IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 17, 2021, accepted October 14, 2021, date of publication November 2, 2021,
date of current version November 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3124951

A Modular Reconfigurable Robot for Future Autonomous Extraterrestrial Missions

ALEXEY M. ROMANOV[1], **(Senior Member, IEEE)**,
VLADIMIR D. YASHUNSKIY[1], **(Member, IEEE)**,
AND WEI-YU CHIU[2], **(Member, IEEE)**
[1]Institute of Cybernetics, MIREA—Russian Technological University, 119454 Moscow, Russia
[2]Department of Electrical Engineering, National Tsing Hua University, Hsinchu 300044, Taiwan

Corresponding author: Alexey M. Romanov (romanov@mirea.ru)

The work of Alexey M. Romanov was supported by the Ministry of Science and Higher Education of the Russian Federation under Grant FSFZ-2020-0019. The work of Wei-Yu Chiu was supported by the Ministry of Science and Technology of Taiwan under Grant MOST 110-2221-E-007-097-MY2.

ABSTRACT This study proposes a heterogeneous modular reconfigurable robot called SABER (short for Step, Assembler, and Bridge Explorer Robot) that is suitable for future autonomous extraterrestrial missions. SABER comprises a platform and a reconfigurable rail and can operate in monowheel, rail trolley, and manipulator configurations. The monowheel configuration provides locomotion at a top speed of 10 km/h. The rail trolley configuration allows the robot to traverse gaps whose length is greater than its wheel diameter, climb steps whose height is no taller than the wheel radius, and pass through passages narrower than the wheel diameter. The manipulator configuration enables the robot to manipulate objects using two robotic arms with changeable tools. This design allows several SABERs to work as a team to enlarge their working area or enhance their mechanical capabilities. As such, SABERs can undertake various tasks in extraterrestrial missions, including exploration, outpost building and equipment maintenance. The key feature of the SABER is its mechanical transmission that allows its platform to move rapidly along a rail made from the manipulators of the robot. We also formulated the novel concept of a jamming safety diagram that is used to determine controller precision requirements to prevent the transmission from jamming. Finally, a new positioning controller meeting those requirements was introduced and implemented as field-programmable-gate-array firmware that is compatible with a wide range of low-cost integrated circuits. In numerical simulations, we evaluated SABER with respect to its speed, acceleration, oscillation, and ability to negotiate gaps and steps; SABER performed as well as or better than existing autonomous mobile robots and modular reconfigurable robots.

INDEX TERMS Mobile robots, mechatronics, mechanical engineering, mechanical power transmission, reconfigurable architectures, actuators, robot motion, robot kinematics, motion control, field programmable gate arrays (FPGA).

I. INTRODUCTION

Interplanetary space missions had long been considered the purview of the most powerful nations, but recent technological advances have opened this domain to private enterprises. In particular, studies have indicated the commercial potential of exploring Mars or the Moon [1]. However, interplanetary travel and prolonged stays on another planet are hazardous [2], and housing people on space bases is resource intensive [3], [4]. Thus, it is much safer and more efficient to build outposts on other planets or the Moon using autonomous robot swarms.

A wide variety of space rovers have been used in missions since the Moon landing of Lunokhod 1 in 1970 [5]. However, space robots have been specifically designed for their mission and cannot be reused for a different task. Currently, precise long-term planning is primarily used to circumvent this limitation [6]. This approach may be suitable for the scientific endeavors of governments, but businesses seeking to capitalize on space-oriented commercial activities will require more flexible solutions.

Reusability is currently the most promising concept in the space industry, and it promises to greatly reduce the cost

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

of a single spacecraft launch [7]. Reusability can also be implemented in rover design: a rover that reaches another planet should be capable of being repurposed to, for example, establish and maintain facilities for research or production using different tools, load and unload cargo from incoming ships, and explore the area around the outpost. Such rovers should leverage the versatility of modern assembly-line techniques and industrial maintenance robots [8], [9] in combination with autonomous all-terrain capabilities for extraterrestrial exploration [10]–[12].

This study proposes a new heterogeneous modular reconfigurable robot (MRR) design, which includes two manipulators with seven degrees of freedom (DOFs) that are suitable for industrial assembly applications, such as those considered in [9] and [13]. For exploration, our robot can transform to a monowheel configuration, providing locomotion comparable with those of wheel-legged robots [12], [14], [15]. To traverse steps and gaps, our robot can create a bridge from its own manipulator. We call our robot SABER, which is short for *Step, Assembler, Bridge, Explorer Robot*. The key feature of SABER is its mechanical transmission that allows its platform to move rapidly along a rail made from the manipulators of the robot. To ensure the robust operation of the transmission, we constructed a novel jamming safety diagram that can be used to prevent collisions and determine the requirements for controller precision. Finally, we designed a positioning controller that met these requirements; the controller was also compatible with field programmable gate array (FPGA) firmware and made from a wide variety of low-cost integrated circuits.

Our main contributions are as follows. The first is SABER, which features a monowheel made from two manipulators; the use of the same structure for both manipulation and locomotion; a stable platform for sensors (e.g., LIDAR, GPS, and cameras) and a payload; the capability to manipulate objects, move on flat surfaces, climb steps no higher than the robot's wheel radius, traverse gaps larger than the robot's wheel diameter, and navigate passages narrower than the wheel diameter. The second contribution is our proposed structure of a mechanical transmission, the solution to an inverse kinematics problem, and formulation of a motion control law for the proposed transmission. These allow SABER to move a carriage along a rigid polygonal roller chain against gravity. The roller chain can be used as a reconfigurable rail, the application of which is based on a robot kinematic scheme that enables a speed of 10 km/h to be achieved for a 12.7-mm-step roller chain. The third contribution is our novel concept of a jamming safety diagram for determining controller precision requirements, which guarantees the absence of transmission jamming in any configuration of SABER's rail. The fourth contribution is our formulation of a control law that was implemented as FPGA firmware; it is compatible with a wide range of low-cost reconfigurable integrated circuits. We verified our firmware in a cycle-accurate firmware-in-loop simulation, demonstrating that it can provide the desired precision when synchronous motors available on the market are used.

The remaining parts of this paper are organized as follows. Section II presents a review of related works. Section III details the design of SABER. Section IV describes the multiaxis linear drive and corresponding control law, which are key to implementing SABER. Section V presents the concept underlying the control system. Section VI details the evaluation experiments. Finally, Section VII summarizes and concludes the study.[1]

II. RELATED WORK

Robots for extraterrestrial exploration and outpost building must navigate rough terrain because they are generally deployed to an unprepared site. Wheeled robots [17], tracked robots [18], and walking robots [19] are the most types of all-terrain robots developed.

Wheeled robots are similar to industrial automated guided vehicles (AGVs) [20] or maintenance robots [8], but their rover suspension mechanism is optimized for negotiating regoliths [17], [21]. The highest rover speed is currently 1.8 km/h (0.5 m/s) [21]. Tracked robots adhere well with the surface they are traveling on; however, they cannot negotiate sandy soil well, which is found on Mars' surface: in the ExoMars project, a tracked rover was found to consume up to 65% more power than a wheeled rover [22]. Walking robots mimic the locomotion of animals by moving their legs. Such robots move slower than wheeled robots but negotiate obstacles better. For example, a six-legged walking robot proposed in [19] has a maximum speed less than 1 km/h. A walking robot also requires complex control algorithms that must be processed in real time, making them more difficult to implement relative to wheeled and tracked robots [22].

Reconfigurable robots can switch between wheeled, tracked, and legged configurations, allowing them to possess the advantages of the three aforementioned robot types. A self-balancing two-wheel mobile platform moving at a speed up to 7.2 km/h was proposed in [23]. The platform was equipped with four modules that can transform between a wheel, leg, and track, which make it capable of negotiating, for example, stairs that are no higher than 40 cm, trenches that are no wider than 50 cm, and slopes that are no steeper than 40°. Another robot called Scorpio [24] can switch between a monowheel rolling configuration to reach a speed of 1.5 km/h and a four-legged walking configuration, but it lacks a stable platform that is suitable for transporting payloads or for housing a sensor.

Wheel-legged robots, which combine the capabilities of wheels and tracks but not legs, are more common than reconfigurable robots [25]. Wheel-legged robots can reach a high speed and can climb steps and traverse gaps smaller than itself. For example, a wheel-legged robot in [15] can achieve

[1] A concise version of the research results was reported in the 2021 IEEE 17th International Conference on Automation Science and Engineering and published in [16].

a cruising speed of 1.08 km/h and top speed of 3.88 km/h. Moreover, during vehicle locomotion, legs can be used as an active suspension system [26]. A wheel-legged design was further used to improve the cross-country capabilities of a planetary exploration rover [12].

Some robots change their configuration by dynamically changing their width or length, and this design has been applied to a wheeled robot [11] and tracked robot [27]. Generally, the robot increases its length to overcome larger gaps and increase its width to drive above obstacles instead of overcoming them. Conversely, the robot decreases its size to be more manoeuvrable and capable of negotiating narrow passages.

Recent advances in multirotor unmanned aerial vehicles (UAVs) have enabled novel designs of UAVs as reconfigurable robots [25]. Zhong *et al.* [28] proposed a triphibious robot with a tilting-rotor structure; the robot is equipped with four legs for walking and four more rotors for overcoming obstacles much larger than itself. However, this robot cannot function on the Moon because it can only function in an extraterrestrial environment with a dense atmosphere.

Most existing robots have neither arms nor manipulator. Even some parts of the robots, such as their wheels, may be transformed into short arms, which can be used for excavation or simple manipulation; these arms are generally unsuitable for assembling parts or maintaining equipment. Recent studies [29]–[31] have demonstrated that robots must be equipped with at least two 6-8 DOF manipulators to effectively replace humans in most operations, including assembly and maintenance. Notably, manipulators can be mounted on most of the aforementioned mobile platforms at the cost of a larger and heavier robot.

MRRs are a type of reconfigurable robot for extraterrestrial exploration [32]–[34]. MRRs can move itself to a region of interest and then undertake production or maintenance operations. Compared with a traditional design featuring a mobile platform and set of manipulators, the MRR's design offers the following benefits: a reduction in size because hardware can be reused, an improvement in the ability to overcome obstacles, the ability for several robots to be combined into one, and the accommodation of a larger working area. In addition, the MRR has the unique capability of recovering itself after failures by replacing its own faulty parts by cannibalizing other robots or even building new kinematic structures for locomotion that do not use faulty parts [35]. This key feature makes the MRR more operationally robust because repairs (which often require the delivery of spare parts) are highly complicated in extraterrestrial missions.

MRRs can be homogeneous or heterogeneous. Chained homogeneous MRRs, such as MTRAN-III, PolyBot-G3, and Morpho [36], can form multi-DOF manipulators and move by changing their shape into a wheeled or spider-shaped structure. This approach results in almost no overhead on structures, but has the disadvantage of low locomotion speed. For example, the maximum land speed of the SMORES robot is only 0.4 km/h (1.1 body length/s) [37], [38]. One of the fastest homogeneous MMRs is PolyBot-G3, which can roll itself up to take on the shape of a thread to move at up to 5.76 km/h (1.6 m/s) [39].

However, constraints on motor power and module size markedly limit the applications of homogeneous MRRs on planatery surfaces. Typically, robotic manipulators are built using motors of various sizes from larger at the beginning effector to smaller at the end effector. This enables the operation of lighter robots that have manipulators with more DOFs. Because the modules of a homogeneous MRR are identical, each module can lift no more than three other modules on Earth. For example, developers of the M-TRAN robot assumed that one module of their robot has sufficient torque to lift one other module in any posture [40]. Nonetheless, this disadvantage is less pronounced (and even nonexistent) in low gravity (or zero gravity) situations, making homogeneous MRRs more attractive for solving practical tasks [41].

By contrast, heterogeneous MRRs, which are composed of modules of different types, can be better tailored to the physical constraints of their environment than their homogeneous counterparts [36]. With regard to manipulation, this heterogeneity allows a hierarchy of modules to be defined and different drive sizes to be used [13].

III. DESIGN OF SABER

This section presents the design of our heterogeneous MRR, SABER. It comprises a reconfigurable rail and a platform sliding along this rail (Fig. 1). The robot can function as a monowheel, a manipulator, or a rail trolley. The reconfigurable rail is formed by 2-DOF motorised modules with locks on all joints. These locks can fix the joints in various positions and do not consume power when locking is underway [42]. The boundary modules are equipped with genderless connectors similar to the ones introduced in [13], [43]. This combination renders several configurations possible, including the monowheel [Fig. 1(a)] and two-manipulator [Fig. 1(b)] configurations. The genderless connectors are used to equip those manipulators with changeable tools and to lock the rail in a closed shape. The reconfigurable rails of several modules can be connected with each other using genderless connectors to form a longer rail that can be used by the modules' platforms [Fig. 1(d)].

The reconfigurable rail stays rigid during locomotion, which is powered by the motor installed on the platform. In a monowheel configuration, the robot acts as hubless monowheel [44] and executes turns using the methods in [45]. In the rail trolley configuration, the platform can position itself on the rail, using the same drive that is used in the monowheel configuration and using the same balancing system, to remain on top of the rail. This rail configuration enables the robot to traverse gaps greater than the robot's diameter in a monowheel configuration, to climb steps and greatly reduce its height, and to pass underneath a low passage. Unlike the platform introduced in [24] that turns by inclining itself at a limited angle, our design accommodates

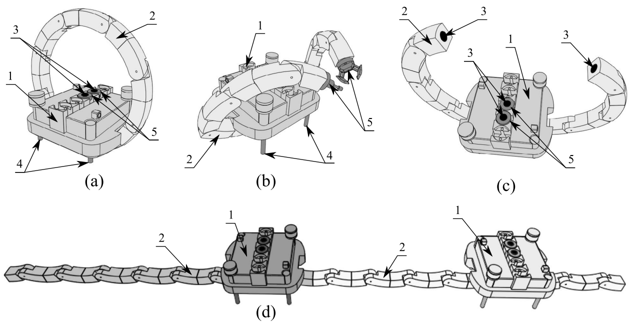


FIGURE 1. Overview of SABER. (a) Monowheel configuration, (b) manipulator configuration, (c) transition from monowheel to manipulator configuration, (d) two robots connected in a rail trolley configuration. 1. Main platform, 2. Reconfigurable rail, 3. Genderless connectors; 4. Extendable support leg, 5. Tools and actuators.

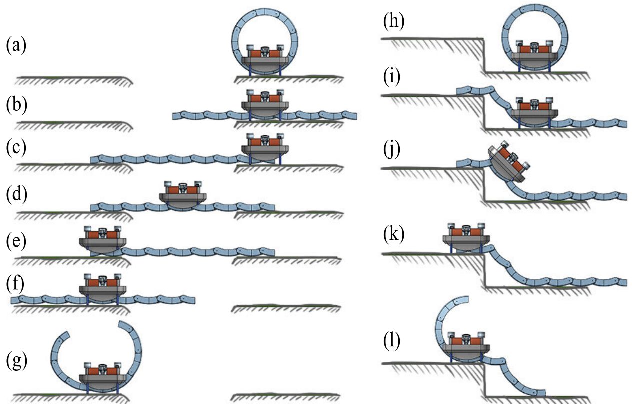


FIGURE 3. SABER traversing a gap, illustrated in (a)–(g), and a step, illustrated in (h)–(l).

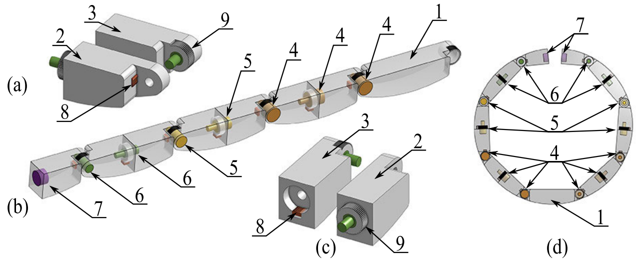


FIGURE 2. Reconfigurable rail structure. (a) Pair of units, forming a crosswise joint, (b) half of the rail, connected to a base module, (c) pair of units, forming a lengthwise joint, (d) full rail, almost closed. 1. Base unit, 2. Unit with lengthwise drive and crosswise lock, 3. Unit with crosswise drive and lengthwise lock, 4. Large motor; 5. Medium motor, 6. Small motor; 7. Genderless connector, 8. Lock, 9. Geared surface, forming the opposite side of the lock.

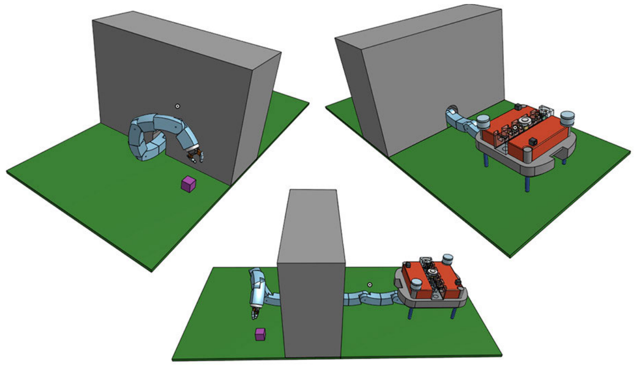


FIGURE 4. SABER reaching a distant object using an extended manipulator.

a payload and the robot's tools and navigation sensors, such as cameras and LIDARs.

Fig. 2 details the placement of the drives, locks, and genderless connectors of the reconfigurable rail. The reconfigurable rail's module comprises two units that form the diagonal and lengthwise DOFs. As in the case in the manipulator configuration, the central part of the reconfigurable rail is under the platform; because the central lengthwise DOF is not required, two reconfigurable modules jointly form the base unit.

To illustrate the advantages of our MRR design, we consider three scenarios. The first features the robot handling a payload in the monowheel configuration. The support legs simplify the loading process, and the wheel whose diameter is comparable with the robot's dimensions provides the robot with a high speed and the capability to negotiate small obstacles, such as on-floor cable channels or potholes. This allows a single robot to transport materials not only indoors but also outdoors between different production facilities. The second scenario involves through gaps and steps (Fig. 3). The third scenario involves the robot reaching out to grab a distant object using extended manipulators, which is relevant to situations where the robot cannot be close to a piece of equipment it is to maintain or inspect. The extended manipulator can be created by moving the reconfigurable rail toward one of the rail's ends (Fig. 4) and equipping the other end with tool for the relevant application.

IV. ROLLER CHAIN MECHANICAL TRANSMISSION

With regard to hardware implementation, monowheels and robotic arms have been well investigated. This section thus focuses on the mechanical design and control law underlying SABER's transmission; these allow the platform to move along the rail and the robot to move in both wheel and rail configurations.

In [46], racks and spring loaded gears were used to move along a reconfigurable rail. Unfortunately, a rack-based construction does not allow for continual reconfiguration and the use of spring loaded gears for the supporting structure is suboptimal. To solve these problems, we propose replacing the racks and gears with a roller chain and set of sprockets. Fig. 5 illustrates our design featuring a roller chain fastened along each module. Two sprocket modules, placed at the front and back of the platform, ensure that the platform is fixed relative to the modules. Each module comprises two sprockets, one above and the other below the roller chain. Fig. 6 indicates that the whole module can rotate, providing the most favorable angle between the roller chain and the sprockets.

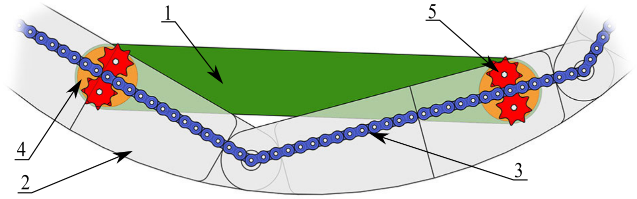


FIGURE 5. Drives on platform. 1. Platform, 2. Modules, 3. Roller chain, 4. Sprocket module, 5. Sprocket.

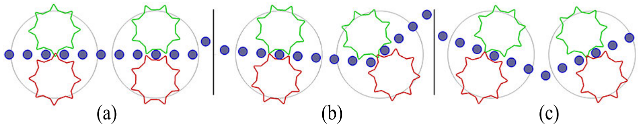


FIGURE 6. Rotation of sprocket modules (a) on a straight part of the roller chain, (b) right after passing the edge, and (c) passing the edge.

The use of an eight-cog sprocket enables positioning at angles from 135° to 225°. To prevent bending moments, roller chains and sprocket modules are placed on both side of the robot.

To implement the sprocket module, we propose a cheeseburger drive (CB drive; Fig. 7). The entire module is rotated by a drive, as depicted in Fig. 7(a). A geared ring transmits the rotation from the outer gear on the second drive to an inner gear on the sprocket axis at any module rotation angle, as illustrated in Fig. 7(b). As illustrated in Fig. 7(c), equal gears on the sprocket axis synchronize the sprockets' rotation. The opposite bevel of the sprockets' cogs allows the sprockets to overlap with each other.

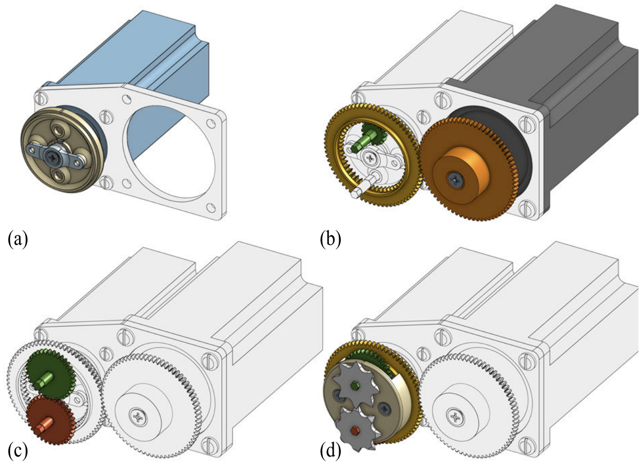


FIGURE 7. CB drive construction by layers. (a) Rear half of case with module-rotating drive, (b) Sprocket-rotating drive and gear transmission up to the sprocket axis, (c) Sprocket axis with gears, (d) Entire assembly with sprockets.

Our robot, based on the aforementioned design, possesses the following features.

1) The platform can move along the roller chains in all configurations when the angles of all modules are between 135° and 225°.

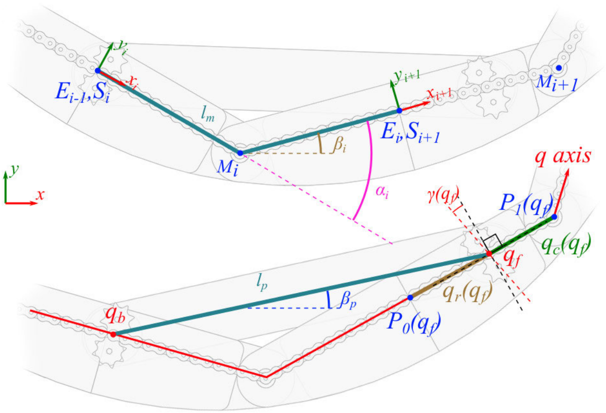


FIGURE 8. Symbols associated with parts of roller chain.

2) When the platform carries the weight of the rail, the support and transmission are provided by the lower sprockets. When the rail carries the weight of the platform, the support and transmission are provided by the upper sprockets.

3) The transmission of roller chains and sprockets produces high force;

4) A straight-linear module configuration leads to straight-linear roller chains.

Fig. 8 summarizes our notation that we adopt in the following discussion. SABER's characteristics are described by the following parameters: the number of modules N, where $N = 8$; the length of each module l_m, which is equal to the distance between two crosswise axis in Fig. 2; the distance l_p between the front and rear CB drives; the CB drive inner gear ratio U; and the roller chain step l_c. The robot's entire state can be described by an array containing 1) the modules' angles α_i, where i represents the module ID, and 2) the position q_f containing the front sprocket module on the q-axis, where the q-axis is defined as a polyline starting at the initial position of the first module (whose index is $i = 0$) and passing along the roller chain. In this study, the index i of a given module is determined by its position on the q-axis and the length of each module l_m:

$$i(q) = \left\lfloor \frac{q}{l_m} \right\rfloor. \tag{1}$$

The positions of all modules are expressed in Cartesian coordinates. The ith module in its local coordinates can be described by the initial point $\vec{s}_i$, midpoint $\vec{m}_i$, and endpoint $\vec{e}_i$:

$$\begin{aligned} \vec{s}_i &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \vec{m}_i &= \vec{s}_i + \begin{bmatrix} \frac{l_m}{2} \\ 0 \end{bmatrix} \\ \vec{e}_i &= \vec{m}_i + \vec{r}(\begin{bmatrix} \frac{l_m}{2} \\ 0 \end{bmatrix}, \alpha_i), \end{aligned} \tag{2}$$

where the rotation function $\vec{r}$ is defined as

$$\vec{r}(\vec{p},\alpha) = \begin{bmatrix}\cos(\alpha) & -\sin(\alpha)\\ \sin(\alpha) & \cos(\alpha)\end{bmatrix}\begin{bmatrix}p_x\\ p_y\end{bmatrix}. \tag{3}$$

The initial point, midpoint, and endpoint in global coordinates are denoted $\vec{S}_i$, $\vec{M}_i$ and $\vec{E}_i$, respectively. When the modules are connected in a chain, the endpoint of the next module coincides with the initial point of the subsequent module, and the rotation angle is cumulative from module to module as the global rotation angle β_i. Thus, the global coordinates can be determined after i iterations from 0 to N by

$$\begin{aligned}\vec{S}_i &= \vec{r}(\vec{s}_i,\beta_{i-1}) + \vec{E}_{i-1}\\ \vec{M}_i &= \vec{r}(\vec{m}_i,\beta_{i-1}) + \vec{E}_{i-1}\\ \vec{E}_i &= \vec{r}(\vec{e}_i,\beta_{i-1}) + \vec{E}_{i-1}\\ \beta_i &= \beta_{i-1} + \alpha_i,\end{aligned} \tag{4}$$

where $\beta_{-1} = 0$ and $\vec{E}_{-1} = (0, 0)$.

As indicated in Fig. 6, the CB drive must stay perpendicular to the roller chain most of the time. To provide smooth continuous rotation, the CB drive must start rotating before reaching the center of the module. To determine the CB drive rotation, we must know the index of the module i at its given location, the relative coordinate of this module q_r (from the module's initial point) and the coordinate of the drive relative to the center of the module q_c; the latter two are defined as

$$\begin{aligned}q_r(q) &= q \bmod l_m\\ q_c(q) &= q_r(q) - \frac{l_m}{2}.\end{aligned} \tag{5}$$

If the rotation begins at an offset of q_o from the center, the transition angle $\gamma(q)$ can be calculated as

$$\gamma(q) = \begin{cases} 0, & \text{if } |q_c(q)| > q_o\\ \dfrac{\alpha_{i(q)}(cos(\frac{q_c(q)}{q_o}\pi)+1)}{4}, & \text{if } q_c(q) < 0\\ -\dfrac{\alpha_{i(q)}(cos(\frac{q_c(q)}{q_o}\pi)+1)}{4}, & \text{otherwise.}\end{cases} \tag{6}$$

The CB drive global orientation angle $\eta(q)$ can be expressed as

$$\eta(q) = \begin{cases}\eta_n(q) - 2\pi, & \text{if } \eta_n(q) > \pi\\ \eta_n(q), & \text{otherwise,}\end{cases} \tag{7}$$

where

$$\eta_n(q) = \gamma(q) + \begin{cases}\beta(i(q)), & \text{if } q_c(q) < 0\\ \beta(i(q)+1), & \text{otherwise.}\end{cases} \tag{8}$$

The sprocket angle is defined as

$$\xi(q) = \frac{2\pi q}{l_c N}, \tag{9}$$

where q represents the position of the chain, and l_c represents the roller chain step.

The global coordinates of the point on the chain can be evaluated from the known global coordinates of the nearest module points described in (4). The nearest points $\vec{P}_0(q)$ and $\vec{P}_1(q)$ can be expressed as

$$\begin{aligned}\vec{P}_0(q) &= \begin{cases}\vec{S}_{i(q)}, & \text{if } q_c(q) < 0\\ \vec{M}_{i(q)}, & \text{otherwise}\end{cases}\\ \vec{P}_1(q) &= \begin{cases}\vec{M}_{i(q)}, & \text{if } q_c(q) < 0\\ \vec{E}_{i(q)}, & \text{otherwise.}\end{cases}\end{aligned} \tag{10}$$

The interpolation coefficient $a(q)$ can be evaluated as

$$a(q) = \begin{cases}2\dfrac{q_r(q)}{l_m}, & \text{if } q_c(q) < 0\\ 2\dfrac{q_c(q)}{l_m}, & \text{otherwise.}\end{cases} \tag{11}$$

From (10) and (11), the global coordinates of the point on the chain $\vec{P}(q)$ can be evaluated as

$$\vec{P}(q) = \vec{P}_0(q) + a(q)(\vec{P}_1(q) - \vec{P}_0(q)). \tag{12}$$

Given the position on the chain of the front CB drive q_f, the position of the back CB drive q_b must satisfy

$$\begin{cases} q_b \in \begin{cases}[0, q_f), & \text{if the reconfigurable rail is opened}\\ [q_f - l_m N, q_f), & \text{otherwise}\end{cases}\\ \left|\vec{P}(q_f) - \vec{P}(q_b))\right| = l_p\end{cases} \tag{13}$$

After the rear CB drive position q_b is determined, the global orientation of the platform β_p can be evaluated as

$$\beta_p(q_f) = \arctan 2(\vec{P}(q_f) - \vec{P}(q_b(q_f))). \tag{14}$$

The front CB drive rotation ϕ_{c_f} and rear CB drive rotation ϕ_{c_b} at q_f can be calculated as

$$\begin{aligned}\phi_{c_f}(q_f) &= \eta(q_f) - \beta_p(q_f)\\ \phi_{c_b}(q_f) &= \eta(q_b(q_f)) - \beta_p(q_f).\end{aligned} \tag{15}$$

The rotation of the entire module causes the rotation of the sprockets. The required rotation of the CB drive geared ring ϕ_s depends on the CB drive rotation ϕ_c, the sprocket angle ξ, and the CB drive inner gear ratio U as follow:

$$\phi_s(q) = \phi_c(q) - \xi(q)U. \tag{16}$$

The functions in (13)–(16) constitute an ideal control law for the mechanical transmission. This law takes the position of the front CB drive on the q-axis as an input. In the monowheel configuration, the q-axis lies along an N-gon. Thus, for any finite N, a uniform change in q_f results in a nonuniform change in the monowheel rotation angle α_w. Due to the reconfigurable rail's high mass, the nonuniform rotation of the wheel necessarily entails redundant accelerations and deceleration and should thus be avoided. To achieve uniform rotation, we first express the position of the front CB drive q_f as a function of the wheel's rotation angle α_w. This requires the introduction of auxiliary values. The angle of the wheel's rotation while passing one module α_m can be evaluated as

$$\alpha_m = \frac{2\pi}{N}. \tag{17}$$

From (17), the angle of the wheel relative to the initial point of the currently passing module's initial α_r can be evaluated as

$$\alpha_r = \alpha_w \bmod (\alpha_m). \tag{18}$$

The radius R_i of the circle inscribed in the roller chain in the N-gon R_i can be evaluated as

$$R_i = \frac{l_m}{2\tan(\frac{\alpha_m}{2})}. \tag{19}$$

From (17), (18), and (19), the relative position of the front CB drive q_{f_r} on the current module as a function of the wheel's rotation angle α_w can be expressed as

$$q_{f_r}(\alpha_w) = \begin{cases} l_m - R_i \tan(\alpha_m - \alpha_r), & \text{if } \alpha_r > \frac{\alpha_m}{2} \\ R_i \tan(\alpha_r), & \text{otherwise.} \end{cases} \tag{20}$$

From (20), the absolute position of the front CB drive q_f on the current module as a function of the wheel's rotation angle α_w can be expressed as

$$q_f(\alpha_w) = l_m \left\lfloor \frac{\alpha_w}{\alpha_m} \right\rfloor + q_{f_r}(\alpha_w). \tag{21}$$

With (21), we can use a uniform change of α_w as the input to obtain the required q_f values to, in turn, obtain the corresponding control law.

Because the transition angle in (6) entails high-valued second derivatives, we require high motor acceleration followed by immediate motor deceleration, which increase power consumption. For practical implementation, we propose removing the high frequency part of the motion control law by applying a T-point discrete Fourier transform and retaining the K lowest harmonics. The motion control law thus has the following form:

$$\begin{aligned} \Phi_k &= \sum_{t=0}^{T-1} (\phi_t - \phi_0 - \frac{\phi_T - \phi_0}{T} t) e^{-\frac{2\pi i}{T} kt} \\ \widetilde{\phi_t} &= \frac{1}{T} \sum_{k=0}^{K} \Phi_k e^{\frac{2\pi i}{T} kt} + \phi_0 + \frac{\phi_T - \phi_0}{T} t. \end{aligned} \tag{22}$$

The set of all points can be denoted $\widetilde{\phi_{i,t}}$ where i remains identical to the index of the module from 0 to $N-1$ and t remains identical to the index of the point within a module from 0 to $T-1$.

Because the motion control low is periodic for the monowheel configuration and the Fourier transform results in a periodic function, the values at the first and last point are equal. However, although the ideal control law is described using a continuous function, the application of the Fourier transform to its parts, which represents the corresponding individual modules, causes discontinuities in SABER's configurations that are not the monowheel one. To solve this problem, we incorporate the control law from (22) into a piecewise linear function that is tangent to the ideal control law function at the initial point and endpoint of each module; this results in a smooth connection between the curves. Using (1), we can evaluate the coordinates of the initial point q_b and endpoint q_e along q and the drive angle ϕ and its derivative ϕ' that correspond to these two points as follows:

$$\begin{aligned} q_b(q) &= i(q) l_m \\ q_e(q) &= q_b(q) + l_m \\ \phi_b(q) &= \phi(q_b(q)) \\ \phi_e(q) &= \phi(q_e(q)) \\ \phi'{}_b(q) &= \phi'(q_b(q)) \\ \phi'{}_e(q) &= \phi'(q_e(q)) \end{aligned} \tag{23}$$

Therefore, the picewise linear function $\lambda(q)$ is

$$\lambda(q) = \begin{cases} \phi_b(q) + \phi'{}_b(q) q_r(q), & \text{if } q_c(q) < 0 \\ \phi_e(q) - \phi'{}_e(q)(l_m - q_r(q)), & \text{otherwise.} \end{cases} \tag{24}$$

The blending function is defined by four points: the beginning of the fade in q_1, the end of the fade in q_2, the beginning of the fade out in q_3 and the end of the fade out in q_4 at an interval from 0 to l_m. With this notation, the blending function $\mu(q)$ is defined as

$$\mu(q) = \begin{cases} 0, & \text{if } q_r(q) < q_1 \\ \dfrac{1 - cos(\pi \frac{q_r(q) - q_1}{q_2 - q_1})}{2}, & \text{if } q_r(q) < q_2 \\ 1, & \text{if } q_r(q) < q_3 \\ \dfrac{1 + cos(\pi \frac{q_r(q) - q_3}{q_4 - q_3})}{2}, & \text{if } q_r(q) < q_4 \\ 0, & \text{otherwise.} \end{cases} \tag{25}$$

Because the motion control law $\widetilde{\phi}$ from (22) is described in terms of a set of discrete values, the functions from (24) and (25) must also be represented in this form. For the ith module, the T points of the required functions can be evaluated as follows:

$$\begin{aligned} \lambda_{i,t} &= \lambda(l_m(i + \frac{t}{T})) \\ \mu_{i,t} &= \mu(l_m(i + \frac{t}{T})) \end{aligned} \tag{26}$$

From (22) and (26), the motion control law with flattened edges $\widehat{\phi}$ can be evaluated as follows:

$$\widehat{\phi_{i,t}} = \widetilde{\phi_{i,t}} \mu_{i,t} + \lambda_{i,t}(1 - \mu_{i,t}) \tag{27}$$

A general overview of the edge flattering for the control law is illustrated in Fig. 9. The plot corresponds to the front CB drive moving along two modules: one bent at an angle of 135°, and the other bent at an angle of 225°. Illustrating the solution to the discontinuity problem, Fig. 10 presents the control law function at the point where the modules connect with each other. Without edge flattening, the filtered control law gap at the point where the modules connect with each other is approximately 1°, and the edge flattening makes the transition completely smooth.

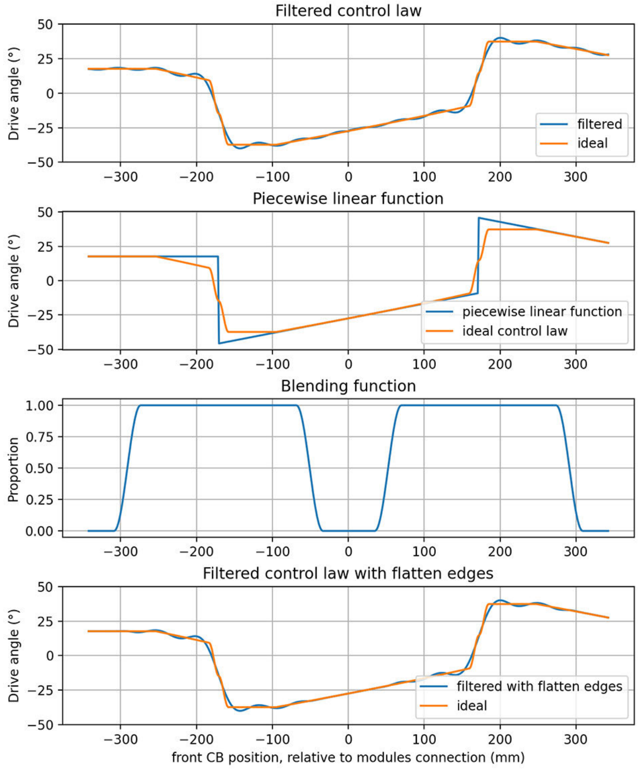


FIGURE 9. Obtaining a control law with flattened edges by incorporating a piecewise linear function into the filtered control law curve.

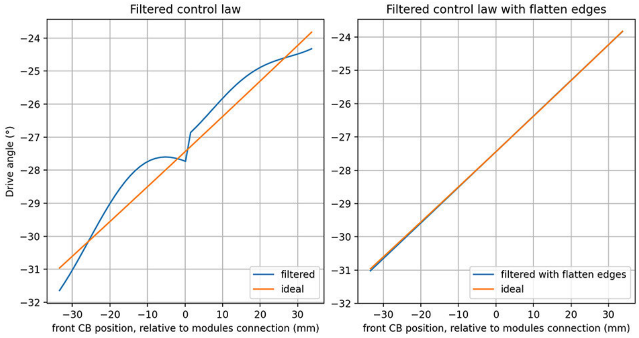


FIGURE 10. Filtered control law without and with edge flattening in the region where modules connect with each other.

V. CONTROL SYSTEM

To control SABER, we propose a two-level control system inspired by the hardware architecture in [47], [48]. A lower level (Fig. 11) comprises several FPGA-based control modules: one for each manipulator joint, one for each CB drive, one for controlling balancing mechanism, and one for controlling the extendable support legs. Generally, each FPGA controls two motors and the corresponding mechanical brakes and power distribution between modules [49]. The only exception is the FPGA module installed on the platform, which controls both the balancing mechanism and the four support legs. In addition, this FPGA performs additional tasks related to wired or wireless communication. Because a single FPGA can control robots with as many as six axes [50], such a division of labor between FPGAs is primarily defined by the location of the motors they control and motivated by the desire to simplify the wiring.

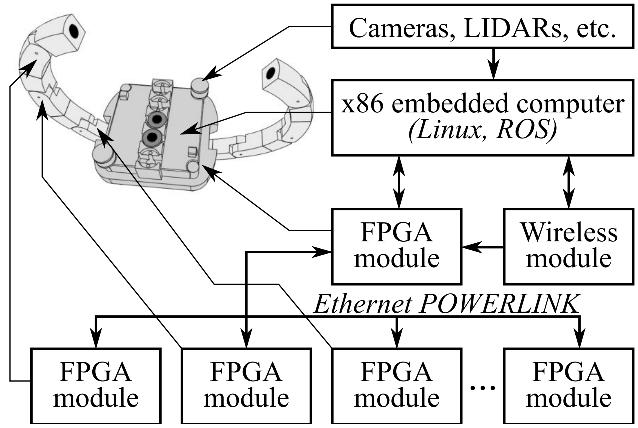


FIGURE 11. General structure of control system.

All FPGA-based modules are connected with each other using an Ethernet POWERLINK real-time network and act as controlled nodes, and an x86-based computer installed on the platform is running on the Robot Operating System (ROS) and used as a managing node [51], [52]. This computer implements a higher level control (Fig. 11) that covers functions such as path planning and computer vision and LIDAR processing.

Because our solution is based on the MRR control system architecture proposed in [47], several SABERs can be integrated to form one robot by connecting the genderless connectors in their manipulators. In this case, the x86-based computer of one of the SABERs becomes the active managing node and gains control of all FPGA-based modules that are connected to the joint Ethernet POWERLINK while the other SABERs remain passive until they are separated from each other. This design allows multiple SABERs to overcome large obstacles and reach distant objects by combining their rails and manipulators [Fig. 1(d)]. To communicate with other SABERs, each SABER is equipped with a synchronized Wi-Fi channel that is implemented in part on an x86 computer and in part on an FPGA; specifically, the computer generates data frames and the FPGA provides sufficient precise hardware synchronization using the method proposed in [53].

Control algorithms for robotic arm drives, monowheel balancing systems, linear support-leg drives, and power management systems have been extensively researched (e.g., in [45], [50], [54], and [49]). Thus, the rest of this section focuses on the implementation of the CB-drive control law introduced in Section IV.

Each CB drive includes two separate motors: one rotates the CB-drive module itself, and the other rotates the geared ring of the CB drive making platform move along the rail. To prevent the transmission from jamming, both of those motors should move synchronously following a set

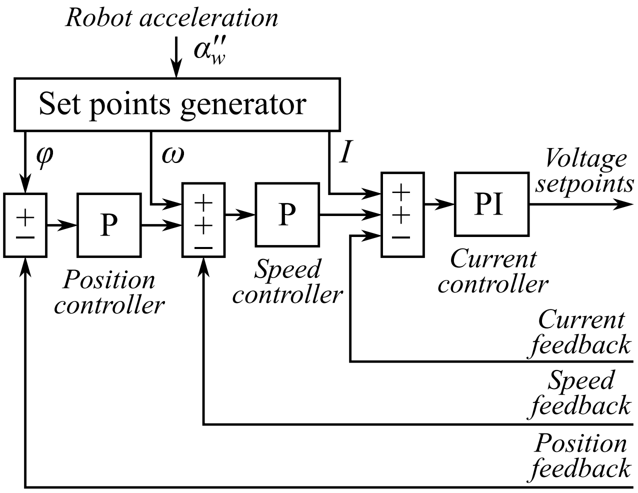


FIGURE 12. Three-level cascade P+P+PI-controller with feed-forward channels.

of predetermined points generated according to (27). This partially explains why both motors are controlled from one FPGA.

To achieve the desired precision, we propose a three-level cascade P+P+PI controller with feed-forward channels for speed and current (Fig. 12). The current controller can be implemented as a regular PI controller in cases where DC motors are used or as part of field oriented control in cases where a permanent magnet synchronous motor (PMSM) [55] is used.

The predetermined position, speed, and current of the CB drive are calculated using (28) in relation to the desired platform acceleration α''_w:

$$\begin{cases} \varphi_c = \widehat{\phi}_c(\alpha_w) \\ \omega_c = \alpha'_w \cdot \widehat{\phi}'_c(\alpha_w) \\ I_c = (\alpha''_w \cdot \widehat{\phi}'_c(\alpha_w) + (\alpha'_w)^2 \cdot \widehat{\phi}''_c(\alpha_w))\frac{J_c}{K_{t_c}} \\ \varphi_s = \widehat{\phi}_s(\alpha_w) \\ \omega_s = \alpha'_w \cdot \widehat{\phi}'_s(\alpha_w) \\ I_s = (\alpha''_w \cdot \widehat{\phi}'_s(\alpha_w) + (\alpha'_w)^2 \cdot \widehat{\phi}''_s(\alpha_w))\frac{J_s}{K_{t_s}}, \end{cases} \tag{28}$$

where φ_c and φ_s are the predetermined angular positions of the CB drive and its geared ring, respectively; ω_c and ω_s are the predetermined angular positions of the CB drive and its geared ring, respectively; I_c and I_s are the predetermined values of the currents of the CB drive and its geared ring motors, respectively; J_c and J_s are the moments of inertia of connections to the motor rotating the CB drive and the motor rotating the geared ring, respectively; and K_{t_c} and K_{t_s} are torque constants for the motor rotating the CB drive and the motor rotating the geared ring, respectively.

To achieve real-time performance $\widehat{\phi}_c(\alpha_w)$, $\widehat{\phi}_s(\alpha_w)$, $\widehat{\phi}'_c(\alpha_w)$, $\widehat{\phi}'_s(\alpha_w)$, $\widehat{\phi}''_c(\alpha_w)$ and $\widehat{\phi}''_s(\alpha_w)$ nonlinearities can be implemented using lookup tables. When $\widehat{\phi}_c(\alpha_w)$ and $\widehat{\phi}_s(\alpha_w)$ can be created directly from values calculated using (27), evaluating the rest of lookup tables requires numerical differentiation. The full set of equations proposed for lookup table generation are listed below:

$$\begin{cases} f_{c_{i,t}} = \widehat{\phi}_{c_{i,t}} \\ f'_{c_{i,t}} = (f_{c_{i,t}} - f_{c_{i,t-1}})\frac{T}{l_m}, \quad t = 1..T,\ f'_{c_{i,0}} = f'_{c_{i,1}} \\ f''_{c_{i,t}} = (f_{c_{i,t}} - f_{c_{i,t-1}})\frac{T}{l_m}, \quad t = 1..T,\ f''_{c_{i,0}} = f''_{c_{i,1}} \\ f_{s_{i,t}} = \widehat{\phi}_{s_{i,t}} \\ f'_{s_{i,t}} = (f_{s_{i,t}} - f_{s_{i,t-1}})\frac{T}{l_m}, \quad t = 1..T,\ f'_{s_{i,0}} = f'_{s_{i,1}} \\ f''_{s_{i,t}} = (f_{s_{i,t}} - f_{s_{i,t-1}})\frac{T}{l_m}, \quad t = 1..T,\ f''_{s_{i,0}} = f''_{s_{i,1}}, \end{cases} \tag{29}$$

where f_c, f'_c, and f''_c are lookup tables used to generate predetermined points for the motor rotating the CB drive and f_s, f'_s and f''_s are lookup tables used to generate predetermined points for the motor rotating the geared ring.

Notably, if T and N are chosen as the powers of two indexes of lookup tables, the lookup tables automatically switch between the last and first ones at the end of each full turn of the monowheel, enabling the robot to cover distances longer than its rail length without additional effort.

The amount of memory required to store all lookup tables for one CB drive can be calculated using

$$M_{sz} = \frac{6 \cdot N \cdot T \cdot b}{1000}, \tag{30}$$

where M_{sz} is the size (in Kb) of all lookup tables required to control one CB drive and b is the size of one entry in the lookup table (in bits).

Using (30), we can easily show that the proposed controller can be implemented using a wide range of FPGAs. For example, for the parameters $N = 8$, $T = 256$, and $b = 16$ (which were used in our experiments), M_{s_z} does not exceed 200 Kb, which means that the controller can easily fit in low-cost Intel FPGAs from the Cyclone IV E (any part) or MAX10 (any part but 10M02 and 10M04) families. Meanwhile, the controller's lookup tables are large and are to be transferred on each cycle of communications through the Ethernet POWERLINK. However, in practice, doing so is unnecessary because the contents of these tables changes only when SABER reconfigures its rail. Thus, the contents of the lookup table can be reloaded either in the asynchronous part of the Ethernet POWERLINK communication cycle by using Service Data Objects (SDO) or even by using the full FPGA reconfiguration executed by the robot's integrated computer [56].

The simulation results obtained using OpenModelica [57] when the controller was tuned indicated that our design can provide high precision in positioning the CB drive; however, the motor dedicated to rotating the geared ring exhibited a large displacement, which is undesirable, during the acceleration phase (Fig. 13). This displacement was present because the geared ring motor did not accelerate the much heavier robot to a similar extent as it did to parts of the CB drive. Consequently, when the robot reaches a constant speed, most

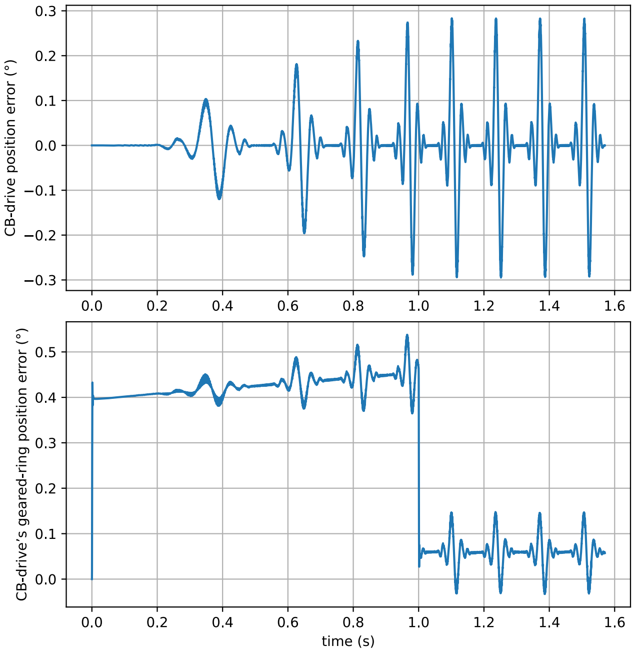


FIGURE 13. Positioning error of CB drive itself and the geared ring of the CB drive.

of the torque provided by the motor is used to move the CB drive's sprockets to prevent jamming; this made the robot's inertia much less influential on errors in the geared ring's positioning. To compensate for the error during the acceleration phase, we introduced an additional feed-forward channel to the input of current controller. The final design of the controller for the CB drive's motor is illustrated in Fig. 14.

With respect to the motor rotating the CB drive, this current boost channel can be disabled by setting its coefficient K_b to 0, and the coefficient for the motor connected to the geared ring K_b is calculated using

$$K_b = \frac{J_w}{K_{t_s}}, \tag{31}$$

where J_w is the robot's moment of inertia with respect to the axis of the motor rotating the geared ring.

As evident in Fig. 15, our enhancement significantly improves positioning precision in the acceleration; the error is much lower than the error for the position of the CB drive's geared ring illustrated in Fig. 13.

Finally, and notably, even J_w can be difficult to calculate for a real-world robot with its complex mechanical interactions and variable payload. By contrast, K_b can be manually tuned with ease because it is the only parameter in the generation of set points that may introduce difficulties in calculations of its value.

VI. EXPERIMENTAL RESULTS

Our evaluation experiments proceeded in several parts:

A. Geometrical simulation.
B. Simulation of the generation of predetermined points
C. CB drive simulation.
D. FPGA firmware synthesis.
E. Firmware-in-loop FPGA simulation.

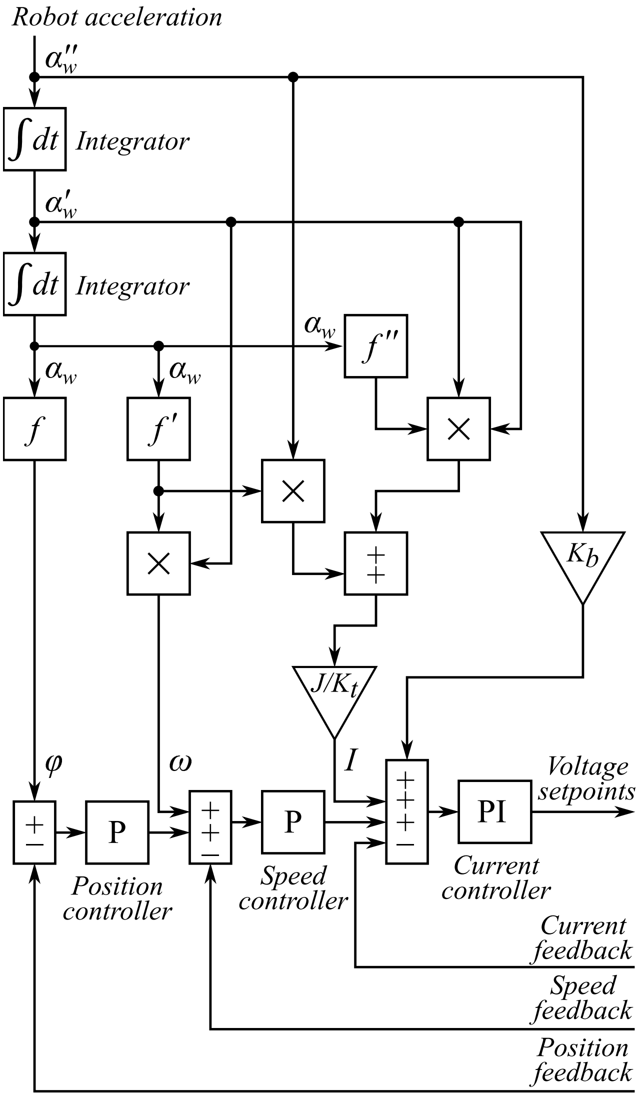


FIGURE 14. Final design of controller of CB drive motor, which includes lookup tables and additional current feedforward.

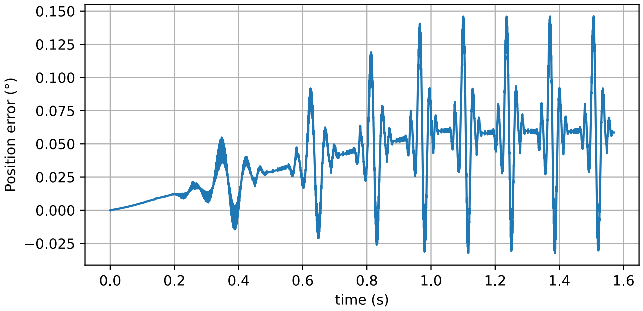


FIGURE 15. Error in positioning of the CB drive's geared ring with additional current feedforward.

The geometrical simulation (Fig. 16) was conducted in purpose-built software that we created and to verify SABER's

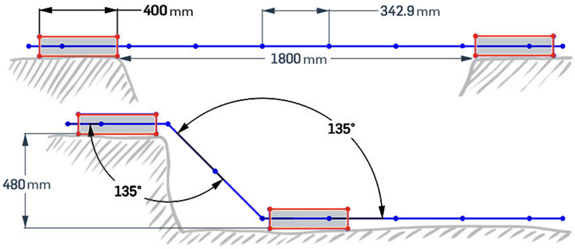


FIGURE 16. Geometric analysis of SABER negotiating a gap and step.

ability to climb steps and traverse gaps. The remaining parts of the experiments were used to evaluate the CB drive and corresponding control law, which are our most novel contributions. Specifically, in the second part of the experiment, we determined which set of fast fourier transform (FFT) parameters yielded the best trade-off between precision and computational cost. In the third part, we conducted a dynamic simulation to demonstrate that the control law and controller design satisfied the precision requirements and did not cause the transmission to jam. In the last two parts of the experiment, we synthesized the FPGA firmware for the CB drive controller and tested its performance in a cycle-accurate firmware-in-loop simulation.

A. GEOMETRICAL SIMULATION

Table 1 presents the characteristics of the robot used in our experiments. The reconfigurable rail was positioned such that the platform can move along it for the robot to traverse a gap with a width of 1.8 m, which is greater than the diameter of the robot in the monowheel configuration, or climb a step with a height of 48 cm, which is greater than the radius of the robot's wheel.

We used the software developed in [58], which was written in Python and implemented using the *numpy* package, to simulate the behavior of SABER and its mechanical transmission. See the simulation results video in the media on IEEE Xplore; refer to [58] for the source codes. All modules were positioned at 135°, which required the CB drives to rotate most actively; the robot was set to move at 10 km/h, faster than competing MRRs proposed in [14], [15], [24], [38], [59] and sufficient for exploration and outpost building in extraterrestrial environments.

B. SIMULATION OF THE GENERATION OF PREDETERMINED POINTS

Per (22), the output of the proposed point generator depends on two parameters: the number of FFT points T and the number of left lower-frequency harmonics K. A larger T results in more precise estimates, made by using (22), that better match the ideal trajectory described in (15) and (16), but it entails greater memory requirements, according to (30). The value of T also depends on the number of roller chain joints per module N_{rc}; this is because the output of the point generator should be sufficiently precise for the CB drives' sprockets to be correctly aligned relative to the chain rollers. Finally, T should be equal to some power of 2 to simplify lookup table switching during continuous movement in the monowheel configuration. Thus, a reasonable value for T can be obtained as follows:

TABLE 1. Robot parameters.

Parameter	Value
Number of modules	8
Roller chain step	12.7 mm
Roller chain joints per module (N_{rc})	27
Module's outer radius	478.02 mm
Distance between front and rear CB drives	400 mm
Distance along crosswise joint axis	342.9 mm
Wheel diameter	956.04 mm
Motor type	Permanent magnet synchronous motors
Motor positioning sensor type	Resolver

$$\begin{aligned} N_{rc} &= \frac{l_m}{l_c} \\ T &= 2^{\lfloor \log_2(10 \cdot N_{rc}) \rceil} \end{aligned} \tag{32}$$

In essence, T is equal to the power of 2 that is closest to a value higher than N_{rc} by one decimal order. Substituting the parameters for the robot (Table 1) into equation (32), we obtain $T = 256$.

Because an FFT with a finite number of harmonics introduces additional error (i.e., results in predetermined points that are farther from their values in the ideal control law), a lower K value entails a lower tolerance to error in the positioning of the CB drive. Conversely, a higher K results in a higher acceleration, which is difficult to reach using motors available on the market.

One of the key parameters limiting the ability of an engineer to tune the drive regulator to the required accuracy is the deviations of sensor measurements from the true value. For the resolver sensors used in the simulated robot (Table 1), this deviation was usually within 6–10 arcmin. Generally, this means that precision requirements determined by the choice of K must be greater than 10 arcmin. Practically, it is almost impossible to tune the controller to reach a dynamical positioning error that is comparable with the sensor precision. In our experiments, we set a minimum required threshold for the positioning accuracy of 70 arcmin (7 times higher than the sensor's capabilities). Thus, any K value that demanded better precision was considered to be too small and ineligible.

To determine the positioning accuracy requirements for each K value, we introduced a CB drive jamming safety diagram, which is created as follows. Each position of the CB drive along the chain was defined by the rotation of the sprocket geared ring and corresponded to an allowable range of the CB drive's rotation that caused no collision between the sprockets' cogs and the chain's rollers. The critical values of the CB drive rotation were calculated for each point of the chain. An example of critical CB drive rotation angles

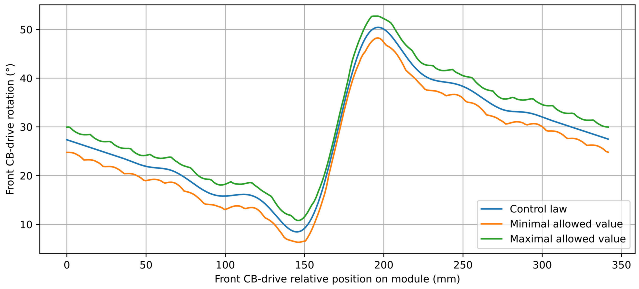


FIGURE 17. **Critical values of rotation of the front CB drive for a module bent by 135°.**

for a module bent by 135° is illustrated in Fig. 17. The error of the geared ring drive is defined as the distance between the position of the CB drive on the chain and its expected position. Thus, the safe range of the CB drive angles should be considered for the actual position of the CB drive (i.e., with the error). This error is calculated using (16). Correspondingly, we can define a set of deviation pairs containing the deviations of the geared ring drive and CB drive from the control law that are large enough for a collision to occur. We can then visualize a safe zone by plotting this set of points on a plane. The size of the safe zone varies depending on the angles of the modules. It is smallest and largest when the module is the most (135° or 225°) and least bent (0°), respectively. The safe zone for our experiment is illustrated in Fig. 18. A control system keeping the errors within the intersection of the smallest safe zones for the front and rear CB drives results in no collision in any configuration.

Notably, the jamming safety diagram gives us a set of sufficient but not necessary conditions (pertaining to the positioning error of the CB-drive motors) that, if met, results in no jams. This does not mean that a jam occurs if these conditions are violated.

Two jamming safety diagrams can be constructed for each K value: one for the front and the other for the rear CB drive. Using these diagrams, one can verify whether the minimum accuracy requirement exceeds the minimum threshold of 70 arcmin. The minimum K value, whose accuracy requirement is above the threshold, should be chosen as the parameter for the point generator. In our experiments, we used $K = 6$ per the aforementioned procedure.

Table 2 presents the numerical simulation results of SABER's movement at parameters $K = 6$ and $T = 256$ relative to the predefined points for an ideal CB drive motor defined in (15) and (16); Figs. 19–21 illustrate the transient response in terms of the position, speed, and acceleration at different parts of the transmission during SABER's motion along one module.

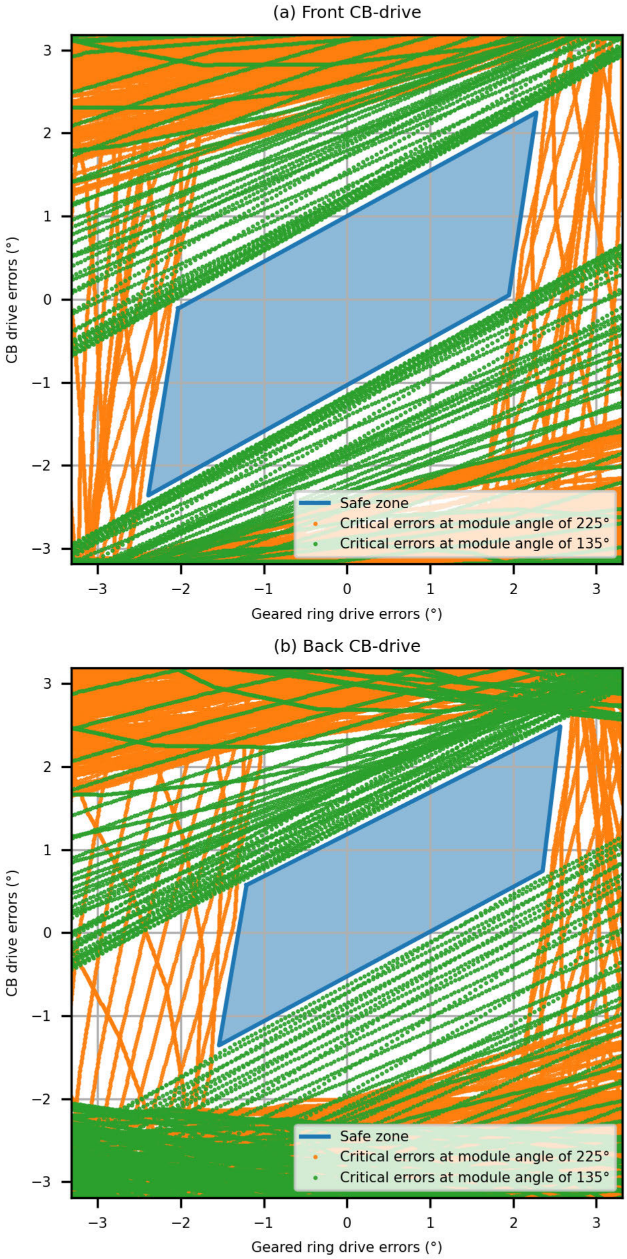


FIGURE 18. **Safe zone illustrated in jamming safety diagram for (a) a front CB drive and (b) rear CB drive.**

TABLE 2. **Numerical Simulation Results for SABER's Movement.**

Parameter	**Value**
Number of harmonics K	6
Number of FFT points T	256
Maximum angular speed (ideal control law)	1364.8 rpm
Maximum angular acceleration (ideal control law)	91971.9 rad/s^2
Maximum angular speed (six lower harmonics)	683.0 rpm
Maximum angular acceleration (six lower harmonics)	8835.9 rad/s^2
Angular oscillation	±3.7°
Linear oscillation	±11.4 mm

The results indicated that to reach a top speed of 10 km/h, the robot (with the proposed point generator and T and K parameters) requires drives with approximately 2 times less angular speed and at least 10 times less angular acceleration than an ideal CB drive motor [defined in (15) and (16)]. The requisite angular velocity of 683 rpm and acceleration of 8836 rad/s^2 of the transmission motors can be easily reached by a wide range of modern, industrial synchronous motors.

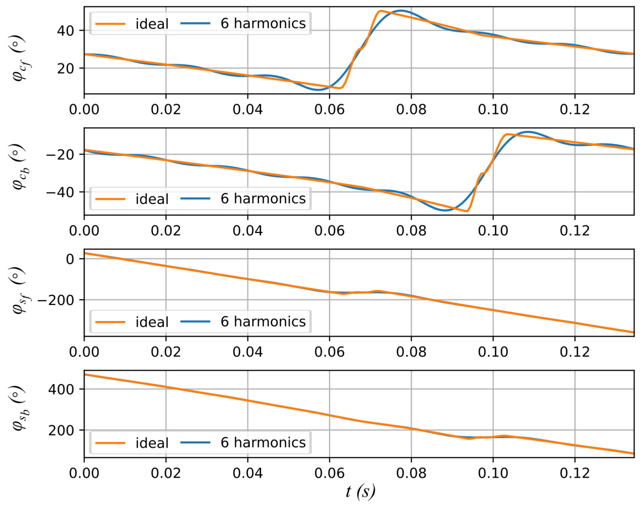


FIGURE 19. Drive angles in degrees over time *t*: front-CB-drive angle ϕ_{c_f}, rear-CB-drive angle ϕ_{c_b}, front-CB-drive geared-ring angle ϕ_{s_f}, and rear-CB-drive geared-ring angle ϕ_{s_b}.

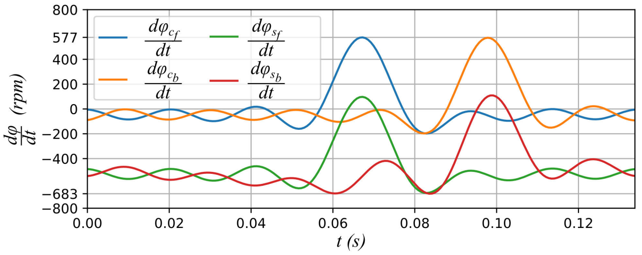


FIGURE 20. Drive speed over time *t*: front-CB-drive speed $d\phi_{c_f}/dt$, rear-CB-drive speed $d\phi_{c_b}/dt$, front-CB-drive geared-ring speed $d\phi_{s_f}/dt$, and rear-CB-drive geared-ring speed $d\phi_{s_b}/dt$.

Finally, we analyzed the oscillations of the robot in the monowheel configuration (Table 2). The maximum amplitude was ±3.7° for pitch and ±11.4 mm for vertical linear movement. These values are acceptable for most payloads and for movement in rough terrain. Although the oscillations can affect quality of vision systems, we can use stabilization algorithms to compensate for the effects of oscillation within this amplitude range, as demonstrated in by [60].

C. CB DRIVE SIMULATION

To analyze the dynamic behavior of the CB drive, its model and proposed controller were implemented in OpenModelica software [57]. The motor was simulated using the same parameters as those of actual PMSMs: B&R 8LVA23.R00150000-0 was used to rotate the CB drive and B&R 8LVA33.R00150000-0 was used to rotate its geared ring. The same motors were previously used for size and weight estimations in geometric simulations and for the mechanical design of the CB drive (Fig. 7). The maximum voltage provided by the power supply was 80 V.

Fig. 23 illustrates the positioning error of the proposed controller during the transition from acceleration to motion at 10 km/h in the monowheel configuration (Fig. 22a) at a speed of 10 km/h. Fig. 24 illustrates the same but for the configuration where the robot platform moves along the rail in the most extreme configuration (module angles of 135° and 225° in Fig. 22b). In our simulations, we assumed that the rail was infinitely long and we did not consider the situation where the platform reaches the end of the rail. We assumed these to evaluate the controller's performance under the most extreme working conditions; specifically, a normal rail reconfigured from manipulators of a single SABER is insufficiently long to allow for motion at such high speeds (of 10 km/h) to be practicable.

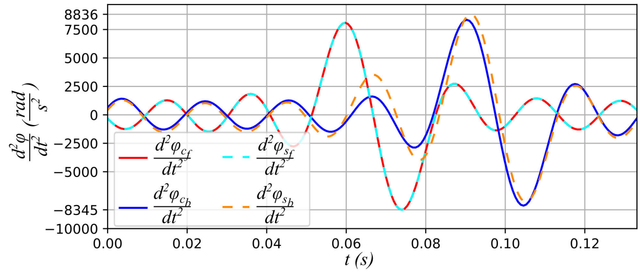


FIGURE 21. Drive acceleration over time *t*: front-CB-drive acceleration $d^2\phi_{c_f}/dt^2$, rear-CB-drive acceleration $d^2\phi_{c_b}/dt^2$, front-CB-drive geared-ring acceleration $d^2\phi_{s_f}/dt^2$, and rear-CB-drive geared-ring acceleration $d^2\phi_{s_b}/dt^2$.

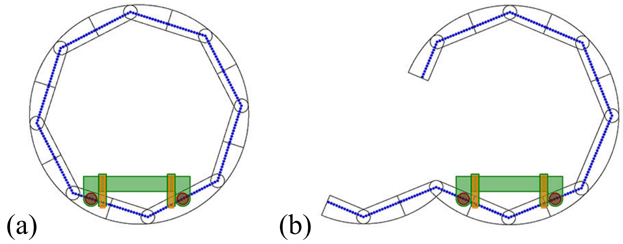


FIGURE 22. Simulated SABER rail configuration: (a) monowheel and (b) rail, including extreme module angles of 135° and 225°.

The results demonstrated that our proposed controller achieved high dynamic precision on both drives in both simulated configurations. As expected, the maximum positioning error was reached when the acceleration was largest during the movement phase. Furthermore, and as mentioned, no error range was defined for a single CB-drive motor, which guaranteed no transmission jamming—which is realized when the positioning error of the motors are within the safe zone in the jamming safety diagram.

Figs. 25 and 26 present safety jamming diagrams for the positioning errors of SABER's CB drives for movement at 10 km/h in the monowheel (Fig. 22a) and rail (Fig. 22b) configurations.[2] The simulation results demonstrated that the proposed controller provides sufficient precision for guaranteeing no transmission jamming even in the most extreme rail configurations when SABER is moving at maximum speed.

[2] As evident in from Fig. 18, none of the critical points, corresponding to modules bent by 225°, lie within ±1°. Thus none of these points are presented in Fig. 25 and the following figures.

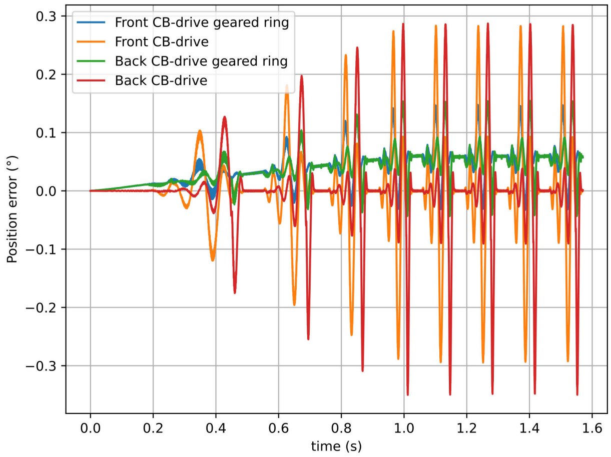


FIGURE 23. Positioning error during the transition from acceleration to motion at 10 km/h in a monowheel configuration for front CB drive geared ring, front CB drive, rear CB drive geared ring and rear CB drive motor control.

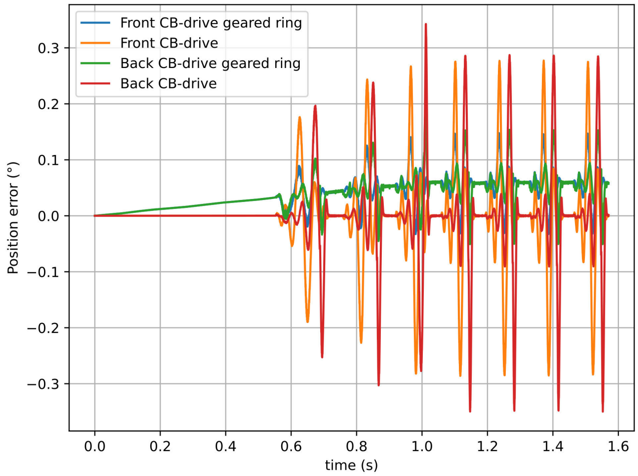


FIGURE 24. Positioning error during the transition from acceleration and motion at 10 km/h in a rail configuration for front CB drive geared ring, front CB drive, rear CB drive geared ring and rear CB drive motor control.

D. FPGA FIRMWARE SYNTHESIS

The controller introduced in Section IV was implemented in the Verilog hardware description language and synthesized for an Intel Cyclone IV EP4CE22 FPGA. To reduce both area consumption and latencies, all signals were represented in terms of a 50-MHz sigma-delta modulation, which was directly processed without demodulation using approaches described in [61] and [62]. Finally, the point generator's non-linearities were implement in the form of content-addressed memory [63].

Table 3 illustrates the synthesis results. Its abbreviations are as follows: Qty, quantity; LE, Intel Cyclone IV E logic element; and MEM, size of utilized memory blocks. Notably due to our use of direct sigma-delta bitstream processing, our design requires no hardware multipliers or any other type of vendor-specific digital signal processing block, which makes it compatible with a wide range of FPGAs available on the market, including those not manufactured by Intel.

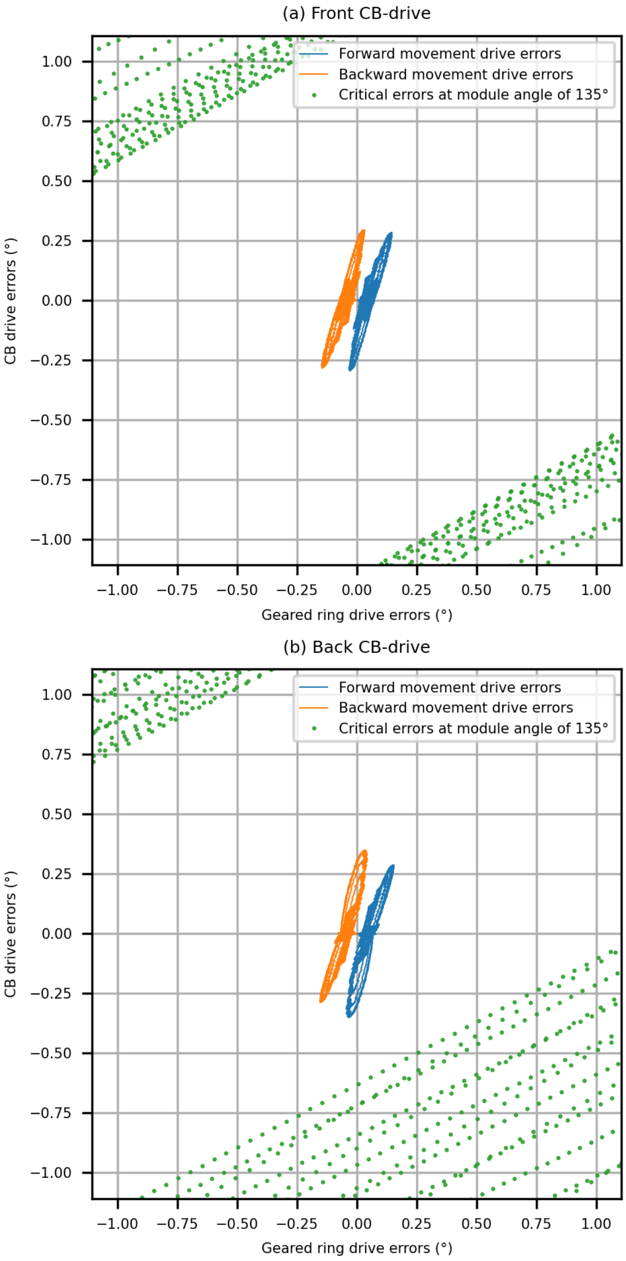


FIGURE 25. Jamming safety diagram (OpenModelica simulation) for the monowheel configuration for (a) front CB drive and (b) rear CB drive.

Finally, the experimental results (Table 3) unequivocally demonstrate that our design can easily fit not only in chosen low-cost FPGAs but also in any FPGA from the Cyclone IV E device family (except for EP4CE6, the smallest one).

E. FIRMWARE-IN-LOOP FPGA SIMULATION

To verify the utility of the FPGA firmware that we developed, we generated a cycle-accurate blackbox model in Vmodel toolbox software [67] and co-simulated the model with the SABER transmission model. Our simulations in this experiment were identical to those in Subsection VI-C except

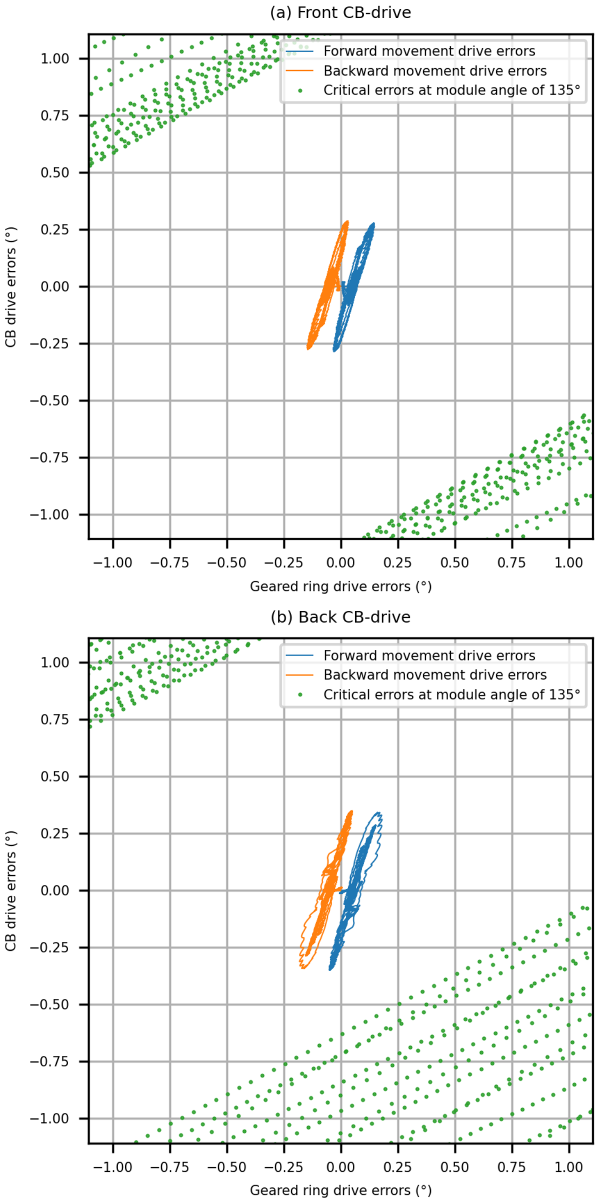


FIGURE 26. Jamming safety diagram (OpenModelica simulation) for the rail configuration for (a) front CB drive and (b) rear CB drive.

TABLE 3. FPGA Firmware Synthesis Results.

IP core	Qty	LE	MEM, bit	LE, %	MEM, %
Proposed CB drive motor controller	2	571	98304	2.55	16.16
FOC current controller [64]	2	1231	0	5.52	0
Resolver processing core based on a Kalman filter [65]	2	1549	0	6.94	0
Ethernet POWERLINK controlled node core [66]	1	1506	32792	6.75	5.39
Four-port Ethernet POWERLINK Intelligent hub [47]	1	391	128	1.75	0.02
Overall design	-	**8599**	**229528**	**38.52**	**37.73**

for additional simulations that accounted for limited sensor precision and output rate, dead times of the power stage transistors, and delays in the analogue-to-digital converter.

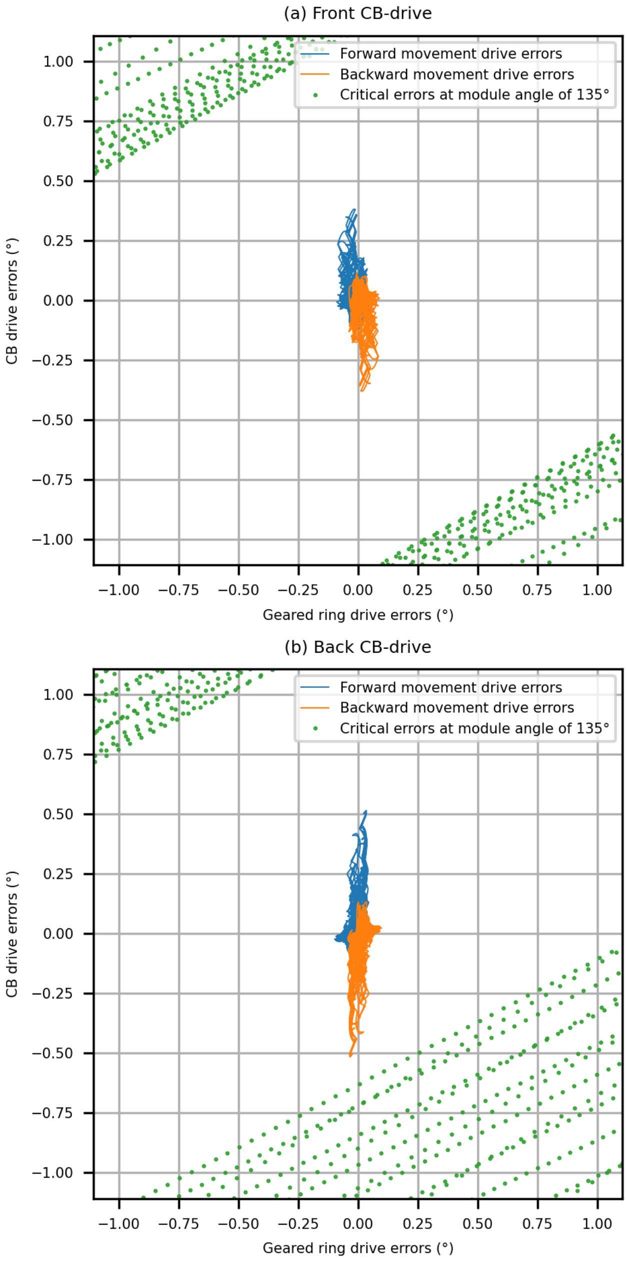


FIGURE 27. Jamming safety diagrams representing the FPGA firmware-in-loop simulation (monowheel configuration) for (a) front CB drive and (b) rear CB drive.

The resulting jamming safety diagrams are presented in Figs. 27 and 28, where parts a and b of these figures are for the monowheel and rail configurations, respectively.

According to the results, the firmware-in-the-loop simulation had worse precision relative to the pure OpenModelica simulation except for the front CB drive in the rail configuration (Fig.28a). This result could not be explained by the limited simulation time. Nonetheless, because the front and rear CB drives operate under the same conditions, it can be assumed that in the worst case scenario, the jamming safety diagrams of the front and rear CB drives are similar.

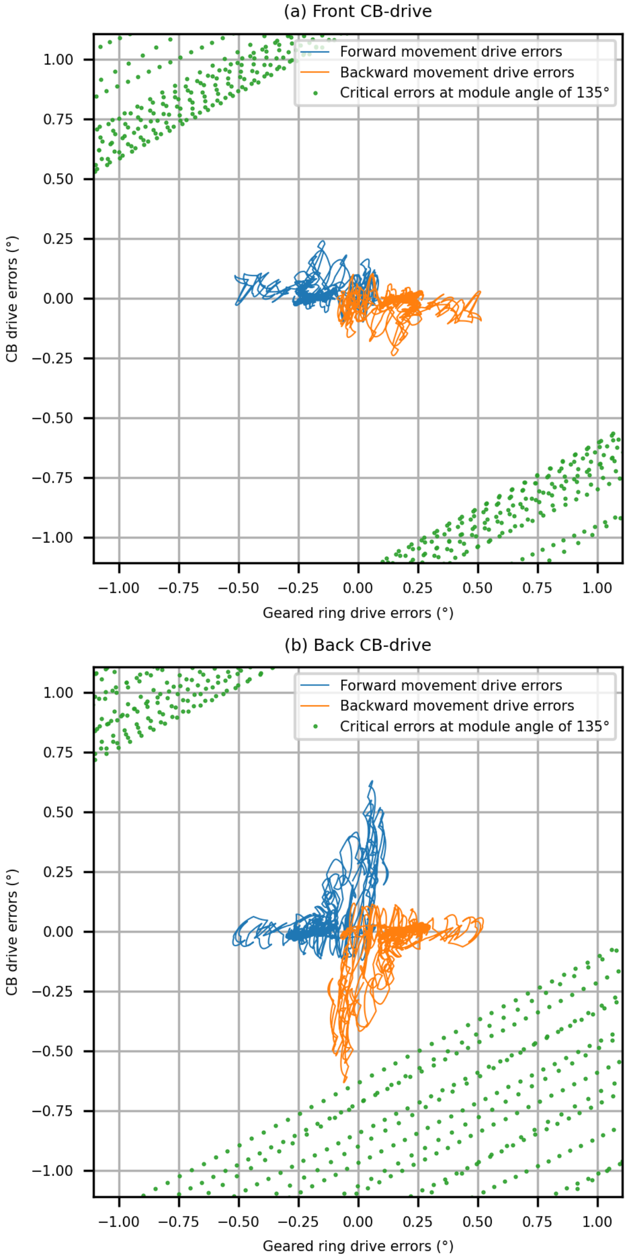


FIGURE 28. Jamming safety diagrams representing the FPGA firmware-in-loop simulation (rail configuration) for (a) front CB drive and (b) rear CB drive.

Generally, a higher positioning error can be represented by both a lower precision in direct sigma-delta bitstream processing relative to that in the floating-point arithmetic used in OpenModelica and more realistic simulations (including simulations that account for limitations in sensor precision or rate and the behavior of the power stage transistor). Notably, the positioning error of the FPGA-based controller in firmware-in-the-loop simulations did not exceed the limits in the jamming safety diagrams. The current and velocity of the motors during the experiments were also within their respective specifications. Therefore, our implementation of the FPGA-based controller allows SABER to move at a top speed of 10 km/h in all configurations, providing sufficient positioning precision to prevent the transmission from jamming.

TABLE 4. Abbreviations.

Abbriviation	Description
SABER	Step, Assembler, Bridge, Explorer Robot
FPGA	Field Programmable Gate Array
MRR	Modular Reconfigurable Robot
DOF	Degree of Freedom
LIDAR	Light Detection and Ranging
GPS	Global Positioning System
AGV	Automated Guided Vehicles
UAV	Unmanned Aerial Vehicle
CB drive	Cheese-Burger drive
ID	Identifier
DC motor	Direct Current Motor
FOC	Field Oriented Control
PMSM	Permanent Magnet Synchronous Motor
SDO	Service Data Object
FFT	Fast Fourier Transform
Qty	Quantity
LE	Intel Cyclone IV E logic element
MEM	Size of utilized memory blocks

TABLE 5. Symbols. Part I.

Symbol	Description
N	Number of modules
l_m	Module length (in m)
l_p	Distance between the front and rear CB drives (in m)
U	CB drive inner gear ratio
l_c	Roller chain step (in m)
α_i	ith module angle (in rad)
i	Module ID
q	q-axis, a polyline starting at the initial position of the first module(with ID $i = 0$) and passing along the roller chain
q_f	Position of the front sprocket module on the q-axis (in m)
$\vec{r}$	Vector rotation function
$\vec{s}_i$	Initial point of the ith module in its local coordinates
$\vec{m}_i$	Midpoint of the ith module in its local coordinates
$\vec{e}_i$	Endpoint of the ith module in its local coordinates
$\vec{S}_i$	Start point of the ith module in global coordinates
$\vec{M}_i$	Middle point of the ith module in global coordinates
$\vec{E}_i$	End point of the ith module in global coordinates
β_i	ith module global rotation angle
$q_r(q)$	Coordinate on the corresponding module, relative to the module's start point (in m)
$q_c(q)$	Coordinate on the corresponding module, relative to the module's middle point (in m)
q_o	Offset distance from module's midpoint, which is where the CB drive starts rotating (in m)
$\gamma(q)$	Auxiliary transition angle, used to calculate the CB drive rotation around the module's midpoint (in rad)
$\eta(q)$	Global orientation angle of the CB drive (in rad)
$\eta_n(q)$	Auxiliary non-normalized value of global orientation angle of CB drive $\eta(q)$ (in rad)
$\xi(q)$	Sprocket angle (in rad)

TABLE 6. Symbols. Part II.

Symbol	Description
$\vec{P}_0(q)$	Initial point, midpoint, and endpoint of the nearest module to the left from q
$\vec{P}_1(q)$	Initial point, midpoint, and endpoint of the nearest module to the right from q
$a(q)$	Interpolation coefficient between $\vec{P}_0(q)$ and $\vec{P}_1(q)$
$\vec{P}(q)$	Global coordinates of the point on the chain
q_f	Position of the front CB drive on the chain (in m)
q_b	Position of the rear CB drive on the chain (in m)
β_p	Global orientation of the platform (in rad)
ϕ_c	CB drive rotation (in general) (in rad)
ϕ_{c_f}	Front CB drive rotation (in rad)
ϕ_{c_b}	Back CB drive rotation (in rad)
ϕ_s	CB drive geared ring rotation (in rad)
α_w	Wheel rotation angle in the monowheel configuration (in rad)
α_m	Wheel rotation while passing one module (in rad)
α_r	Angle of wheel relative to the initial point of the currently passing module (in rad)
R_i	Radius of the circle inscribed in the roller chain N-gon (in m)
$q_{f_r}(\alpha_w)$	Relative position of the front CB drive on the current module as a function of the wheel's rotation angle α_w (in m)
$q_f(\alpha_w)$	Absolute position of the front CB drive on the current module as a function of the wheel's rotation angle α_w (in m)
T	Number of points in discrete Fourier transform
K	Number of lowest kept harmonics from discrete Fourier transform
$\widetilde{\phi}_{i,t}$	Set of points, describing the motion control law after filtering
q_b	Coordinates of initial point of the corresponding module (in m)
q_e	Coordinates of endpoint of the corresponding module (in m)
ϕ	Drive angle (in rad)
$\lambda(q)$	Picewise linear function, describing the drive angle's derivative at modules' initial points and endpoints (in rad)
$\mu(q)$	Blending function used to incorporate the piecewise linear function into the filtered control law
$\lambda_{i,t}$	Set of discrete values of $\lambda(q)$ (in rad)
$\mu_{i,t}$	Set of discrete values of $\mu(q)$
$\widehat{\phi}_{i,t}$	Set of discrete values of the filtered motion control law with flattened edges (in rad)
φ_c	Predetermined points of CB drive angular position (in rad)
φ_s	Predetermined points of CB drive geared ring angular position (in rad)
ω_c	Predetermined points of CB drive angular velocity (in rpm)
ω_s	Predetermined points of CB drive geared ring angular velocity (in rpm)
I_c	Predetermined points of CB drive motor current (in A)
I_s	Predetermined points of CB drive geared ring motor current (in A)

TABLE 7. Symbols. Part III.

Symbol	Description
J_c	Moment of inertia of connections to the motor rotating the CB drive (in $\mathrm{kg} \cdot \mathrm{m}^2$)
J_s	Moment of inertia of connections to the motor rotating the CB drive geared ring (in $\mathrm{kg} \cdot \mathrm{m}^2$)
K_{t_c}	Torque constant of the motor rotating the CB drive (in $\mathrm{N{\cdot}m}/\mathrm{A}$)
K_{t_s}	Torque constant of the motor rotating the CB drive geared ring (in $\mathrm{N{\cdot}m}/\mathrm{A}$)
f_c, f'_c, f''_c	Lookup tables used to generate predetermined points for motor rotating CB drive
f_s, f'_s, f''_s	Lookup tables used to generate predetermined points for motor rotating the geared ring
M_{sz}	Size (in Kb) of all lookup tables required to control one CB drive (in bits)
b	Size of one lookup table entry (in bits)
K_b	Boost channel coefficient (in $\mathrm{A{\cdot}s^2}/\mathrm{rad}$)
J_w	Robot's moment of inertia with respect to the axis of motor rotating the geared ring (in $\mathrm{kg} \cdot \mathrm{m}^2$)
N_{rc}	Number of roller chain joints on one module

VII. CONCLUSION

This study proposed a heterogeneous modular reconfigurable robot that is suitable for future autonomous extraterrestrial missions. The design comprises a platform and a chain with 2-DOF modules. SABER functions in monowheel, rail trolley, and manipulator configurations. Our novel design of SABER's mechanical transmission enables the platform to move along the reconfigurable rail in both monowheel and rail trolley configurations; it also enables the hanging reconfigurable rail to move when the platform is fixed on support legs.

For robust and collision-free transmission, we formulated a novel jamming safety diagram, which is used to determine controller precision requirements. A new positioning controller was designed to meet those requirements and was implemented in FPGA firmware that is compatible with a wide range of low-cost integrated circuits.

SABER has several advantages relative to its counterparts: 1) high speed locomotion in the monowheel configuration, 2) the ability to traverse gaps whose length is longer, 3) the ability to climb steps whose height is no taller than its wheel radius, 4) the ability to pass through passages narrower than its wheel diameter, 5) the ability to manipulate objects using two robotic arms with changeable tools, and 6) an improved working area. All of these can be achieved by combining several robots into one, if necessary.

APPENDIX
ABBREVIATIONS AND SYMBOLS

Herein, the abbreviations (Table 4) and symbols (Tables 5–7) used in the paper are outlined.

REFERENCES

[1] A. Reneau, "Private lunar initiatives," in *Moon First and Mars Second*. Big Sky, MT, USA: Springer, 2021, pp. 63–68.

[2] Z. S. Patel, T. J. Brunstetter, W. J. Tarver, A. M. Whitmire, S. R. Zwart, S. M. Smith, and J. L. Huff, "Red risks for a journey to the red planet: The highest priority human health risks for a mission to Mars," *NPJ Microgr.*, vol. 6, no. 1, pp. 1–13, Dec. 2020.

[3] J. Thangavelautham, A. Chandra, and E. Jensen, "Autonomous robot teams for lunar mining base construction and operation," in *Proc. IEEE Aerosp. Conf.*, Mar. 2020, pp. 1–16.

[4] J. Thangavelautham and Y. Xu, "Modeling excavation, site preparation, and construction of a lunar mining base using robot swarms," in *Proc. Earth Space*, Apr. 2021, pp. 1310–1325.

[5] M. Malenkov, "Self-propelled automatic chassis of Lunokhod-1: History of creation in episodes," *Frontiers Mech. Eng.*, vol. 11, no. 1, pp. 60–86, Mar. 2016.

[6] J. Wu and A. Giménez, "On the maximization of the science output of space missions," *Space Sci. Rev.*, vol. 216, no. 1, pp. 1–13, Feb. 2020.

[7] A. Orlova, R. Nogueira, and P. Chimenti, "The present and future of the space sector: A business ecosystem approach," *Space Policy*, vol. 52, May 2020, Art. no. 101374.

[8] M. Di Castro, M. Ferre, and A. Masi, "CERNTAURO: A modular architecture for robotic inspection and telemanipulation in harsh and semi-structured environments," *IEEE Access*, vol. 6, pp. 37506–37522, 2018.

[9] S. Makris, "Mobile dual arm robots in cooperation with humans," in *Cooperating Robots for Flexible Manufacturing*. Cham, Switzerland: Springer, 2021, pp. 355–372. [Online]. Available: https://link.springer.com/book/10.1007/978-3-030-51591-1#authorsandaffiliationsbook

[10] T. Cordie, R. Steindl, R. Dungavell, and T. Bandyopadhyay, "Modular field robots for extraterrestrial exploration," *Adv. Astronaut. Sci. Technol.*, vol. 3, no. 1, pp. 37–47, Jun. 2020.

[11] E. Karamipour, S. F. Dehkordi, and M. H. Korayem, "Reconfigurable mobile robot with adjustable width and length: Conceptual design, motion equations and simulation," *J. Intell. Robot. Syst.*, vol. 99, nos. 3–4, pp. 797–814, Sep. 2020.

[12] W. Reid, R. Fitch, A. H. Göktoğan, and S. Sukkarieh, "Sampling-based hierarchical motion planning for a reconfigurable wheel-on-leg planetary analogue exploration rover," *J. Field Robot.*, vol. 37, no. 5, pp. 786–811, Aug. 2020.

[13] A. Yun, D. Moon, J. Ha, S. Kang, and W. Lee, "ModMan: An advanced reconfigurable manipulator system with genderless connector and automatic kinematic modeling algorithm," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4225–4232, Jul. 2020.

[14] K. Tadakuma, R. Tadakuma, A. Maruyama, E. Rohmer, K. Nagatani, K. Yoshida, A. Ming, M. Shimojo, M. Higashimori, and M. Kaneko, "Mechanical design of the wheel-leg hybrid mobile robot to realize a large wheel diameter," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 3358–3365.

[15] S.-C. Chen, K.-J. Huang, W.-H. Chen, S.-Y. Shen, C.-H. Li, and P.-C. Lin, "Quattroped: A leg–wheel transformable robot," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 2, pp. 730–742, Apr. 2014.

[16] A. M. Romanov, V. D. Yashunskiy, and W.-Y. Chiu, "SABER: Modular reconfigurable robot for industrial applications," in *Proc. IEEE 17th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2021, pp. 53–59.

[17] M. Cross, A. Ellery, and A. Qadi, "Estimating terrain parameters for a rigid wheeled rover using neural networks," *J. Terramech.*, vol. 50, no. 3, pp. 165–174, Jun. 2013.

[18] J. Iqbal, M. Rehman-Saad, A. Malik, and A. Mahmood-Tahir, "State estimation technique for a planetary robotic rover," *Revista Facultad Ingeniería Univ. Antioquia*, no. 73, pp. 58–68, 2014.

[19] H. Zhuang, N. Wang, H. Gao, and Z. Deng, "Quickly obtaining range of articulated rotating speed for electrically driven large-load-ratio six-legged robot based on maximum walking speed method," *IEEE Access*, vol. 7, pp. 29453–29470, 2019.

[20] E. A. Oyekanlu, A. C. Smith, W. P. Thomas, G. Mulroy, D. Hitesh, M. Ramsey, D. J. Kuhn, J. D. Mcghinnis, S. C. Buonavita, N. A. Looper, M. Ng, A. Ng'oma, W. Liu, P. G. Mcbride, M. G. Shultz, C. Cerasi, and D. Sun, "A review of recent advances in automated guided vehicle technologies: Integration challenges and research areas for 5G-based smart manufacturing applications," *IEEE Access*, vol. 8, pp. 202312–202353, 2020.

[21] D. Heczko, Z. Zeman, and P. Siroky, "Modular rover design for exploration and analytical tasks," in *Modelling and Simulation for Autonomous Systems*, vol. 11995. Palermo, Italy: Springer, 2020, p. 203.

[22] A. Ellery, N. Patel, L. Richter, R. Bertrand, and J. Dalcomo, "ExoMars rover chassis analysis and design," in *Proc. 8th Int. Symp. Artif. Intell., Robot. Autom. Space (iSAIRAS ESTEC)*, 2005, p. 17.1.

[23] Y. Su, T. Wang, K. Zhang, C. Yao, and Z. Wang, "Adaptive nonlinear control algorithm for a self-balancing robot," *IEEE Access*, vol. 8, pp. 3751–3760, 2020.

[24] N. Tan, R. E. Mohan, and K. Elangovan, "Scorpio: A biomimetic reconfigurable rolling–crawling robot," *Int. J. Adv. Robot. Syst.*, vol. 13, no. 5, 2016, Art. no. 1729881416658180.

[25] M. Russo and M. Ceccarelli, "A survey on mechanical solutions for hybrid mobile robots," *Robotics*, vol. 9, no. 2, p. 32, May 2020.

[26] L. Ni, F. Ma, and L. Wu, "Posture control of a four-wheel-legged robot with a suspension system," *IEEE Access*, vol. 8, pp. 152790–152804, 2020.

[27] Y. Bai, L. Sun, and M. Zhang, "Terramechanics modeling and grouser optimization for multistage adaptive lateral deformation tracked robot," *IEEE Access*, vol. 8, pp. 171387–171396, 2020.

[28] G. Zhong, J. Cao, X. Chai, and Y. Bai, "Design and performance analysis of a triphibious robot with tilting-rotor structure," *IEEE Access*, vol. 9, pp. 10871–10879, 2021.

[29] S. Makris, P. Tsarouchi, A.-S. Matthaiakis, A. Athanasatos, X. Chatzigeorgiou, M. Stefos, K. Giavridis, and S. Aivaliotis, "Dual arm robot in cooperation with humans for flexible assembly," *CIRP Ann.-Manuf. Technol.*, vol. 66, no. 1, pp. 13–16, 2017.

[30] E. Ackerman, "KIT's ARMAR-6 humanoid will help humans fix other robots," *IEEE Spectr.*, 2018. [Online]. Available: https://spectrum.ieee.org/kit-armar6-humanoid

[31] T. Klamt *et al.*, "Remote mobile manipulation with the centauro robot: Full-body telepresence and autonomous operator assistance," *J. Field Robot.*, vol. 37, no. 5, pp. 889–919, Aug. 2020.

[32] K. Stoy, D. Brandt, and D. J. Christensen, *Self-Reconfigurable Robots: An Introduction*. Cambridge, MA, USA: MIT Press, 2010.

[33] A. M. Romanov, "A review on control systems hardware and software for robots of various scale and purpose. Part 2. Service robotics," *Russian Technol. J.*, vol. 7, no. 6, pp. 68–86, Jan. 2020.

[34] R. J. Alattas, S. Patel, and T. M. Sobh, "Evolutionary modular robotics: Survey and analysis," *J. Intell. Robot. Syst.*, vol. 95, nos. 3–4, pp. 815–828, Sep. 2019.

[35] F. Wang, Z. Qian, Z. Yan, C. Yuan, and W. Zhang, "A novel resilient robot: Kinematic analysis and experimentation," *IEEE Access*, vol. 8, pp. 2885–2892, 2020.

[36] N. Tan, A. A. Hayat, M. R. Elara, and K. L. Wood, "A framework for taxonomy and evaluation of self-reconfigurable robotic systems," *IEEE Access*, vol. 8, pp. 13969–13986, 2020.

[37] C. Liu, M. Whitzer, and M. Yim, "A distributed reconfiguration planning algorithm for modular robots," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4231–4238, Oct. 2019.

[38] J. Davey, N. Kwok, and M. Yim, "Emulating self-reconfigurable robots–design of the SMORES system," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 4464–4469.

[39] M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirikjian, "Modular self-reconfigurable robot systems [grand challenges of robotics]," *IEEE Robot. Autom. Mag.*, vol. 14, no. 1, pp. 43–52, Mar. 2007.

[40] S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji, "M-TRAN: Self-reconfigurable modular robotic system," *IEEE/ASME Trans. Mechatronics*, vol. 7, no. 4, pp. 431–441, Dec. 2002.

[41] M. Yim, K. Roufas, D. Duff, Y. Zhang, C. Eldershaw, and S. Homans, "Modular reconfigurable robots in space applications," *Auton. Robots*, vol. 14, no. 2, pp. 225–237, 2003.

[42] M. Plooij, G. Mathijssen, P. Cherelle, D. Lefeber, and B. Vanderborght, "Lock your robot: A review of locking devices in robotics," *IEEE Robot. Autom. Mag.*, vol. 22, no. 1, pp. 106–117, Mar. 2015.

[43] A. Lyder, R. F. M. Garcia, and K. Stoy, "Genderless connection mechanism for modular robots introducing torque transmission between modules," in *Proc. ICRA Workshop Modular Robots, State Art*, 2010, pp. 77–81.

[44] D. Šotola, Ž. Ivandić, and I. Grgić, "Generating different constructional solutions on the hubless wheel example," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 393, Aug. 2018, Art. no. 012124.

[45] P. Cieslak, T. Buratowski, T. Uhl, and M. Giergiel, "The mono-wheel robot with dynamic stabilisation," *Robot. Auton. Syst.*, vol. 59, no. 9, pp. 611–619, Sep. 2011.

[46] Y. Ayalon, L. Damti, and D. Zarrouk, "Design and modelling of a minimally actuated serial robot," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4899–4906, Jul. 2020.

[47] A. M. Romanov, M. P. Romanov, and E. I. Shestakov, "A novel architecture for control systems of modular reconfigurable robots," in *Proc. IEEE II Int. Conf. Control Tech. Syst. (CTS)*, Oct. 2017, pp. 131–134.

[48] A. M. Romanov, M. P. Romanov, S. V. Manko, M. A. Volkova, W.-Y. Chiu, H.-P. Ma, and K.-Y. Chiu, "Modular reconfigurable robot distributed computing system for tracking multiple objects," *IEEE Syst. J.*, vol. 15, no. 1, pp. 802–813, Mar. 2021.

[49] A. M. Romanov and I. S. Mikheenko, "A novel approach for creating modular reconfigurable robots with distributed power system," in *Proc. IEEE Conf. Russian Young Res. Electr. Electron. Eng. (EIConRus)*, Jan. 2018, pp. 974–978.

[50] M.-A. Martínez-Prado, J. Rodriguez-Resendiz, R.-A. Gomez-Loenzo, G. Herrera-Ruiz, and L.-A. Franco-Gasca, "An FPGA-based open architecture industrial robot controller," *IEEE Access*, vol. 6, pp. 13407–13417, 2018.

[51] A. Romanov and E. Slepynina, "Real-time Ethernet POWERLINK communication for ROS. Part I. General concept," in *Proc. Ural Smart Energy Conf. (USEC)*, Nov. 2020, pp. 159–162.

[52] A. Romanov and E. Slepynina, "Real-time Ethernet POWERLINK communication for ROS. Part II. Hardware and software," in *Proc. Ural Smart Energy Conf. (USEC)*, Nov. 2020, pp. 163–166.

[53] A. M. Romanov, F. Gringoli, and A. Sikora, "A precise synchronization method for future wireless TSN networks," *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 3682–3692, May 2021.

[54] V. D. Yashunskiy and A. M. Romanov, "A novel approach to touchdown impact damping for the walking robot based on low-cost geared servo drives," in *Proc. IEEE Conf. Russian Young Res. Electr. Electron. Eng. (EIConRus)*, Jan. 2020, pp. 946–951.

[55] T.-D. Ton, M.-F. Hsieh, and P.-H. Chen, "A novel robust sensorless technique for field-oriented control drive of permanent magnet synchronous motor," *IEEE Access*, vol. 9, pp. 100882–100894, 2021.

[56] A. Romanov, M. Romanov, and A. Kharchenko, "FPGA-based control system reconfiguration using open source software," in *Proc. IEEE Conf. Russian Young Res. Electr. Electron. Eng. (EIConRus)*, Feb. 2017, pp. 976–981.

[57] P. Fritzson *et al.*, "The OpenModelica integrated environment for modeling, simulation, and model-based development," *Model., Identificat. Control, Norwegian Res. Bull.*, vol. 41, no. 4, pp. 241–295, 2020.

[58] A. Romanov and V. Yashunskiy, "A novel modular reconfigurable robot platform drive control simulation," MIREA-Russian Technol. Univ., Moscow, Russia, Tech. Rep., 2021, doi: 10.17632/bmrkvtm498.2.

[59] R. Thakker, A. Kamat, S. Bharambe, S. Chiddarwar, and K. M. Bhurchandi, "ReBiS—Reconfigurable bipedal snake robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 309–314.

[60] M. Odelga, N. Kochanek, and H. H. Bulthoff, "Efficient real-time video stabilization for UAVs using only IMU data," in *Proc. Workshop Res., Educ. Develop. Unmanned Aerial Syst. (RED-UAS)*, Oct. 2017, pp. 210–215.

[61] A. Klein and W. Schumacher, "Algebraic operations on delta-sigma bitstreams," *Math. Comput. Appl.*, vol. 23, no. 3, p. 49, Sep. 2018.

[62] A. Romanov, "High-efficient implementation of neural networks and ANFIS using direct bitstream processing," in *Proc. IEEE 60th Int. Sci. Conf. Power Electr. Eng. Riga Tech. Univ. (RTUCON)*, Oct. 2019, pp. 1–5.

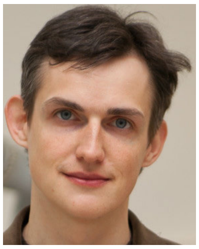
[63] R. Alexey and R. Mikhail, "FPGA based implementation of content-addressed memory based on using direct sigma-delta bitstream processing," in *Proc. IEEE NW Russia Young Res. Electr. Electron. Eng. Conf. (EIConRusNW)*, Feb. 2016, pp. 320–324.

[64] A. M. Romanov, B. V. Slaschev, and M. A. Volkova, "A comparison of hardware implementations of FOC controllers for asynchronous motor drive based on FPGA," in *Proc. 2nd Int. Conf. Ind. Eng., Appl. Manuf. (ICIEAM)*, 2016, pp. 1–5.

[65] L. Harnefors, ''Speed estimation from noisy resolver signals,'' in *Proc. IET Conf.*, Jan. 1996, pp. 279–282.

[66] A. M. Romanov, "A novel architecture for field-programmable gate array-based Ethernet powerlink controlled nodes," *Trudy MAI*, no. 106, pp. 1–25, Jun. 2019.

[67] A. Romanov and S. Bogdan, ''Open source tools for model-based FPGA design,'' in *Proc. Int. Siberian Conf. Control Commun. (SIBCON)*, May 2015, pp. 1–6.



ALEXEY M. ROMANOV (Senior Member, IEEE) received the degree (Hons.) in mechatronic engineering and the Ph.D. degree in electrical and electronics engineering from MIREA—Russian Technological University, Moscow, Russia, in 2010 and 2014, respectively. He is currently an Associate Professor with MIREA—Russian Technological University. During his career, he took part in a wide range of scientific and industrial projects and coauthored more than 80 articles and patents. His current research interests include motion control, robotics, energy, real-time communication, and FPGA design.

VLADIMIR D. YASHUNSKIY (Member, IEEE) received the degree (Hons.) in robotic engineering from MIREA—Russian Technological University, Moscow, Russia, in 2009. For the next decade, he worked as an Engineer and a Software Designer in multiple industrial projects. He is the coauthor of several patents in robotics. His current research interests include robotics, mechanical engineering, and computer simulation.

WEI-YU CHIU (Member, IEEE) received the Ph.D. degree in communications engineering from National Tsing Hua University, Hsinchu, Taiwan, in 2010. He is currently an Associate Professor of electrical engineering with National Tsing Hua University. His research interests include multiobjective optimization, reinforcement learning, and their applications to control systems, robotics, and smart energy systems.