Detachable Modular Robot capable of Cooperative Climbing and Multi Agent Exploration

Sri Harsha Turlapati*, Ankur Srivastava*, K Madhava Krishna, Suril V. Shah

***Abstract*— At the cross section of the fields of Uneven Terrain Navigation and Multi Agent Systems (MAS), in this work, a Detachable Compliant Modular Robot (DCMR) which can perform concurrent scene exploration by detaching into numerous parts, while preserving its ability to climb stairs is proposed and built.**

A spring is designed and used in the modular robot taking the worst-case-scenario of stairs encountered in an urban setting. In addition to the actuators at the wheels, an additional set of actuators per module are introduced to enable the detachment and re-attachment. The design additions and their trade-offs are discussed. Potential applications are presented with special focus on improving coverage of a map with obstacles/slabs large enough to merit exploration by climbing them. The problem of turning in crammed spaces is solved using the ability to detach of DCMR. The detaching & re-attaching capability, and stair climbing of the composite modular robot are demonstrated through experimentation using the prototype.

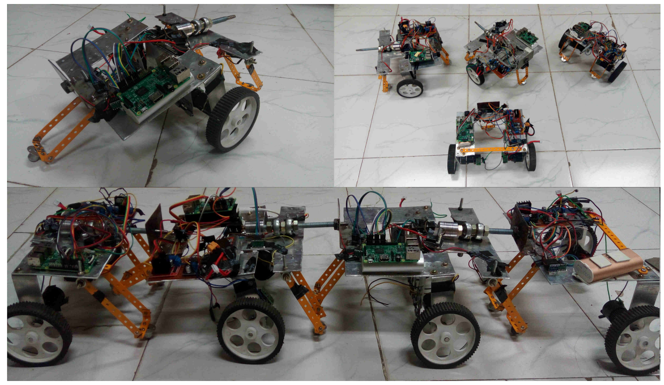

Fig. 1: Prototype of Detachable Compliant Modular Robot (DCMR)

I. Introduction

This work discusses the conception, design and prototyping of a novel compliant modular robot that can climb and explore uneven terrain as well as detach itself into multiple robots to explore flat terrain. With its ability to detach into multiple robots, the system can behave as a Multi Agent System (MAS). On the other hand, its ability to reattach and conquer unstructured terrain, brings in several ramifications in the fields of exploration, resource allocation and task scheduling among robots.

In addition to ground vehicles like [1], the state of the art in unstructured terrain navigation, has spawned many climbing robots like [2], [3], [4], [5], [6], [7] and tracked robots like [8], [9], [10]. These robots can individually facilitate in exploring uneven terrain or climbing obstacles. However, the modularity of these robots that could have been put to use to obtain cooperative behaviour between modules, as well as address redundancy in the case of failure of some modules, hasn't been exploited owing to design constraints. A system that uses cooperative behaviour between robots to climb obstacles is discussed in [11]. However, this system also cannot stably explore a given map because of its gait characteristics.

Many problems in exploration and coverage have been posed and solved using MAS's. Distributed systems such as those presented in [12], [13], [14] have solved the offline as well as online planning problems in MAS. There are different methods like [15] and [16] which also take care of coordination between multiple robots while achieving a common goal. More powerfully, the decomposition methods shown in [17], [18] and [19] have yielded algorithms that solve the offline motion planning to maximize coverage while avoiding obstacles in maps. Furthermore, the work presented in [20] has discussed the potential of using multiple robot systems to help each other climb difficult terrain. Although these algorithms solve the cooperative problems in philosophy, they are mechanism-agnostic and thus can be modified to best fit different robots. Inspired thence, we find motivation to develop the DCMR, given wide-ranging applications like Urban Search and Rescue (USAR), exploration of different types of terrain simultaneously, navigating crammed spaces, etc.

It was shown in [21], [22] that multiple modules can be used to climb greater heights. We build on these works to implement a scheme to unite MAS, unstructured terrain navigation and yield a system with better versatility. The proposed DCMR is shown in Fig. 1. This robot consists of the three modules, detachable into four different individual robots. Compliant elements are used to enable the composite modular robot to climb unstructured terrain. Certain design additions are introduced to help the individual robots remain mobile and stable enough for exploration purposes after detachment.

The key novelty of this work is the design contribution coupled with the application of an exploration scheme, including obstacle exploration. This work being unique in its ability to climb steep stairs and also perform Multi Agent exploration by detaching would spawn a plethora of research problems and applications as a consequence. An extension of

*These authors contributed equally to this work.

Sri Harsha Turlapati, Ankur Srivastava and K Madhava Krishna are with the Robotics Research Lab, IIIT-Hyderabad, TS 500032, India. Suril V. Shah is with IIT Jodhpur, India. Corresponding Author: harsha.turlapati at research.iiit.ac.in

Preprint submitted to 2017 IEEE International Conference on Robotics and Automation. Received September 15, 2016.

coverage planning to uneven terrain with obstacle allocation is another important contribution. The idea of dividing a modular robot into its components and conquering tight spaces which were otherwise inaccessible owing to the large size of the composite modular robot is also demonstrated.

The paper is organized as follows. In Section II, the design of a compliant modular robot for climbing is discussed in brief and the challenges in designing the individual robots and their solutions are presented. In Section III, the design parameters of the individual explorer-bots (EB's) and the composite DCMR are determined in a systematic manner. Section IV presents the mechanical arrangement used for attaching and detaching, and the scheme followed in the detachment and re-attachment process. Section V presents an algorithm developed to use DCMR in MAS applications. Section VI displays the simulations and experiments carried out showcasing the detaching and reattaching ability while elaborating on the applications of DCMR. Finally, conclusions and the scope for future work are discussed in Section VII.

II. Evolution of the Proposed Design

The compliant modular robot discussed above can climb steep obstacles and staircases. The primary idea was to combine the speed of a wheeled robot and the versatility of a modular robot to achieve speedy uneven terrain navigation. The design of the compliant modular robot in [22] was actuated only at the wheels. Modules were designed to house motors and were connected to each other through compliant joints. This design is shown in Fig. 2 (a) where the modules are labelled as M_1, M_2 and M_3 and the joints (marked in red) are passively actuated by optimally designed springs, i.e., the spring loaded joints.

In this section, we propose the design of individual robots that can unite into a compliant modular robot capable of climbing obstacles. The individual robots should be able to reattach with each other, to create a chain of modules. We now discuss the challenges faced to abstract the idea of detachment of the modular robot shown in Fig. 2 (a) into individual robots in the following subsection.

A. Challenges in detaching

To physically mark where the detachment would occur in the modular robot, we define Plane of Detachment (POD) as the junction where any module splits into two explorer-bots (EB's). It is necessary to determine how stable individual EB's would be after detachment to decide where the POD would be introduced. Introducing the POD at the spring loaded joint is ruled out, since during reattachment, the arrangement of the spring could be disturbed. Furthermore, stair climbing ability could be impaired in case of an unverified reattachment. Hence, PODs were introduced along each of the planes P_1, P_2, P_3, perpendicular to the sagittal plane (passing through the midpoint of each of the modules) as shown in Fig. 2. Post detachment into individual EB's, the following are the generic issues faced:

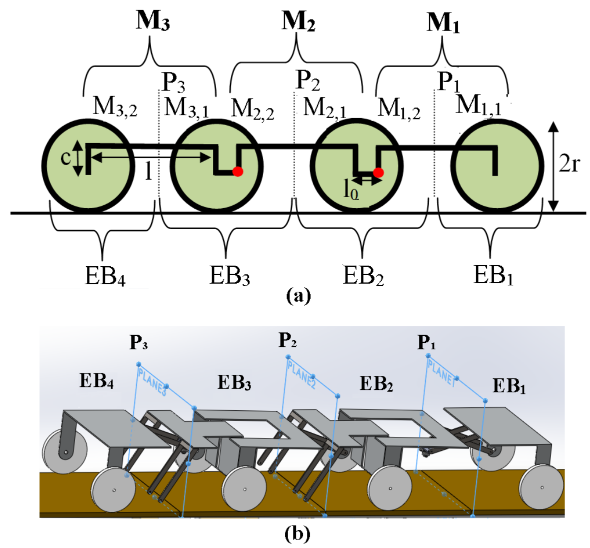

Fig. 2: (a) Design presented in [22] about the sagittal plane and (b) the design modifications

1) Mobility and Alignment: After detachment, $M_{1,1}$ would be mounted on two actuated wheels alone and thus, would be unstable while moving around. Furthermore, aligning $M_{1,1}$ with $M_{1,2}$ for reattachment would be difficult if they aren't stable individually and aligned with each other. We propose to add a castor wheel support (CWS), which is actively controlled (by a servo motor providing τ_{CWS} torque) to solve these problems. The CWS will be lowered at the time of detachment as shown in Fig. 3 (a).

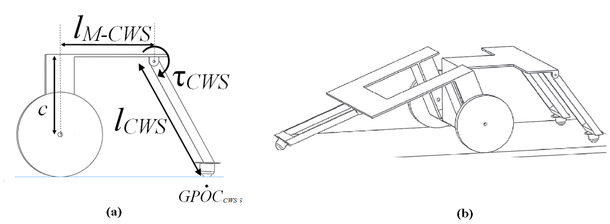

Fig. 3: (a) Actively controlled passive Castor Wheel mount for Individual EB Stability and ensuring Alignment for reattachment, (b) Loose end problem on Spring Loaded EB's

2) Loose end: It may be noted from Fig. 2 (a) that $M_{1,2}$ is the end of M_1 which is connected to the first spring loaded joint. In addition to the inherent elasticity, the spring loaded joint is also bound to get loosened because of repeated spring action during stair climbing. The castor wheel support discussed above aids in controlling this loose end as shown in Fig. 3 (b).

3) Static Stability of Explorer Bots: To ensure the static stability of a EB, we need to view the setting of length of the castor wheel support, l_{CWS}, the servo motor providing τ_{CWS}, along with the positioning of the mount for the servo motor at l_{M-CWS} as an ensemble. Please refer to Fig 3 (a). We wish to set l_{CWS} and l_{M-CWS} such that τ_{CWS} is low enough, so that a commercially abundant servo motor, can supply it.

The CWS solves all these problems. Assuming the individual EB's are stable and mobile, it still remains a challenge to design them in such a way that the composite modular robot after reattachment can retain its ability to climb obstacles and stairs as shown in [21], [22]. Hence the design of the individual EB as well as the resulting compliant modular robot after attachment are both important. We shall now discuss this in the following section.

III. Determination of Design Parameters

The design parameters (l, c and r) as labeled in Fig. 2 (a) and Fig. 5 were determined for the compliant modular robot using the analysis presented in [21], [22] to climb stairs built as prescribed by the International Building Code (IBC) [23]. These are shown in Table I. Since our individual EB's are intended to reattach to climb obstacles, we fix these design parameters and evaluate the design considerations of the EB's next.

A. Design of Individual EB's

Consider the configuration shown in Fig. 4. The castor wheel support must be folded inwards to re-align the EB's belly with the horizontal plane.

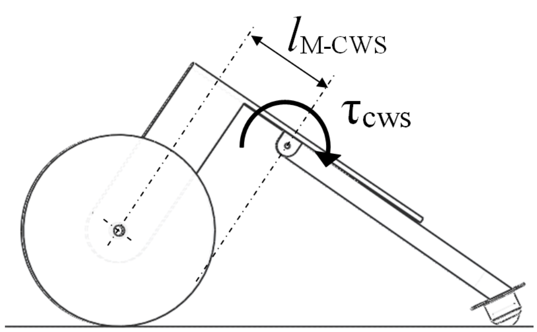

Fig. 4: A configuration of a EB_{nsl}

The free body diagram of the configuration depicted above is shown in Fig. 5. Assuming that the wheel-motor's stall torque is high enough to resist any moment supplied about the wheel-joint W, we can approximate W to be grounded. The moment required to be supplied by the servo motor, τ_{CWS}, to align the EB_{nsl} (EB without a spring loaded joint) with the horizontal plane and hold such a configuration, would materialize as reaction forces R_3 and R_4 acting at the mount of the servo motor (Fig 5). A moment τ_W would thus be applied about W over the EB by these forces.

We aim to minimize the required τ_{CWS} when the EB is parallel to the ground, i.e., to ensure minimum energy is consumed to maintain the horizontal alignment of the bot. Upon analyzing the configuration at angle θ, from simple trigonometry it follows that : $l_1 = c\ sin\theta - l_{M-CWS}\ cos\theta$ and $l_2 = c\ cos\theta + l_{M-CWS}\ sin\theta$. R_3 and R_4 are directly proportional to the τ_{CWS} (say $R_3 \approx R_4 = k\tau_{CWS}$) applied

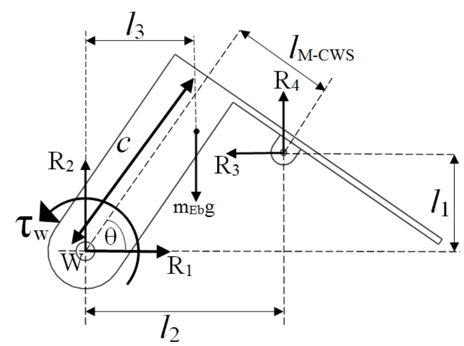

Fig. 5: Choice of l_{M-CWS} to minimize τ_{CWS}

by the servo motor. A static analysis yields the following equations:

$$\tau_W = R_4 l_2 + R_3 l_1 - m_{EB_{nsl}} g l_3 \tag{1}$$

$$\tau_W \propto k\tau_{CWS}\ l_{M-CWS}(sin\theta - cos\theta) \tag{2}$$

From equation 2, at $\theta = \pi/2$, it follows that τ_W is directly proportional to l_{M-CWS}. Thus a lesser value of τ_{CWS} is sufficient for static stability if l_{M-CWS} is maximized. Since l = 0.26 m, the EB_{nsl}'s length would be 0.13 m as we partition the modules for detachment equally. We set the l_{M-CWS} to 0.12 m, a conservative length to accommodate the servo shaft clearance and ensure the servo motor is well within the EB's periphery.

With c = 0.09 m and r = 0.045 m, we require the l_{CWS} to be at least 0.11 m (accommodating for the servo shaft clearance of $\approx$ 0.02 m) to touch the ground at $GPOC_{CWS}$ (Fig.3 (a)). Moreover, the castor wheel must be lowered to a point where its contacts (along with wheel contacts) helps create a large enough support polygon (ABC in Fig. 6) for the EB to be stable after detaching. The greater the area of the support polygon the greater will be the stability of the EB. This justifies the need to maximize l_{CWS}.

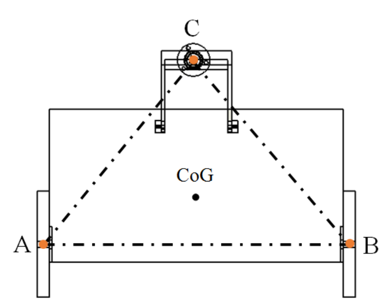

Fig. 6: Support Polygon of a EB_{nsl}

The limiting constraint on the value of l_{CWS} is determined by the requirement that the CWS must be tucked along the length of the module as shown in Fig. 8(d) after attaching to other EB's. This is to provide the necessary clearance for the combined modular robot to climb obstacles and stairs [21], [22] and is of importance since the underbelly of every module will come close to the terrain while climbing convex

Preprint submitted to 2017 IEEE International Conference on Robotics and Automation. Received September 15, 2016.

obstacles. Thus, l_{CWS} needs to be set as low as possible, yet high enough to be able to reach the ground and was fixed at 0.14 m. We design l_{CWS} and l_{M-CWS} to ensure stability of EB_{nsl} and use the same results in EB_{sl}.

B. Design of composite modular robot

In [22], the quasi-static analysis that helped design a stair climbing compliant modular robot was presented. The parameters of this robot are presented in Table I.

TABLE I: Model Parameters of the Compliant Module Robot

Symbols	Quantity	Values
l	Link Length	0.26 m
b	Link Breadth	0.25 m
r	Wheel Radius	0.045 m
c	Link Height from Wheel Center	0.1 m
l_{CWS}	Castor wheel support length	0.14 m
$m_{EB_{nsl}}$	Mass of EB w/o spring	1.65 Kg
$m_{EB_{sl}}$	Mass of EB with spring	2.15 Kg

Assuming that after attachment of EB's the composite modular robot's modules are rigid enough, the springs designed by the optimization routine discussed in [22] are deployed on the prototype at Joints J_1 and J_2. They are respectively valued at :

$$k_1^+ = k_1^- = 0.07447 \ N-m/deg,$$
$$k_2^+ = k_2^- = 0.05761 \ N-m/deg$$

Given these design considerations ensuring the stability of individual EB's and the climbing ability of the composite modular robot, we now discuss the methodology and constraints involved in attaching two EB's and creating a chain of modules required to climb obstacles.

IV. Mechanical arrangement for detaching/reattaching

The objective is to ensure that after the EB's attach with each other, the composite modular robot's modules are rigid enough to aid in climbing. Moreover, no change in the module length must occur after attachment is finished. To achieve this, on one EB participating in attaching/detaching at the POD, a rigid mechanical nut was mounted as shown in Fig. 7 (a). Pieces of 15cm length were cut from a SS threaded rod and attached as an extension to each of the modules on the adjacent side of the POD as shown in Fig. 7 (b). The threaded rod used is made of Grade 304 Stainless Steel and is of tensile strength corresponding to around 515 MPa - 600 MPa, high enough to withstand longitudinal tension encountered while climbing stairs. This justifies the rigidity of each module and the analysis discussed in section III.B. The threaded rods are mounted on two ball bearing supports, which are rigidly attached to the module.

The arrangement is such that the threaded rod rotates about a fixed axis (Z_2) as shown in Fig. 7 (b). Attachment will occur when the threaded rod is inserted into the rigid mount on another EB (such that the axes Z_1 and Z_2 overlap) and the screwing action fastens the threaded rod into the mounted nut to create a module.

With this set-up, there are three types of EB's :

- Type-1 (EB_{sl})
- Type-2 (EB_{nsl} with a threaded rod mount)
- Type-3 (EB_{nsl} with a mounted nut)

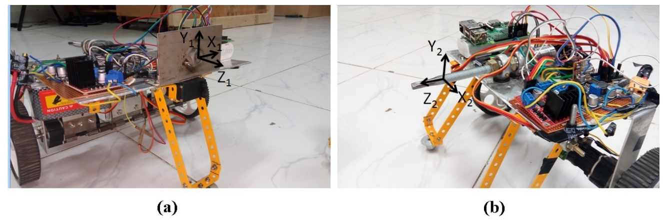

Fig. 7: Mounting for (a) Nut and (b) Threaded Rod

At any point, to create a feasible compliant modular robot able to climb obstacles, one Type-2, one Type-3 and a variable number (depending on the obstacle height) of Type-1 EB's are required. We will now discuss the attaching sequence of two EB's:

1) The two EB's are aligned as shown in Fig. 8 (a)
2) The threaded rod is inserted into the rigid nut across the POD as shown in Fig. 8(b)
3) The threaded rod is actuated by a motor through a coupler, to fasten it into the rigid nut (Fig. 8(c))
4) Following this, the CWS's are lifted up to tuck themselves along the module's length as shown in Fig. 8 (d).

The advantage of this set-up is the ability to set variable module length, which is highly useful in climbing stairs of different heights or obstacles of different sizes while minimizing energy consumption.

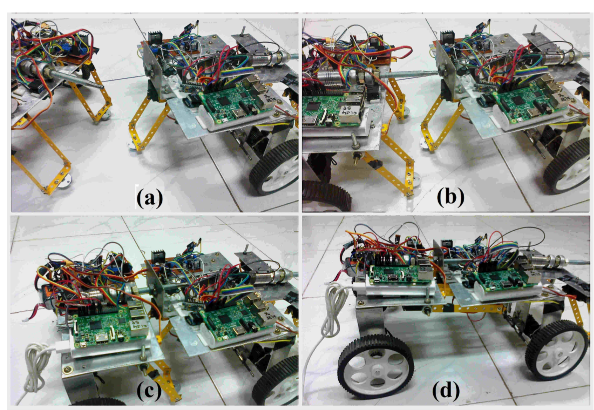

Fig. 8: Attaching using Mechanical set-up

As a consequence of such an ability to detach a compliant modular robot into individual robots, new applications in the domain of MAS open up. In this work, we extend two such well known applications : The Coverage problem (based on a MAS) and a traditional planning problem in tight spaces, incorporating the exploitative ability of DCMR as well as constrains posed by the composite DCMR.

V. Multi Agent Exploration with Obstacle climbing

Various coverage algorithms have been proposed in the past [17], [18] and [19]. Choset in [17] suggested a trapezoid-based exhaustive exploration scheme to maximize coverage while avoiding obstacles. We build a system which can improve coverage by climbing obstacles which are large enough to merit exploration that were otherwise avoided by MAS's owing to mobile robots' scope. While [18] does speak of weighted terrain coverage, the DCMR is physically capable of climbing obstacles and also carry out multi-agent exploration.

We propose to approach the coverage problem by giving precedence to the task of exploring obstacles in a given map while preserving conventional coverage. Given r number of EB's deployed and o number of obstacles with each requiring a specific n_{opt} number of modules to climb them, and an unknown initial state of the system, we propose an obstacle allocation scheme. This scheme is to task some of the robots deployed to climb an obstacle while the rest of the system could be tasked to the coverage problem or to another obstacle, depending on the map. The system will be tasked to explore the area around an obstacle. As the EB's converge onto the obstacle, attachment to form a compliant modular robot to climb is carried out and the obstacle is explored. This algorithm is presented next.

Algorithm 1: OBSTACLE INCLUSIVE MULTI-AGENT EXPLORATION

procedure MAS WITH OBSTACLES
  $Decompose()$
top:
  $Allocate\_obstacles\_and\_explore()$
  $\forall obstacles\_allocated$: $Climb\_and\_explore(n_{opt})$
  if *more obstacles remain unexplored* **then**
    goto *top*

procedure DECOMPOSE
The cell decomposition is done in an object-centric manner, as shown in Fig. 9(a). Cells that don't border an obstacle or are of size lesser than EB's are merged with those that qualify so, as shown in Fig. 9(b).

procedure ALLOCATE_OBSTACLES_AND_EXPLORE
Counting one module as a EB for simplicity, following the allocation scheme shown in subsection IV.A, the EB's are tasked to the obstacles. The cell decomposition is modified and cells surrounding the obstacle are assigned to EB's in a similar fashion. Exploration is initiated in each of these cells ensuring that the EB's finally reach the designated obstacle to attach and initiate climbing.

procedure CLIMB_AND_EXPLORE The allocation scheme discussed ensures an obstacle is assigned the required number (n_{opt}) of EB's to climb it. Subsection IV.C discusses the computation of n_{opt}. Given this, the system climbs the obstacle and explores it.

end

It was discussed in Section IV that one Type-2, one Type-3 and a variable number (n_{opt}-1) of Type-1 EB's are required to constitute a compliant modular robot capable of climbing. The essential contribution of this algorithm is to inculcate this design requirement of the compliant modular robot to climb obstacles into the resource allocation problem of a pool of EB's composing a MAS. We now discuss the determination of n_{opt} (and consequently the number of EB's which were labelled in Fig. 9 (a) as A^3, B^2 and C^2) required to scale the respective obstacle. The optimization formulation which combines the design requirements with the obstacle allocation scheme is also presented.

A. Computing optimal number of explorer bots required to climb h

Given the Obstacle height h, the equations governing the static stability of n modules of a compliant modular robot to climb such an obstacle are given in [21]. Here, the challenge is to estimate the minimum number (n_{opt}) of EB's that need to be allocated to collaborate for the task of overcoming the obstacle in question, while allowing for the rest of the agents to continue exploring other cells/obstacles. Primarily, the parameters that affect this metric (n_{opt}) are the coefficient of friction, μ, module length, l, wheel motor torque, τ_w and finally the suboptimal spring stiffness, $k_i^{\pm}$ that we are keeping fixed. After having evaluated this problem empirically, the n_{opt} values (with the design parameters fixed as listed in Table I), for various heights h are listed in Table II.

TABLE II: Number of EB's required for different obstacle heights

#n_{opt}	#EB's	h
2	1 EB_{sl} and 2 EB_{nsl}	0 - 22 cm
3	2 EB_{sl} and 2 EB_{nsl}	22 - 37 cm
4	3 EB_{sl} and 2 EB_{nsl}	37 - 62 cm

B. Optimization formulation for allocation of Robots to Obstacles

Starting with an unknown initial state of the system, we would require to task the robots to optimally cover the map as well as climb obstacles and explore them if need be. An allocation scheme is called for to serve this purpose. Our decomposition scheme enables us to reach the obstacle as the system completes the coverage around the obstacle. A simpler cell decomposition would require additional travel to-fro unexplored obstacles after covering the cells on the flat terrain. The objective has been chosen to minimize the overall distance traveled by the system.

Preprint submitted to 2017 IEEE International Conference on Robotics and Automation. Received September 15, 2016.

$$\underset{\mathbf{A}}{\text{Minimize}} \quad \sum_{i=1}^{r}\sum_{j=1}^{o} d[i,j]A[i,j]$$

$$\begin{aligned}
\textit{subject to} \quad & (1) \text{ every entry in } A \in \{0,1\} \\
& (2)\ \forall\ k \in [1,r] : \sum_{l=1}^{o} A[k,l] = 1 \\
& (3)\ \forall\ l \in [1,o] : \sum_{k=1}^{r} A[k,l]I[k,1] = n_{opt}[l] - 1 \\
& (4)\ \forall\ l \in [1,o] : \sum_{k=1}^{r} A[k,l]I[k,2] = 1 \\
& (5)\ \forall\ l \in [1,o] : \sum_{k=1}^{r} A[k,l]I[k,3] = 1 \\
& (6)\ \forall\ l \in [1,o] : s[l]\sum_{k=1}^{r} A^2[k,l] = n_{opt}[l] \\
& (7)\ \sum_{l=1}^{0} s^2[l] = p
\end{aligned}$$

Furthermore, we have the following convention:

- $A[i,j] = 1$ if the i^{th} robot is allocated to j^{th} obstacle,
- $I[i,k] = 1$ if the i^{th} robot is of Type-k (as discussed in Section IV)
- $d[i,j]$ is the average distance from the i^{th} robot to the j^{th} obstacle,
- $n_{opt}[l]$ is the number of $EB's$ required to climb the l^{th} obstacle.

Constraints (3), (4) and (5) ensure that if an allocation of a group of robots to an obstacle happens, one Type-2, one Type-3 and the required number (n_{opt}-1) of Type-1 EB's are allocated. Constraint (7) ensures that at least p of the set of constraints in (6) are met, i.e., at least p of the o obstacles are allocated to r robots deployed. The choice of p is case dependent. We set this quantity depending on the number of obstacles present and their respective heights. This formulation is philosophically similar to allotting frontiers to robots with constraints on the number of robots allocated per frontier. This is an Integer Programming Problem, with non linear constraints. We attempt to solve this by relaxing the integer constraints and feeding to a non-linear solver ($fmincon$ in MATLAB) and rounding the solution.

In the case that there are more robots allocated than there are surrounding cells for an obstacle, the extra robots will be tasked to wait at the obstacle to climb. We will now demonstrate a case study of the algorithm presented on a map.

C. Case Study of the MAS scheme

In Fig. 9 (a), we show a generic map with polygonal obstacles. Fig. 9 (b) showcases the initial state of the system and the denotation of the distance matrix d. In this case, a scaled version of the map shown yields the
(i) Distance matrix:

$$d = \begin{bmatrix} 3.7 & 8.8 & 12.3 \\ 6.3 & 10.5 & 8.6 \\ 9.2 & 11.5 & 6.6 \\ 9.2 & 10.8 & 5.4 \\ 8.3 & 3.2 & 5.7 \\ 5 & 4.5 & 10.9 \\ 12.2 & 9.3 & 3.6 \end{bmatrix}$$

and (ii) Optimal number of EB's required Matrix:

$$n_{opt} = \begin{bmatrix} 2 & 1 & 2 \end{bmatrix}$$

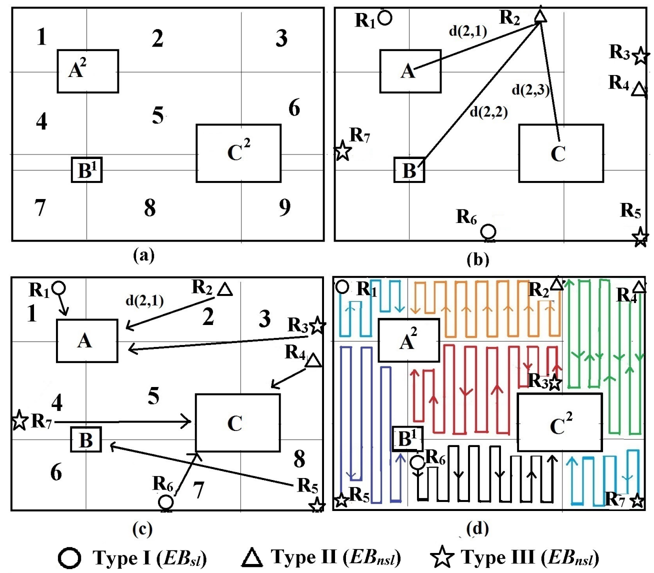

Fig. 9: (a) Cell Decomposition, (b) Denotation of the distance matrix convention in merged cell decomposition, (c) Allocation of the robots to different obstacles and (d) Boustrophedon Path Planning in each of the cells to ensure EB's converge onto the obstacle to avoid scheduling of tasks

We choose an arbitrary Identification matrix I as input:

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad A = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

Hence, 2 EB_{sl}'s, 2 EB_{nsl}'s with at threaded rod and 3 EB_{nsl}'s with a mounted nut are deployed on this map. Here, the number of robots deployed, r is 7 and the number of obstacles, o is 3. Keeping in mind all this, we set p to 2, i.e., to cover at least 2 obstacles in the first iteration of the algorithm.

The optimization routine shown in subsection IV.A was solved by relaxing the IP in a nonlinear solver and rounding yielding the Allocation Matrix A. This result is depicted in Fig. 9 (c).

This algorithm is an attempt at showing the usability of

DCMR in such exploration scenarios. We leave the work of formalizing this notion to the most generic scenarios, including cases like when new maps could be uncovered by climbing a staircase figuring as an obstacle in the map, etc., for future research.

VI. Experiments and Applications

Uneven Terrain navigation has spawned many robots that climb stairs and other obstacles in both urban and unstructured terrain. With such systems, it is a tedious job to be able to speedily explore an unknown scene, even in a non-autonomous manner, since only one unit of such robots is usually deployed in a scene. The DCMR fills this void.

1) Distributed Exploration: In a USAR scenario, the EB's can be individually controlled and can help in flagging all the liabilities/survivors-in-need-of-help.

In addition to this, the groups of EB's of the DCMR can attach and climb obstacles and reach oddly inaccessible regions in a USAR scene while the other EB's can continue exploring the flat terrain. As can be seen in Fig. 10, the DCMR is capable of carrying out concurrent obstacle and flat terrain exploration.

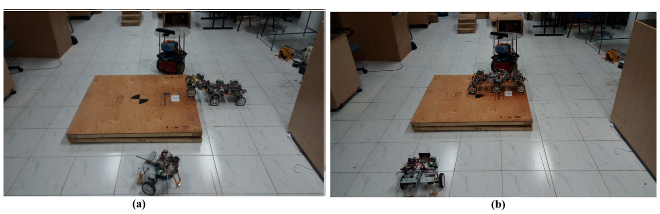

Fig. 10: An illustration of how a composite modular robot of 3 EB's can climb an obstacle large enough to merit MAS for exploration while another EB explores the flat terrain around

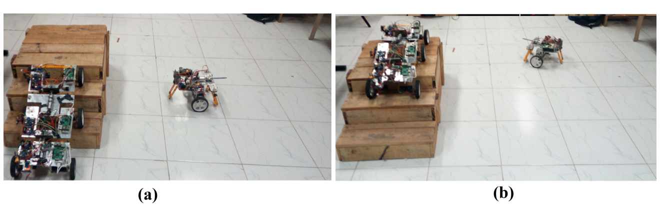

Fig. 11: An illustration of how a composite modular robot of 3 EB's can climb a staircase while another EB explores the flat terrain around

2) Multi-storey exploration: After exploring a particular storey, the individual / groups-of EB's can come together and re-attach and reach further storeys. They can be then used to repeat the exploration task by detaching again. In Fig. 12, the composite DCMR is shown to be climbing staircases of dimension typically found in an Urban setting. This was achieved at low speeds, thus successfully adhering to the static stability criterion presented in [22].

Fig. 12: Composite modular robot climbing stairs successfully

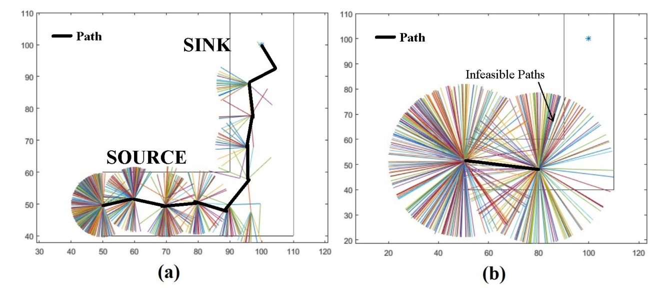

Fig. 13: (a) : RRT with obstacle avoidance (nodes that aren't feasible are abandoned) executed for an individual EB to manoeuvre this turn (b) RRT fails to converge for the case of the composite modular robot attempting to manoeuvre this turn

3) Turning in Tight Spaces: Path planning to manoeuvre crammed spaces or follow any trajectory is dependent on the robot dimensions. Bringing in variability in the robot dimensions could be greatly used to the advantage of increasing mobility. Specifically in the context of DCMR, the number of times the DCMR has to detach in order to manoeuvre a tight space can be solved with this kind of analysis. Tasked to manoeuvre a turn like that shown in Fig. 13, we sought to plan the path for different lengths of the DCMR. We implemented RRT to abandon paths that were infeasible (i.e., that led to the robot colliding with the walls). While for the robot dimensions of three modules' length, no path was found (Fig. 13 (b)), for a single module length, the RRT converged with the result shown in Fig. 13 (a)

If turning the composite DCMR by simple skidsteering of wheels on either sides of the robot is done, the entire length of the modular robot would get stuck in a tight spaced turn (Fig. 14 (a)-(b)).

However, with the capability of detaching, our limit on the area of a tight space which is navigable comes down to the be determined by the turning radius of an individual EB. This enables the EB's to navigate through tight spots like the one shown in Fig. 14 (e).

Preprint submitted to 2017 IEEE International Conference on Robotics and Automation. Received September 15, 2016.

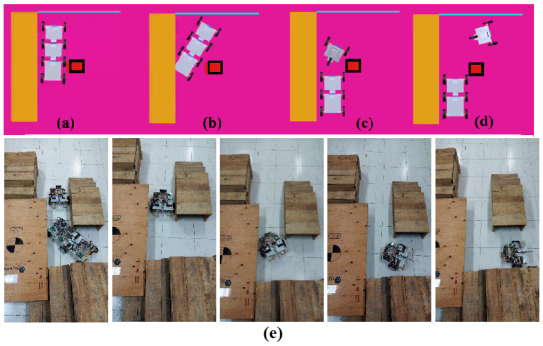

Fig. 14: (a)-(b) : Combined Modular robot unable to manoeuvre the tight turn, (c)-(d) : Individual EB's successfully passing through after detaching and (e) Experimental Demonstration of how an individual EB can maneuver tight spaces, while a composite modular robot could get jammed

VII. Conclusions and Future Work

In this work, a detachable version of the compliant modular robot which can climb obstacles including the stairs was presented. The challenges of stabilizing individual robots after detachment were discussed and the solution of Castor Wheel Support (CWS) was proposed. The CWS was designed to preserve the climbing ability after re-attachment and the scheme in which the attaching/detaching happens was described. A Multi Agent System based exploration scheme inclusive of Obstacle exploration was proposed. A prototype was developed and individual explorer-bots (EB's) were shown to attach and climb obstacles and stairs. Moreover, the use of DCMR's ability to detach and attach repeatedly in navigating crammed spaces was demonstrated. The implications of such a robot in wide ranging research topics was appreciated.

In addition to the applications presented here, we imagine that a variety of optimal control problems may be posed with the DCMR at the heart of many MAS applications. With its climbing ability, the DCMR can extend MAS applications to maps of various floors in a building. A multi-dimensional exploration problem to time-optimally maximize coverage on different storeys could thus be posed. For future direction of work, we aim to create an autonomous version of the DCMR capable of detaching and attaching with relative ease. Furthermore, we envision a system that could explore and climb autonomously.

References

[1] S. Shimoda, Y. Kuroda, and K. Iagnemma. Potential field navigation of high speed unmanned ground vehicles on uneven terrain. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2828–2833, April 2005.

[2] T Estier, Y Crausaz, Bertrand Merminod, Michel Lauria, R Piguet, and Roland Siegwart. *An innovative space rover with extended climbing abilities*. ETH-Zurich, 2000.

[3] K. Suzuki, A. Nakano, G. Endo, and S. Hirose. Development of multi-wheeled snake-like rescue robots with active elastic trunk. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4602–4607, Oct 2012.

[4] C. Wright, A. Johnson, A. Peck, Z. McCord, A. Naaktgeboren, P. Gianfortoni, M. Gonzalez-Rivero, R. Hatton, and H. Choset. Design of a modular snake robot. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2609–2614, Oct 2007.

[5] Mark Yim, Sam Homans, and Kimon Roufas. Climbing with snake-like robots. In *IFAC workshop on mobile robot technology*, pages 21–22, 2001.

[6] Masayuki Arai, Yoshinori Tanaka, Shigeo Hirose, Hiroyuki Kuwahara, and Shingo Tsukui. Development of souryu-iv and souryu-v: serially connected crawler vehicles for in-rubble searching operations. *Journal of Field Robotics*, 25(1-2):31–65, 2008.

[7] Hagen Schempf. Self-rappelling robot system for inspection and reconnaissance in search and rescue applications. *Advanced Robotics*, 23(9):1025–1056, 2009.

[8] Z. Li, S. Ma, B. Li, M. Wang, and Y. Wang. Design and basic experiments of a transformable wheel-track robot with self-adaptive mobile mechanism. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1334–1339, Oct 2010.

[9] Ehsan Mihankhah, Arash Kalantari, Ehsan Aboosaeedan, Hamid D Taghirad, S Ali, and A Moosavian. Autonomous staircase detection and stair climbing for a tracked mobile robot using fuzzy controller. In *Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on*, pages 1980–1985. IEEE, 2009.

[10] Nan Li, S. Ma, B. Li, Minghui Wang, and Y. Wang. An online stair-climbing control method for a transformable tracked robot. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 923–929, May 2012.

[11] Carlos S Casarez and Ronald S Fearing. Step climbing cooperation primitives for legged robots with a reversible connection. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3791–3798. IEEE, 2016.

[12] Fabio Luigi Bellifemine, Giovanni Caire, and Dominic Greenwood. *Developing multi-agent systems with JADE*, volume 7. John Wiley & Sons, 2007.

[13] A. Viseras, T. Wiedemann, C. Manss, L. Magel, J. Mueller, D. Shutin, and L. Merino. Decentralized multi-agent exploration with online-learning of gaussian processes. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4222–4229, May 2016.

[14] C. Leung and A. Al-Jumaily. A hybrid system for multi-agent exploration. In *Fuzzy Systems, 2004. Proceedings. 2004 IEEE International Conference on*, volume 1, pages 209–213 vol.1, July 2004.

[15] M. Baglietto, M. Paolucci, L. Scardovi, and R. Zoppoli. Information-based multi-agent exploration. In *Robot Motion and Control, 2002. RoMoCo '02. Proceedings of the Third International Workshop on*, 2002.

[16] V. Spirin and S. Cameron. Rendezvous through obstacles in multi-agent exploration. In *2014 IEEE International Symposium on Safety, Security, and Rescue Robotics (2014)*, 2014.

[17] Howie Choset. Coverage of known spaces: The boustrophedon cellular decomposition. *Autonomous Robots*, 9(3):247–253, 2000.

[18] X. Zheng, S. Koenig, D. Kempe, and S. Jain. Multirobot forest coverage for weighted and unweighted terrain. *IEEE Transactions on Robotics*, 2010.

[19] E. Gonzalez, M. Alarcon, P. Aristizabal, and C. Parra. Bsa: a coverage algorithm. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, 2003.

[20] Paul S Schenker, Paolo Pirjanian, J Balaram, KS Ali, Ashitey Trebi-Ollennu, Terrance L Huntsberger, Hrand Aghazarian, Brett A Kennedy, Eric T Baumgartner, Karl D Iagnemma, et al. Reconfigurable robots for all-terrain exploration. In *Intelligent Systems and Smart Manufacturing*, pages 454–468. International Society for Optics and Photonics, 2000.

[21] Siravuru Avinash, Ankur Srivastava, Akshaya Purohit, Suril V Shah, and K Madhava Krishna. A compliant multi-module robot for climbing big step-like obstacles. In *IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China*, 2014.

[22] S. H. Turlapati, M. Shah, S. P. Teja, A. Siravuru, S. V. Shah, and Madhava Krishna K. Stair climbing using a compliant modular robot. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 3332–3339, 2015.

[23] International Code Council. *International Building Code 2015*. ICC, Country Club Hills, Ill., 2014.

Preprint submitted to 2017 IEEE International Conference on Robotics and Automation. Received September 15, 2016.