inspection and maintenance," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 3970–3977.

[10] M. Nayyerloo, S. Yeganehparast, A. Barati, and M. Foumani, "Mechanical implementation and simulation of monolab, a mobile robot for inspection of power transmission lines," *Int. J. Adv. Robotic Syst.*, vol. 4, no. 3, pp. 381–386, 2007.

[11] A. J. Phillips, J. M. Major, and G. R. Bartlett, "Line inspection robot and system," U.S. Patent 8 666 553, Mar. 4, 2014.

[12] J. Sawada, K. Kusumoto, T. Munakata, Y. Maikawa, and Y. Ishikawa, "A mobile robot for inspection of power transmission lines," *IEEE Power Eng. Rev.*, vol. 11, no. 1, p. 57, Jan. 1991.

[13] N. Ekren, Z. Karagöz, and M. Şahin, "A review of line suspended inspection robots for power transmission lines," *J. Electr. Eng. Technol.*, vol. 19, no. 4, pp. 2549–2583, May 2024.

[14] A. B. Alhassan, X. Zhang, H. Shen, and H. Xu, "Power transmission line inspection robots: A review, trends and challenges for future research," *Int. J. Electr. Power Energy Syst.*, vol. 118, Jun. 2020, Art. no. 105862.

[15] P.-L. Richard, N. Pouliot, F. Morin, M. Lepage, P. Hamelin, M. Lagac, A. Sartor, G. Lambert, and S. Montambault, "LineRanger: Analysis and field testing of an innovative robot for efficient assessment of bundled high-voltage powerlines," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 9130–9136.

[16] M. R. Bahrami, "A novel design of an electrical transmission line inspection machine," in *Proc. Adv. Mech. Eng., Sel. Contrib. Conf. 'Mod. Eng., Sci. Educ.'*, Saint Petersburg, Russia. Cham, Switzerland: Springer, Jan. 2016, pp. 67–73.

[17] B. Zhang, J. Hou, G. Huang, X. Liu, D. Xue, and X. Wen, "Discussion on a new design of overhead transmission line inspection robot," in *Proc. 3rd Int. Conf. Robot., Control Autom. Eng. (RCAE)*, Nov. 2020, pp. 1–5.

[18] G. Fang and J. Cheng, "Design and implementation of a wire rope climbing robot for sluices," *Machines*, vol. 10, no. 11, p. 1000, Oct. 2022.

[19] K. H. Cho, Y. H. Jin, H. M. Kim, and H. R. Choi, "Development of novel multifunctional robotic crawler for inspection of hanger cables in suspension bridges," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 2673–2678.

[20] H. M. Kim, K. H. Cho, Y. H. Jin, F. Liu, J. C. Koo, and H. R. Choi, "Development of cable climbing robot for maintenance of suspension bridges," in *Proc. IEEE Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2012, pp. 606–611.

[21] K. H. Cho, Y. H. Jin, H. M. Kim, H. Moon, J. C. Koo, and H. R. Choi, "Caterpillar-based cable climbing robot for inspection of suspension bridge hanger rope," in *Proc. IEEE Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2013, pp. 1059–1062.

[22] G. Fang, G. Zhang, G. Zheng, and L. Yao, "Development of pneumatic robot for climbing sluice wire rope," *Chin. Hydraul. Pneum.*, vol. 2, pp. 170–176, 2021.

[23] S. Kirchgeorg and S. Mintchev, "Multimodal aerial-tethered robot for tree canopy exploration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 6080–6086.

[24] S. Kirchgeorg, E. Aucone, F. Wenk, and S. Mintchev, "Design, modeling, and control of AVOCADO: A multimodal aerial-tethered robot for tree canopy exploration," *IEEE Trans. Robot.*, vol. 40, pp. 592–605, 2023.

[25] G. Notomista, Y. Emam, and M. Egerstedt, "The SlothBot: A novel design for a wire-traversing robot," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1993–1998, Apr. 2019.

[26] S. Farzan, V. Azimi, A.-P. Hu, and J. Rogers, "Adaptive control of wire-borne underactuated brachiating robots using control Lyapunov and barrier functions," *IEEE Trans. Control Syst. Technol.*, vol. 30, no. 6, pp. 2598–2614, Nov. 2022.

[27] L. F. P. Oliveira, A. P. Moreira, and M. F. Silva, "Advances in forest robotics: A state-of-the-art survey," *Robotics*, vol. 10, no. 2, p. 53, Mar. 2021.

[28] M. Egerstedt, *Robot Ecology: Constraint-Based Design for Long-Duration Autonomy*. Princeton, NJ, USA: Princeton Univ. Press, 2021.

[29] F. Angelini, P. Angelini, C. Angiolini, S. Bagella, F. Bonomo, M. Caccianiga, C. D. Santina, D. Gigante, M. Hutter, T. Nanayakkara, P. Remagnino, D. Torricelli, and M. Garabini, "Robotic monitoring of habitats: The natural intelligence approach," *IEEE Access*, vol. 11, pp. 72575–72591, 2023.

[30] N. Femia, G. Petrone, G. Spagnuolo, and M. Vitelli, *Power Electronics and Control Techniques for Maximum Energy Harvesting in Photovoltaic Systems*. Boca Raton, FL, USA: CRC Press, 2017.

[31] I. Graessler and J. Hentze, "The new V-model of VDI 2206 and its validation," *at - Automatisierungstechnik*, vol. 68, no. 5, pp. 312–324, May 2020.

[32] M. Fowler and J. Highsmith, "The agile manifesto," *Softw. Develop.*, vol. 9, no. 8, pp. 28–35, Jan. 2001.

[33] K. Petersen, C. Wohlin, and D. Baca, "The waterfall model in large-scale development," in *Proc. 10th Int. Conf. Product-Focused Softw. Process Improvement*, PROFES 2009, Oulu, Finland. Cham, Switzerland: Springer, Jan. 2009, pp. 386–400.

[34] A. O. V. Rapalo, C. A. B. Cortés, and J. L. OrdoñezÁvila, "Design and analysis of a pneumatic end-effector with an air contraction clamp for soft robotics," in *Proc. 9th Int. Conf. Control Robot. Eng. (ICCRE)*, May 2024, pp. 98–103.

[35] S. Balaji and M. S. Murugaiyan, "Waterfall vs. V-model vs. agile: A comparative study on SDLC," *Int. J. Inf. Technol. Bus. Manage.*, vol. 2, no. 1, pp. 26–30, 2012.

[36] N. H. Garcia and A. Wortmann, "Survey on robotic systems integration," in *Proc. IEEE/ACM 5th Int. Workshop Robot. Softw. Eng. (RoSE)*, May 2023, pp. 9–16.

[37] A. Gupta, "Comparative study of different SDLC models," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 9, no. 11, pp. 73–80, Nov. 2021.

[38] Q. Yas, A. Alazzawi, and B. Rahmatullah, "A comprehensive review of software development life cycle methodologies: Pros, cons, and future directions," *Iraqi J. Comput. Sci. Math.*, vol. 4, no. 4, pp. 173–190, Nov. 2023.

[39] S.-K. Kim, "Strategic decision spectrum for software engineering," in *Proc. IEEE Int. Conf. Ind. Eng. Eng. Manage. (IEEM)*, Dec. 2023, pp. 1708–1712.

[40] E. Mendez-Flores, A. Ortiz, I. Macias, and A. Molina, "Experimental validation of an enhanced MPPT algorithm and an optimal DC–DC converter design powered by metaheuristic optimization for PV systems," *Energies*, vol. 15, no. 21, p. 8043, Oct. 2022.

[41] E. Mendez-Flores, I. Macias-Hidalgo, and A. Molina, "Solar irradiation changes detection for photovoltaic systems through ANN trained with a metaheuristic algorithm," in *Metaheuristics in Machine Learning: Theory and Applications* (Studies in Computational Intelligence). Cham, Switzerland: Springer, 2021, pp. 711–744.

[42] P. Ponce-Cruz, A. M. Gutiérrez, R. A. Ramírez-Mendoza, E. M. Flores, A. A. O. Espinoza, and D. C. B. Silva, *A Practical Approach to Metaheuristics Using LabVIEW and MATLAB®*. London, U.K.: Chapman & Hall, 2020.

[43] K. Forsberg and H. Mooz, "The relationship of system engineering to the project cycle," *INCOSE Int. Symp.*, vol. 1, no. 1, pp. 57–65, Oct. 1991.

[44] L. Ullrich, M. Buchholz, K. Dietmayer, and K. Graichen, "Expanding the classical V-model for the development of complex systems incorporating AI," *IEEE Trans. Intell. Vehicles*, early access, Jul. 29, 2024, doi: 10.1109/TIV.2024.3434515.

[45] J. Parri, F. Patara, S. Sampietro, and E. Vicario, "A framework for model-driven engineering of resilient software-controlled systems," *Computing*, vol. 103, no. 4, pp. 589–612, Apr. 2021.

[46] M. D. Xames and T. G. Topcu, "A systematic literature review of digital twin research for healthcare systems: Research trends, gaps, and realization challenges," *IEEE Access*, vol. 12, pp. 4099–4126, 2024.

[47] P. Ponce, E. Mendez, and A. Molina, "Teaching fuzzy controllers through a V-model based methodology," *Comput. Electr. Eng.*, vol. 94, Sep. 2021, Art. no. 107267.

[48] H. Zhu, S. Wilson, and E. Feron, "The design, education and evolution of a robotic baby," *IEEE Trans. Robot.*, vol. 39, no. 3, pp. 2488–2507, Jun. 2023.

[49] J. Gausemeier and S. Moehringer, "VDI 2206—A new guideline for the design of mechatronic systems," *IFAC Proc. Volumes*, vol. 35, no. 2, pp. 785–790, Dec. 2002.

[50] P. Ponce, O. Mata, A. Molina, E. Perez, and M. S. Ramírez, "An improved design methodology for reconfigurable robots during COVID 19 pandemic at university: Robocov case study at tecnologico de monterrey," in *Proc. Mach. Learn.-Driven Digit. Technol. Educ. Innov. Workshop*, Dec. 2021, pp. 1–8.

[51] A. F. Prince, B. Vodermayer, B. Pleintinger, A. Kolb, G. Franchini, E. Staudinger, E. Dietz, S. Schröder, S. Frohmann, F. Seel, and A. Wedler, "Modular mechatronics infrastructure for robotic planetary exploration assets in a field operation scenario," *Acta Astronautica*, vol. 212, pp. 160–176, Nov. 2023.

[52] D. Piette, "Driving electrical systems design from systems modeling," SIEMENS, 2020. [Online]. Available: https://blogs.sw.siemens.com/teamcenter/driving-electrical-systems-design-from-systems-modeling/

[53] T. Takayama and N. Hisamatsu, "Worm gear mechanism with switchable backdrivability," *ROBOMECH J.*, vol. 6, no. 1, pp. 1–10, Dec. 2019.

[54] R. Erickson and D. Maksimović, *Fundamentals of Power Electronics*. Cham, Switzerland: Springer, 2020, doi: 10.1007/978-3-030-43881-4.

[55] T. Nguyen-Duc, H. Nguyen-Duc, T. Le-Viet, and H. Takano, "Single-diode models of PV modules: A comparison of conventional approaches and proposal of a novel model," *Energies*, vol. 13, no. 6, p. 1296, Mar. 2020.

[56] G. Wang, K. Zhao, T. Qiu, X. Yang, Y. Zhang, and Y. Zhao, "The error analysis of the reverse saturation current of the diode in the modeling of photovoltaic modules," *Energy*, vol. 115, pp. 478–485, Nov. 2016.

[57] H. Bellia, R. Youcef, and M. Fatima, "A detailed modeling of photovoltaic module using MATLAB," *NRIAG J. Astron. Geophys.*, vol. 3, no. 1, pp. 53–61, Jun. 2014.

[58] (Jan. 2021). *P108 Solar Panel Datasheet, Voltaic Systems®*. [Online]. Available: https://voltaicsystems.com/9-watt-18v-panel-etfe/#:~:text=Features-,Technical%20Drawing,-Waterproof%20(IP67)

[59] Solar Energy & Solar Power. (2025). *Tracking Clean Energy Progress 2023*. [Online]. Available: https://www.solarenergylocal.com/states/california/irvine/

[60] E. Mendez, A. Ortiz, P. Ponce, I. Macias, D. Balderas, and A. Molina, "Improved MPPT algorithm for photovoltaic systems based on the earthquake optimization algorithm," *Energies*, vol. 13, no. 12, p. 3047, Jun. 2020.

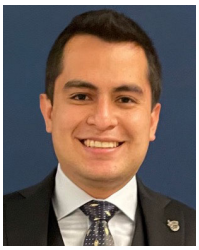
[61] S. I. Palomino-Resendiz, D. A. Flores-Hernández, L. A. Cantera-Cantera, N. Lozada-Castillo, and A. Luviano-Juárez, "Design and implementation of model-based predictive control for two-axis solar tracker," *Sol. Energy*, vol. 265, Nov. 2023, Art. no. 112080.

[62] S. Racharla and K. Rajan, "Solar tracking system—A review," *Int. J. Sustain. Eng.*, vol. 10, no. 2, pp. 72–81, 2017.

[63] K. Kumba, S. P. Simon, K. Sundareswaran, P. S. R. Nayak, K. A. Kumar, and N. P. Padhy, "Performance evaluation of a second-order lever single axis solar tracking system," *IEEE J. Photovolt.*, vol. 12, no. 5, pp. 1219–1229, Sep. 2022.

[64] I. Abadi, A. Soeprijanto, and A. Musyafa, *Design of Single Axis Solar Tracking System at Photovoltaic Panel Using Fuzzy Logic Controller*. Edison, NJ, USA: IET, 2014, pp. 2–4. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7120264

[65] S. I. Palomino-Resendiz, F. A. Ortiz-Martínez, I. V. Paramo-Ortega, J. M. González-Lira, and D. A. Flores-Hernández, "Optimal selection of the control strategy for dual-axis solar tracking systems," *IEEE Access*, vol. 11, pp. 56561–56573, 2023.
[66] Y. Away, A. Novandri, I. Raziah, and M. Melinda, "A new technique for improving the precision of dual-axis tetrahedron-based sensor with switching between PID and ANN," *IEEE Access*, vol. 11, pp. 89138–89151, 2023.
[67] (Dec. 2024). *5203-2402-0005 Spec Sheet, ServoCity®*. [Online]. Available: https://www.servocity.com/content/spec_sheets/5203-2402-0005_spec_sheet.pdf
[68] C. P. Lyman, *Hibernation and Torpor in Mammals and Birds*. Amsterdam, The Netherlands: Elsevier, 2013.
[69] H. Deboucha, M. Kermadi, S. Mekhilef, and S. L. Belaid, "Ultra-fast and accurate MPPT control structure for mobile PV system under fast-changing atmospheric conditions," *IEEE Trans. Sustain. Energy*, vol. 14, no. 4, pp. 2168–2176, Oct. 2023.



EFRAIN MENDEZ-FLORES (Member, IEEE) received the B.S. degree in mechatronics engineering with a robotics concentration and the Ph.D. degree in engineering sciences from Tecnológico de Monterrey, Mexico, in 2016 and 2021, respectively. He also completed the advanced Training in power electronics, completing a specialization from the University of Colorado, Boulder, in 2021.

He is currently a Postdoctoral Researcher with the Robot Ecology Laboratory, University of California, Irvine. His recent work includes leading the development of the bio-inspired "RaccoonBot" for persistent environmental sensing and the redesign of the robotic prop "Jean Luc" for theatrical applications. He has authored multiple publications in peer-reviewed journals and conferences and co-authored the book *A Practical Approach to Metaheuristics Using LabVIEW and MATLAB*. His research interests include robotics, mechatronics, embedded systems, metaheuristic optimization algorithms, and power electronics, with a focus on environmental monitoring applications and sustainable energy solutions.

Dr. Mendez-Flores is a recognized level 1 member of the National System of Investigators (SNI) in Mexico and has received multiple awards for his contributions, including the Best Oral Presentation Awards at ICMEAE and ICCRE conferences.

DIANA MORALES received the B.S. degree in electrical engineering and applied mathematics from California State Polytechnic University, Pomona, in December 2023. She is currently pursuing the Ph.D. degree in electrical engineering with the Robot Ecology Laboratory, Henry Samueli School of Engineering, University of California, Irvine. Previously, she has contributed to multiple projects at LLNL, developing math models to capture population dynamics and developing a coupled electromagnetic-thermal beamformed model for microwave volumetric additive manufacturing. At LBNL, she developed signal-processing algorithms for different FMCW radars, leading to her first IEEE conference publication. At ASU, she designed a Spectrometer for the Completely Hackable Amateur Radio Telescope NSF-funded project, still used today. Her recent work includes sensor functionality and integration, ensuring the bio-inspired "RaccoonBot" can perform persistent environmental monitoring. Her research interests include control theory, sensor fusion, and collaboration of heterogeneous systems for the exploration of unknown environments using machine learning.

MAGNUS EGERSTEDT (Fellow, IEEE) received the B.A. degree in philosophy from Stockholm University, and the M.S. degree in engineering physics and the Ph.D. degree in applied mathematics from the Royal Institute of Technology, Stockholm, Sweden.

He is currently the Dean of Engineering and a Professor with the Department of Electrical Engineering and Computer Science, University of California, Irvine. Prior to joining UCI, he was on the faculty with Georgia Institute of Technology, serving as the Chair of the School of Electrical and Computer Engineering and the Director for Georgia Tech's Institute for Robotics and Intelligent Machines. He was a Postdoctoral Scholar with Harvard University. He conducts research in the areas of control theory and robotics, with a particular focus on the control and coordination of multi-robot systems.

Dr. Egerstedt is a fellow of IFAC, a member of the Royal Swedish Academy of Engineering Science, and served as the President of the IEEE Control Systems Society. He has received a number of teaching and research awards, including the Ragazzini Award, the O. Hugo Schuck Best Paper Award, and the Alumni of the Year Award from the Royal Institute of Technology.