Set-point Control for a Ground-based Reconfigurable Robot

Set-point Control for a Ground-based Reconfigurable Robot

Wei Cheah Bruno Vilhena Adorno Simon Watson Barry Lennox

***Abstract*— Reconfigurable mobile robots are well suited for inspection tasks in legacy nuclear facilities where access is restricted and the environment is often cluttered. A reconfigurable snake robot, MIRRAX, has previously been developed to investigate such facilities. The joints used for the robot's reconfiguration introduce additional constraints on the robot's control, such as balance, on top of the existing actuator and collision constraints. This paper presents a set-point controller for MIRRAX using vector-field inequalities to enforce hard constraints on the robot's balance, actuator limits, and collision avoidance in a single quadratic programming formulation. The controller has been evaluated in simulation and early experiments in some scenarios. The results show that the controller generates feasible control inputs that enable the robot to retain its balance while moving with less oscillation and operating within the actuation and collision constraints.**

I. Introduction

The use of mobile robots for inspection, monitoring and intervention has gained traction in recent years. This traction is in part driven by technological advancement as well as demanding challenges that are best suited for mobile robots such as integrity-compromised structures and radioactive environments [1]. In particular, the challenge with restricted access points is a common place in legacy nuclear facilities where access can be limited to 150 ṁm access ports [2]. Prior work on this challenge employs a reconfigurable design approach [2]–[4]. The typical design consist of two or more rigid bodies connected by actuated joints, with locomotion provided by wheels or tracks. The range of configurations allow such robots to access areas through small ports and to traverse through highly constrained environments such as inside pipes, ducts or between tightly spaced obstacles as shown in Fig. 1.

While reconfigurable mobile robots increases the possible applications to meet user-demands, the control of such robots present additional challenges especially when operated by a human operator. First, the increase in complexity for obstacle avoidance since the configuration space is increased. Second, maintaining static balance as the robot's reconfigurability can result in statically unstable configuration i.e. when the wheels are collinear [5], and configurations which result in significant rolling during motion due to the Centre of Mass (CoM) lying close to the support polygon edge. The rolling experienced during motion is undesirable as it can degrade any sensor measurements attached to the robot, including sensors used for the operator's vision.

This work was supported by UK Research and Innovation through the Engineering and Physical Science Research Council under grant number EP/P01366X/1 and EP/R026084/1

The authors are all with the Department of Electrical and Electronic Engineering, University of Manchester, UK; contact simon.watson@manchester.ac.uk

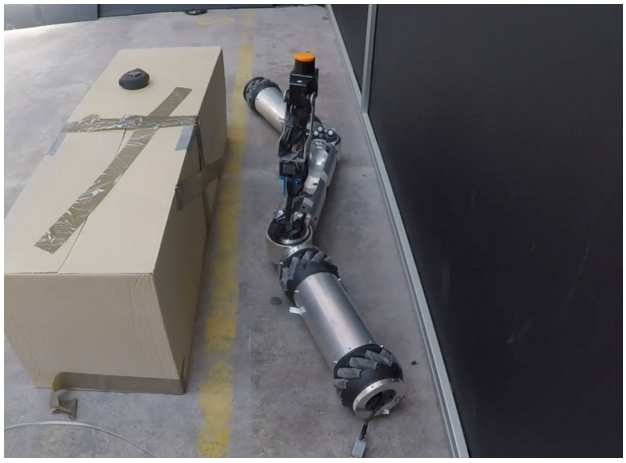

Fig. 1. The MIRRAX reconfigurable robot navigating through a narrow path.

Building on the work of [6], this paper proposes a set-point controller with robot-specific constraints imposed by the robot's reconfigurability, namely in ensuring balancing and smooth motion, while ensuring kinematic feasibility in the presence of obstacles.

This controller would enable both ease of use for the remote operator, enabling them to focus on high-level task such as specifying point of interest for the robot to navigate, or as the local path planner method to be used in sampling-based planners to guarantee feasibility [7]. The platform considered for the research is MIRRAX (Miniature Inspection Robot for Restricted Access eXploration); however, this approach is envisaged to be applicable for other reconfigurable robots.

The remainder of the paper is structured as follows: Section II presents the current state-of-the-art in control design considerations for reconfigurable robots. Section III provides a brief overview of MIRRAX, the reconfigurable robot used in this paper. Section IV details the formulation of the controller and the constraints that are relevant for reconfigurable robots. Section V presents the evaluation of the controller on the MIRRAX robot. Finally, Section VI concludes the work and presents an outlook to future work.

II. Related Work

Prior work on reconfigurable planar omni-directional mobile platforms where the driving wheels are attached to the platform in a non-fixed manner seem to be very limited, although some authors have made some progress in the past. In addition to the inherent capability to move in any direction

instantaneously using Mecanum, omni-wheels, ball-wheel mechanism or steerable wheels, employing these wheels in a non-fixed manner aims to achieve other improvements such as increasing the velocity ratio of the wheel velocities to the robot velocity [8], or for altering the robot's footprint to fit through confined spaces [4], [9], [10].

The design of the controller for such platforms requires further consideration in the form of the robot's balance on top of existing constraints such as obstacle avoidance, actuator saturation, and joint position limits. Although dynamic balance for robots similar to MIRRAX have been shown to be possible [5], more relevant to this study is static balance due to actuation limitations and desired operation behaviour. Prior approaches explicitly considering balance in the controller of planar reconfigurable platforms typically has a rectangular footprint and uses either the static support polygon [9] or dynamic support (zero-moment point) [11].

What has yet to be considered in the literature is the control of planar reconfigurable mobile platforms with non-rectangular footprint where static balance is an important constraint for the robot's performance. This is because when the robot is not in a statically balanced configuration with respect to the wheels, some of them can lose contact with the ground, preventing the robot from moving effectively or falling over. Therefore, coupling this constraint with other more typical environmental and robot-specific constraints in a single controller capable of generating smooth motions would enable operators and mission planners to focus on the high-level task instead.

III. Robot Overview

The annotations for the MIRRAX robot are shown in Fig. 2. The inertial and robot's frame are labelled $\mathcal{F}_I$ and $\mathcal{F}_b$ respectively. The robot has two leg links attached to its base link via joints $\phi_1 \in [0, \pi)$ and $\phi_2 \in [0, -\pi)$. The wheels on each leg link are labelled W_1 to W_4. The configuration of the robot is defined as $\boldsymbol{q} \in \mathbb{R}^5 = [p_x, p_y, \theta, \phi_1, \phi_2]^T$ where the first three terms are the standard x and y coordinates of the base, and rotation angle θ between the axes x_b and x_I, followed by the two joints connecting the two leg links. The robot is actuated by changing the velocities ω_i, with $i \in \{1, \dots, 4\}$, of the four wheels and the velocities $\dot{\phi}_1$ and $\dot{\phi}_2$ of the leg joints. Therefore, the control input vector is defined as

$$\boldsymbol{u} \triangleq \left[\omega_1, \omega_2, \omega_3, \omega_4, \dot{\phi}_1, \dot{\phi}_2\right]^T.$$

From first principles, the inverse mapping between the input and the configuration velocities is given by [4]

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{D}_w \\ \boldsymbol{0}_{2\times 3} \quad \boldsymbol{I}_{2\times 2} \end{bmatrix} \dot{\boldsymbol{q}} \triangleq \boldsymbol{A}\dot{\boldsymbol{q}}, \tag{1}$$

where $\boldsymbol{D}_w \triangleq \boldsymbol{D}_w(\boldsymbol{q}) \in \mathbb{R}^{4\times 5}$, $\boldsymbol{A} \triangleq \boldsymbol{A}(\boldsymbol{q}) \in \mathbb{R}^{6\times 5}$, and the direct mapping is given by

$$\dot{\boldsymbol{q}} = \boldsymbol{A}^+\boldsymbol{u} \tag{2}$$

where $(\cdot)^+$ is the Moore-Penrose pseudoinverse.

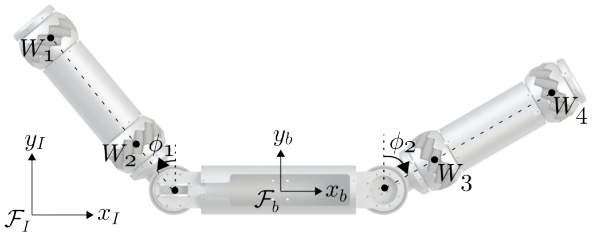

Fig. 2. MIRRAX robot frame of references and annotations.

The matrix $\boldsymbol{D}_w$ explicitly requires the wheels attached to the links to actively rotate when the leg joint moves to prevent dragging. The interested reader is referred to [4] for details on the robot's design and kinematics.

IV. Constrained Controller

To drive the robot to a desired constant configuration $\boldsymbol{q}_d$ while respecting constraints that are linear in the control inputs, we first define a desired closed-loop error dynamics given by

$$\dot{\tilde{\boldsymbol{q}}} + \eta\tilde{\boldsymbol{q}} = \boldsymbol{0}, \tag{3}$$

where $\tilde{\boldsymbol{q}} \triangleq \boldsymbol{q} - \boldsymbol{q}_d$ and $\eta \in [0, \infty)$ is the error gain that is used to determine the convergence rate to zero.

Using (2) in (3), we obtain $\boldsymbol{A}^+\boldsymbol{u} + \eta\tilde{\boldsymbol{q}} = \boldsymbol{0}$, and the optimal control inputs are generated according to the following constrained optimization problem:

$$\begin{aligned} \boldsymbol{u}_{\text{opt}} \in \arg\min_{\boldsymbol{u}} \quad & \|\boldsymbol{A}^+\boldsymbol{u} + \eta\tilde{\boldsymbol{q}}\|_2^2 + \lambda\|\boldsymbol{u}\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{W}\boldsymbol{u} \leq \boldsymbol{w} \end{aligned} \tag{4}$$

where the first term in the objective function is used to enforce the desired closed-loop error dynamics, $\lambda \in [0, \infty)$ is the damping factor to ensure well-conditioned control inputs, and $\boldsymbol{W} \triangleq \boldsymbol{W}(\boldsymbol{q}) \in \mathbb{R}^{r\times 6}$ and $\boldsymbol{w} \in \mathbb{R}^r$ are used to enforce linear constraints in the control inputs.

Since $\boldsymbol{A}$ has full column-rank, then $\boldsymbol{A}^+$ has full row-rank, which implies that the unconstrained control problem

$$\boldsymbol{u}_{\text{opt}} \in \arg\min_{\boldsymbol{u}} \quad \|\boldsymbol{A}^+\boldsymbol{u} + \eta\tilde{\boldsymbol{q}}\|_2^2$$

has infinite solutions, which are given by the set of minimizers

$$U = \left\{\boldsymbol{u} \in \mathbb{R}^n : \boldsymbol{u} = -\boldsymbol{A}\eta\tilde{\boldsymbol{q}} + \left(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^+\right)\boldsymbol{z}\right\}. \tag{5}$$

The set (5) includes solutions for the control input $\boldsymbol{u}$ that are in the nullspace of A^+, which means that, from (2), they do not change the robot configuration and, therefore, are kinematically unfeasible. Although this problem is evident when there are no constraints, it might also occur in the constrained controller (4). Since only solutions in the range space of $\boldsymbol{A}$ are kinematically feasible, they are enforced by means of appropriate constraints, as shown in Section IV-A.1.

A. Constraints

Because the motion of a mobile robot is constrained both by its environment and by itself, we enforce the following constraints directly in the control law (4):

1) Kinematic feasibility,
2) Limits on wheel and leg joint angular velocities,
3) Limits on the leg joint angle,
4) Collision avoidance with the workspace, and
5) Balance.

Similarly to [6], the constraints are formulated as vector-field inequalities by defining nonlinear differentiable functions such as $h(\boldsymbol{q}) : \mathbb{R}^5 \to \mathbb{R}$, in which $h(\boldsymbol{q}) \leq 0$ represents feasible configurations, whereas $h(\boldsymbol{q}) > 0$ represents infeasible configurations, and $\boldsymbol{q} \triangleq \boldsymbol{q}(t)$. The nonlinear constraints are then transformed into linear constraints in the control inputs as follows [12]

$$\frac{\partial h(\boldsymbol{q})}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} \leq -\eta h(\boldsymbol{q}) \implies \frac{\partial h(\boldsymbol{q})}{\partial \boldsymbol{q}}\boldsymbol{A}^{+}\boldsymbol{u} \leq -\eta h(\boldsymbol{q}), \quad (6)$$

where $\eta \in [0, \infty)$. By Gronwall's lemma, $h(\boldsymbol{q}(t)) \leq e^{-\eta t}h(\boldsymbol{q}(0))$, which means that the constraint is satisfied provided $h(\boldsymbol{q}(0)) \leq 0$ [13]. Also, approach velocities to constraints decay exponentially.

1) Kinematic feasibility: As discussed in Section IV, the control inputs must belong to the range space of $\boldsymbol{A}$ to be kinematically feasible. Since

$$\boldsymbol{u} \in \text{range}(\boldsymbol{A}) \iff \boldsymbol{A}\boldsymbol{A}^{+}\boldsymbol{u} = \boldsymbol{u},$$

we introduce the constraint

$$\left(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^{+}\right)\boldsymbol{u} = \boldsymbol{0}, \quad (7)$$

which ensures that the solution is restricted to the set of kinematically feasible minimizers

$$U_{\text{kf}} = \{\boldsymbol{u} \in \mathbb{R}^n : \boldsymbol{u} = -\boldsymbol{A}\eta\tilde{\boldsymbol{q}}\} \subseteq U. \quad (8)$$

2) Limits on Wheels' and Leg Joints' Angular Velocities: To limit the wheel's and leg joints' angular velocities, we have straightforward constraints because those velocities are the control inputs. Therefore, we need to enforce

$$-\boldsymbol{u}_{\text{lim}} \leq \boldsymbol{u} \leq \boldsymbol{u}_{\text{lim}} \quad (9)$$

where $\boldsymbol{u}_{\text{lim}} \in \mathbb{R}^6$ is the vector of wheels' and joints' angular velocity limits. Ineq. (9) can be rewritten to comply with the form in (4) as follows

$$\begin{bmatrix} \boldsymbol{I}_6 \\ -\boldsymbol{I}_6 \end{bmatrix}\boldsymbol{u} \leq \begin{bmatrix} \boldsymbol{u}_{\text{lim}} \\ \boldsymbol{u}_{\text{lim}} \end{bmatrix} \quad (10)$$

3) Limits on Leg Joints' angle: The joint angles of both the legs are constrained to operate within its permitted range of motion. Although in some applications constraints on joint angles are necessary to prevent self-collisions, MIRRAX's legs are constrained mechanically, and thus they will never experience self-collision. First, we define the joints angle error, $\tilde{\boldsymbol{\phi}} \in \mathbb{R}^2$, as

$$\tilde{\boldsymbol{\phi}}_u(t) = \boldsymbol{\phi}(t) - \boldsymbol{\phi}_{u_{\text{lim}}} \quad (11)$$

$$\tilde{\boldsymbol{\phi}}_l(t) = \boldsymbol{\phi}(t) - \boldsymbol{\phi}_{l_{\text{lim}}} \quad (12)$$

where $\boldsymbol{\phi}_{u_{\text{lim}}}$, $\boldsymbol{\phi}_{l_{\text{lim}}} \in \mathbb{R}^2$ are the upper and lower joint angle limits for the two joints on MIRRAX, respectively, and $\boldsymbol{\phi} = \begin{bmatrix}\phi_1 & \phi_2\end{bmatrix}^T$. Because $\boldsymbol{\phi}_{u_{\text{lim}}}$ and $\boldsymbol{\phi}_{l_{\text{lim}}}$ are constant, $\dot{\tilde{\boldsymbol{\phi}}}_u(t) = \dot{\tilde{\boldsymbol{\phi}}}_l(t) = \dot{\boldsymbol{\phi}}(t)$. Hence, to enforce the inequalities $\dot{\tilde{\boldsymbol{\phi}}}_u(t) + \eta_\phi\tilde{\boldsymbol{\phi}}_u(t) \leq \boldsymbol{0}$ and $\dot{\tilde{\boldsymbol{\phi}}}_l(t) + \eta_\phi\tilde{\boldsymbol{\phi}}_l(t) \geq \boldsymbol{0}$, we define

$$\begin{bmatrix} \boldsymbol{B} \\ -\boldsymbol{B} \end{bmatrix}\boldsymbol{u} \leq \eta_\phi \begin{bmatrix} -\tilde{\boldsymbol{\phi}}_u \\ \tilde{\boldsymbol{\phi}}_l \end{bmatrix} \quad (13)$$

where $\boldsymbol{B} = \begin{bmatrix}\boldsymbol{0}_{2\times 4} & \boldsymbol{I}_2\end{bmatrix}$ such that $\dot{\boldsymbol{\phi}} = \boldsymbol{B}\boldsymbol{u}$.

4) Collision avoidance with the workspace: A straightforward representation of the collidable entities is to model both robot and obstacles as a collection of circles.By using a series of fixed-radius circles both for encompassing the robot and the obstacles, the minimum safe distance d_o that would prevent collision between the ith circle on the robot and the jth obstacle is defined as

$$d_{o_{ij}} = r_{\text{robot}_i} + r_{\text{obs}_j} \quad (14)$$

where r_{robot_i} and r_{obs_j} are the radiuses for the ith robot circle and jth obstacle circle, respectively.

To define the collision constraint and its Jacobian between the ith circle on the robot $\boldsymbol{p}_{c_i} \triangleq \boldsymbol{p}_{c_i}(\boldsymbol{q}) \in \mathbb{R}^2$ and the jth static (circular) obstacle, $\boldsymbol{p}_{\text{obs}_j} \in \mathbb{R}^2$, first we define [6]

$$h_{c_ij}(\boldsymbol{q}) = d_{o_{ij}}^2 - ||\boldsymbol{p}_{\text{obs}_j} - \boldsymbol{p}_{c_i}||^2 \quad (15)$$

$$\implies \dot{h}_{c_ij}(\boldsymbol{q}) = 2\left(\boldsymbol{p}_{obs_j} - \boldsymbol{p}_{c_i}\right)^T \boldsymbol{J}_{c_i}\dot{\boldsymbol{q}}, \quad (16)$$

where $\boldsymbol{J}_{c_i} = \partial\boldsymbol{p}_{c_i}/\partial\boldsymbol{q} \in \mathbb{R}^{2\times 5}$. By enforcing $\dot{h}_{c_ij} + \eta_c h_{c_ij} \leq 0$, we obtain the desired inequality

$$2\left(\boldsymbol{p}_{obs_j} - \boldsymbol{p}_{c_i}\right)^T \boldsymbol{J}_{c_i}\dot{\boldsymbol{q}} \leq -\eta_c h_{c_ij} \quad (17)$$

$$\implies 2\left(\boldsymbol{p}_{obs_j} - \boldsymbol{p}_{c_i}\right)^T \boldsymbol{J}_{c_i}\boldsymbol{A}^{+}\boldsymbol{u} \leq -\eta_c h_{c_ij}. \quad (18)$$

Hence for a robot represented by $n_c \in \mathbb{N}$ number of collision circles $\boldsymbol{p}_{c_i}$ where $i = \{1, \ldots, n_c\}$ and an environment having $n_o \in \mathbb{N}$ obstacles $\boldsymbol{p}_{obs_j}$ where $j = \{1, \ldots, n_o\}$, there will be a total of $n_o \cdot n_c$ collision constraints.

5) Balance Constraint: The balance constraint is enforced to ensure that the robot does not roll over, causing the wheels from losing contact with the ground, and also to avoid configurations that can cause significant rolling during motion [4]. In this work, we use a simple constraint to ensure that the projection of the robot's CoM is in its support polygon (see Fig. 3a). Instead of using the convex support polygon (shaded regions in Fig. 3), a line formed by the wheel position nearest and parallel to the base link is used together with an appropriate safety margin (see Fig. 3b).

The distance along the y-axis between the CoM, $\boldsymbol{p}_{\text{com}} \triangleq \boldsymbol{p}_{\text{com}}(\boldsymbol{q})$, and the support line passing through $\boldsymbol{p}_\omega \triangleq \boldsymbol{p}_\omega(\boldsymbol{q})$, given by $\tilde{d}_{\text{com}}(\boldsymbol{q}) = \begin{bmatrix}0 & 1\end{bmatrix}(\boldsymbol{p}_\omega - \boldsymbol{p}_{\text{com}}) \in \mathbb{R}$ is used to define

$$h_b(\boldsymbol{q}) = \alpha\tilde{d}_{\text{com}} + d_b, \quad (19)$$

where $d_b \in [0, \infty)$ is the (constant) safety margin for the distance between the CoM to the support line, and $\alpha = -1$ if the support polygon is above the x_b-axis and $\alpha = 1$ otherwise. Function h_b is a proxy for the support polygon

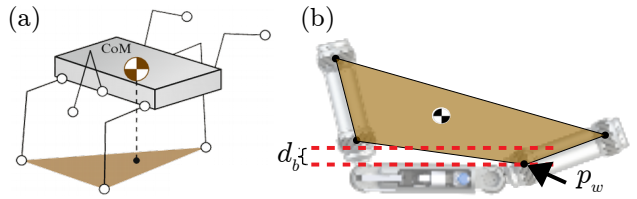

Fig. 3. Static balance is ensured when the CoM projection is within the support polygon. The brown area shows the actual support polygon in both the (a) legged robot [14] and (b) MIRRAX, whereas the red dashed line shows the support line that is used together with a safety margin as an approximation for the boundary of the support polygon closest to MIRRAX's base.

in the sense that balance is maintained by constraining the CoM to lie on either side of the line, depending on the configuration of the robot, and sufficiently far from the base link by choosing an appropriate value for d_b.

Similarly to before, by enforcing $\dot{h}_b + \eta_b h_b \leq 0$, we obtain the desired inequality

$$\boldsymbol{J}_b \dot{\boldsymbol{q}} \leq -\eta_b h_b(\boldsymbol{q}) \implies \boldsymbol{J}_b \boldsymbol{A}^+ \boldsymbol{u} \leq -\eta_b h_b(\boldsymbol{q}) \qquad (20)$$

where $\boldsymbol{J}_b = \partial h_b(\boldsymbol{q})/\partial \boldsymbol{q} \in \mathbb{R}^{1\times 5}$, and $\eta_b \in [0, \infty)$ is the support line approach gain.

Due to discontinuity when the CoM moves between positive and negative coordinates along the y_b-axis (i.e., when α changes), it is useful to analyse the system by using a phase diagram that illustrates the robot configuration as a function of ϕ_1, ϕ_2, and α. The phase diagram is divided into four quadrants, indicated by Q1 to Q4 as shown in Fig. 4. The configurations corresponding to the bottom left and top right of each quadrant is illustrated. It should be noted that the robot can take on any configurations in between the illustrated configurations in Fig. 4.

From the phase diagram, we have a straightforward criterion to automatically select the support line and the value of α such that we prevent the robot from rolling over. For example, if $\phi_1 = 120°$ and $\phi_2 = -140°$, then the robot has a configuration between the two configurations shown in Q4, where the support line passes through wheel 2 and $\alpha = -1$.

The resulting $h_b(\boldsymbol{q})$ map for all possible leg joint configurations is shown in Fig. 5. The area within each red polygon corresponds to infeasible configurations for different values on the safety margin, $d_b = \{0.02, 0.03, 0.04\}$.

As a comparison with the support polygon, Fig. 6 shows the minimum distance between the CoM and the support polygon edges as a function of ϕ_1 and ϕ_2. It also shows the distance to a shrunk (i.e., more conservative) support polygon using safety distances to its edges similar to the ones used in Fig. 5, namely $d_b = \{0.02, 0.04\}$. Since our approach is a rough approximation of the support polygon, Fig. 6 shows that, in order to ensure that the CoM be within the support polygon using constraint (20), a more conservative value for the safety distance must be used.

However the support polygon does not capture situations where wheels lift-off due to the robot's off-centred CoM which was observed to occur more frequently within the infeasible regions in Fig. 5. This situation results in the

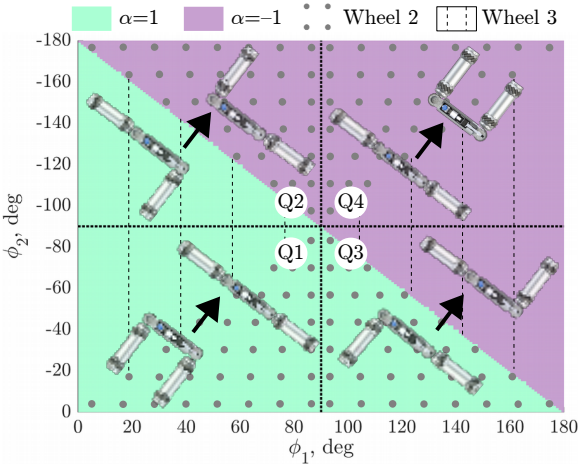

Fig. 4. The balance constraint divided into four quadrants, Q1 to Q4, where each quadrant have a range of joint configurations. The colored diagonal division shows where the sign of α changes.

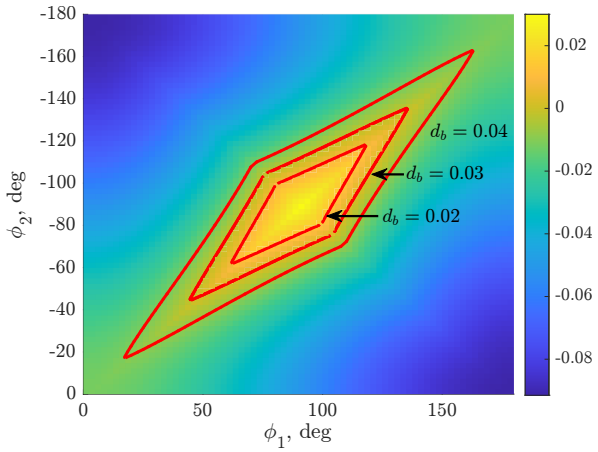

Fig. 5. Surface plot showing $h_b(\boldsymbol{q})$ calculated with $d_b = 0.03$ for all possible leg configurations. Unbalanced configurations (i.e., when the constraint is violated) corresponding to different values of d_b lie within the respective red polygons.

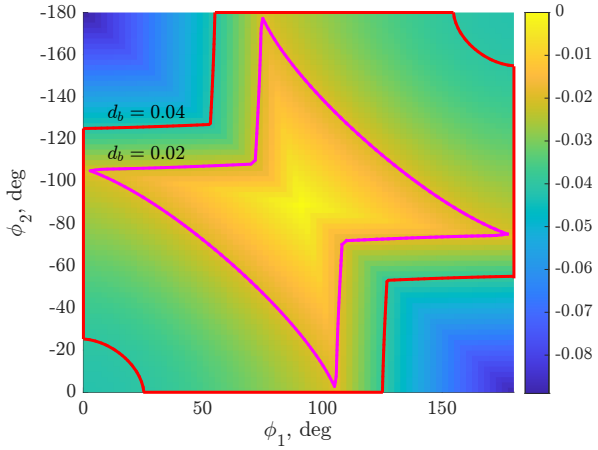

Fig. 6. Surface plot showing the distance between the CoM and the boundary of the support polygon for all possible leg configurations. The area within the red and magenta outlines correspond to unbalanced configurations according to a shrunk (i.e., more conservative) support polygon by a safety distance d_b.

robot's motion deviating significantly from the desired heading direction due to the loss of wheel traction and also rolling during motion. To capture these configurations using the support polygon would require a large safety distance, $d_b > 0.02$, to be used which would significantly limit feasible configurations as seen by the large infeasible region in Fig. 6.

V. Evaluation

This section presents the simulation and early experimental verification of the proposed controller. The experiments were conducted to assess the control inputs generated by the controller in simulation. Applying the same control inputs to the practical robot should show close agreement with the simulated robot on the robot's motion. A video showing the experimental results is available in the supplementary material.

A. Simulation and Experimental Setup

1) Simulation Setup: The simulation was done in MATLAB's Simulink using the built-in QuadProg function for solving (4) at a fixed time-step of 50 ms to generate the control inputs. Using (2) to obtain $\dot{\boldsymbol{q}}$, we then integrate it to obtain the robot's configuration. The planar obstacles used in the simulation were modeled as circles with known centres and radiuses. The parameters used for controller are listed in Table I.

TABLE I
CONTROLLER PARAMETERS

Parameter	Value	Parameter	Value	Parameter	Value
λ	0	d_b	0.035	d_o	0.1
η	10	η_b	1	η_o	0.5
η_ϕ	1				

2) Experimental Setup: All the wheel and joint actuators used on MIRRAX are the Dynamixel XM430-350T. The low-level controller for the two leg joints uses position-based control whereas the four wheels uses velocity-based control. At this stage of early experimental evaluation, the velocity control inputs generated in MATLAB from simulation were first logged as a CSV file. Next, a first-order Euler integration was used to compute the leg joint position from the velocity control input. Finally, the wheel velocities and leg joint positions were published to MIRRAX's actuators at the same control rate of 20 Hz. Ground truth on the robot's pose was obtained using an external VICON motion capture system.

B. Results and Discussion

To show the effect of the balance constraint, the proposed controller was compared with and without the balance constraint in the absence of obstacles. The robot starts in its default configuration, $\boldsymbol{q}(0) = \mathbf{0} \in \mathbb{R}^5$, with an initial set-point $\boldsymbol{q}_{d_1} = [0.5, 0, 0, 85°, -85°]^T$, which is then changed to $\boldsymbol{q}_{d_2} = [0.5, 0, 0, 90°, -90°]^T$ after 20 s. A snapshot of the motions for both controllers are shown in Fig. 7. The robot was able to retain its balance until the set-point was updated at $t = 20$ s. Shortly after, the robot without the balance constraint falls over at $t = 22$s as both the leg joints approach $|90|°$ (see Fig. 8 for the leg joints position).

Fig. 7. Snapshot of motion sequence for $\boldsymbol{q}_{d_2} = [0, 0, 0, 90°, -90°]^T$. (*Top*) Without balance constraint, the robot experiences rolling during motion as it approaches the straight line and finally it falls over at $|\phi_1| = |\phi_2| = \pi/2$. (*Bottom*) With balance constraint, the robot retains its balance throughout the whole motion.

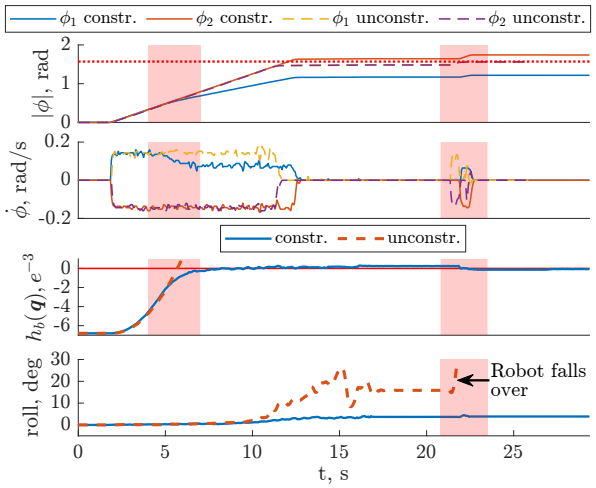

Fig. 8. Comparison of joint motions and constraints with and without the balance constraint. The *top* two figures show the leg joints angles and velocities, respectively. The *bottom* two figures show the balance constraint, $h_b(\boldsymbol{q})$, and the robot's base roll. The dotted-red line in the first plot is the set-point for both ϕ_1 and ϕ_2.

By introducing the balance constraint, the robot is able to retain its balance at the expense of non-zero steady-state error. The amount of roll the robot experienced is also significantly decreased as shown in Fig. 8(bottom). The decrease on $\dot{\phi}_1$ caused by $h_b(\boldsymbol{q})$ becoming closer to zero can be observed in Fig. 8 in the region highlighted in red. The velocity for $\dot{\phi}_1$ does not decay to zero in this patch even though $h_b(\boldsymbol{q}) \approx 0$ since motion along the boundary of the constraint is still possible.

Strictly speaking, $h_b(\boldsymbol{q}) \leq 0$ should be respected at all times and this is in fact observed in the simulation where the control input was taken from. Due to the high safety margin threshold and approach gain used, d_b and η_b respectively, feasible solutions lie very close to the balance constraint as $h_b(\boldsymbol{q}) \to 0$. Hence slight disturbances on the system can lead to the constraint being violated as seen in Fig. 8. A possible solution is to reduce both λ_b and η_b so that $h_b(\boldsymbol{q})$ does not approach zero.

Fig. 9 shows the simulation and open-loop experiment results of MIRRAX traversing a narrow gap of 0.3 m—the robot has a footprint width of 0.51 m in its default U-shape configuration. The robot starts with an initial configuration $\boldsymbol{q}_0 = [0, 0, 0, 30°, -130°]^T$ and an intermediary goal set-point $\boldsymbol{q}_I = [1.3, 0, 0, 30°, -130°]^T$. Once the robot is within a threshold distance to $\boldsymbol{x}_I$, the set-point is then changed to $\boldsymbol{q}_f = [2.3, -0.35, 0, 0, 0]^T$.

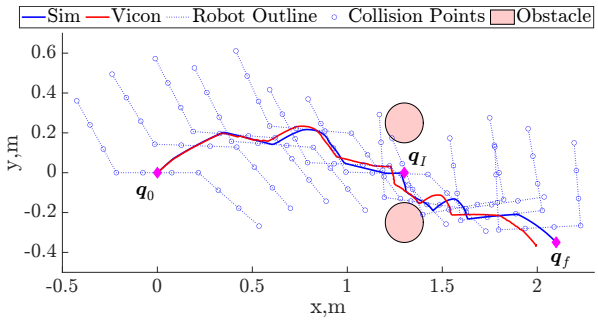

Fig. 9. MIRRAX navigating through a narrow path with two set-points, namely $\boldsymbol{q}_I$ and $\boldsymbol{q}_f$.

From Fig. 9, by simply relying on the local controller, the robot first moves up along the y-axis and then back down again. This motion enables the front half of it to first fit through the narrow gap. Following that, the robot moves diagonally downwards towards the final position. The motion of the rear leg joint, ϕ_1, for the second half of the motion was delayed compared to ϕ_2 due to the presence of the obstacles.

Throughout the motion, the balance constraint for both simulation and experiment data was respected, as shown in Fig. 10. The balance constraint depends only on leg joints ϕ_1 and ϕ_2 which are driven by high gain position-controlled joints, thus the robot was able to track the leg joints set-point closely. The collision constraint was respected in simulation but not in the experiment. This is due to drifts arising from disturbances and imperfect modelling since the robot was operated in open-loop. However the trajectory of the experiment data shows good agreement with the simulation data, strongly suggesting that the control input generated from the controller is feasible for the set of constraints considered in this study.

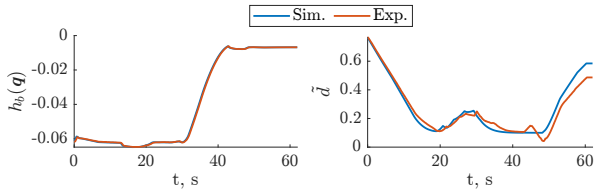

Fig. 10. Results of narrow path tracking for both simulation and open-loop experiment. The *left* plot shows the balance constraint function for maintaining the robot's balance, whereas the *right* plot shows the distance function used in the collision avoidance constraint.

VI. Conclusion

This paper has presented a set-point controller for a reconfigurable mobile platform driven by Mecanum wheels. The balance constraint arising from the robot's reconfigurability has been included as a hard-constraint in the robot's controller alongside environmental and actuation constraints in the form of obstacle avoidance, actuation saturation, and joint angle limit. All of the constraints and motion requirements were formulated as a single optimization problem for finding feasible control inputs. The controller was evaluated in simulation in a set of scenarios that clearly demonstrates its effectiveness in ensuring the constraints are respected. Early experiments to verify the control inputs generated by the controller in simulation shows good agreement between simulation and the actual robot's behaviour.

For future work, the proposed controller will be ported to C++ for real-time closed-loop control. Further areas of development include extending the controller to enable the constraints to be relaxed autonomously. For example, the ingress and egress from a 150 mm access port requires the robot to be in a straight line configuration, which is currently not possible to achieve with the balance constraints active.

References

[1] I. Tsitsimpelis, C. J. Taylor, B. Lennox, and M. J. Joyce, "A review of ground-based robotic systems for the characterization of nuclear environments," *Progress in Nuclear Energy*, vol. 111, pp. 109–124, 2019.

[2] Tepco, "Application of Robot Technology," 2017. [Online]. Available: https://www.tepco.co.jp/en/decommision/principles/robot/index-e.html

[3] Sarcos, "Guardian S Case Study: General Electric (GE)." [Online]. Available: https://www.sarcos.com/products/guardian-s/case-study/ge/

[4] W. Cheah, K. Groves, H. Martin, H. Peel, S. Watson, O. Marjanovic, and B. Lennox, "Mirrax: A reconfigurable robot for limited access environments," *arXiv:2203.00337 [cs.RO]*, 2022.

[5] M. T. Watson, D. T. Gladwin, and T. J. Prescott, "Collinear Mecanum Drive: Modeling, Analysis, Partial Feedback Linearization, and Nonlinear Control," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 642–658, 2020.

[6] M. M. Marinho, B. V. Adorno, K. Harada, and M. Mitsuishi, "Dynamic Active Constraints for Surgical Robots Using Vector-Field Inequalities," *IEEE Transactions on Robotics*, vol. 35, no. 5, pp. 1166–1185, 2019.

[7] Z. Kingston, M. Moll, and L. E. Kavraki, "Sampling-Based Methods for Motion Planning with Constraints," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 159–185, 2018.

[8] K. S. Byun, S. J. Kim, and J. B. Song, "Design of a four-wheeled omnidirectional mobile robot with variable wheel arrangement mechanism," in *IEEE International Conference on Robotics and Automation*, 2002, pp. 720–725.

[9] M. Wada and H. H. Asada, "Design and control of a variable footprint mechanism for holonomic omnidirectional vehicles and its application to wheelchairs," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 6, pp. 978–989, 1999.

[10] Qiushi Fu, X. Zhou, and V. Krovi, "The Reconfigurable Omnidirectional Articulated Mobile Robot (ROAMeR)," *Springer Tracts in Advanced Robotics*, vol. 79, 2014.

[11] H. J. Kim, C. N. Cho, H. S. Kim, and J. B. Song, "Omnidirectional mobile robot capable of variable footprinting based on hub-type drive module," *Transactions of the Korean Society of Mechanical Engineers, A*, vol. 36, no. 3, pp. 289–295, 2012.

[12] B. Faverjon and P. Tournassoud, "A Local Based Approach for Path Planning of Manipulators with a High Number of Degrees of Freedom," in *IEEE Internation Conference on Robotics and Automation*, 1987, pp. 1152–1159.

[13] O. Kanoun, F. Lamiraux, and P.-b. Wieber, "Kinematic control of redundant manipulators : generalizing the task priority framework to inequality tasks," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 785–792, 2011.

[14] S. Kajita and B. Espiau, *Legged Robot Chapter 16*, 2008.