

Sequential Trajectory Optimization for Externally-Actuated Modular Manipulators with Joint Locking

Jaeu Choe, Jeongseob Lee, Hyunsoo Yang, Hai-Nguyen Nguyen[†], and Dongjun Lee[†]

Abstract—In this paper, we present a novel trajectory planning method for externally-actuated modular manipulators (EAMMs), consisting of multiple rotor-actuated links with joints that can be either locked or unlocked. This joint-locking feature allows effective balancing of the payload capacity and dexterity of the robot but significantly complicates the planning problem by introducing binary decision variables. To address this challenge, we leverage the problem’s intrinsic structure, i.e., the payload at the end-effector being enhanced by merely locking its immediate connected links; this allows us to break down the complex planning problem into a series of manageable subproblems and solve them sequentially. Our approach significantly reduces the problem’s complexity: in a serial n -link EAMM with m joint-lock mechanisms, where there could potentially be 2^m distinct configurational dynamics, we require solving only $n + 1$ trajectory optimization problems for single rigid body dynamics sequentially, thereby rendering the problem tractable. We substantiate the efficacy of our method through various simulation and experimental studies, covering ground-free and ground-bound configurations as well as both motion-only and manipulation tasks.

I. INTRODUCTION

Recent advancements in rotor technology have led to the emergence of a new class of actuation systems. This innovation is particularly evident in the field of aerial robotics, specifically in multi-rotor drones. Several newly developed aerial systems not only exhibit high performance but also demonstrate versatility in actuating other platforms. Examples include cable-suspended payload systems [1–4], passively-joined actuated systems [5, 6], multi-link systems [7, 8].

Another research direction in this context is the development of the Externally-Actuated Modular Manipulator (EAMM) [9, 10]. Traditional robot manipulators, characterized by internal actuation, encounter limitations. With motors generating force or torque at the joints between links, there is an inherent challenge: the accumulation of payload from the end-effector reverting back to the base link. This architectural design impedes scalability, especially for expansive workspaces that necessitate potent motors at the base to compensate for the payload at the end-effector.

Conversely, EAMM circumvents this limitation. Given that every link can adequately counterbalance its own weight, a

Jaeu Choe, Jeongseob Lee, Hai-Nguyen Nguyen, and Dongjun Lee are with the Department of Mechanical Engineering, IAMD, and IOER, Seoul National University, Seoul, Republic of Korea (email: {jaeuchoe, overjs94, hanntonkin, djlee}@snu.ac.kr). Hyunsoo Yang is with Humanoids and Human Centered Mechatronics (HHCM), Istituto Italiano di Tecnologia, Genova, Italy (email: hyunsoo.yang@iit.it). Hai-Nguyen Nguyen was with LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France (email: hann@ieec.org).

[†]Corresponding author: Hai-Nguyen Nguyen, Dongjun Lee.

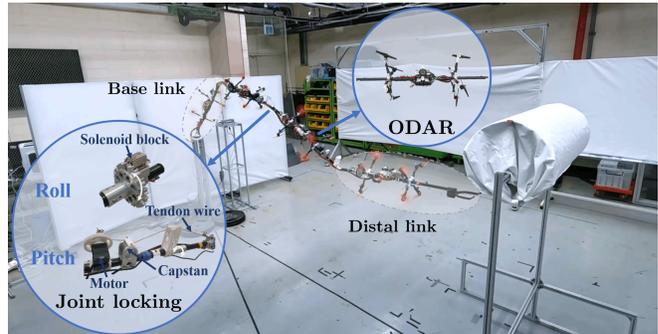


Fig. 1: A 4-link EAMM with 2 joint locking mechanisms.

quality we term as self-sufficiency, the robot can be developed to be of both lightweight and expansive workspaces. This advantage stems from empowering each module to be self-sufficient. However, a critical challenge is that EAMM’s payload capacity predominantly hinges on the capability of the distal link, i.e., the end-effector. Herein lies the rationale for introducing joint locking, allowing multiple links to collaboratively enhance the system’s payload threshold. See [11] for the detailed development of the joint locking mechanism.

The incorporation of joint locking adds a layer of complexity to the planning of the system. The binary states of being either locked or unlocked at each joint intermingle with the system’s continuous states, resulting in multiple modes of smooth dynamics based on the locking status of each joint. This hybrid nature of the dynamics poses a distinct challenge in the development of our robot, setting it apart from other existing multi-link aerial systems [7, 8]. Within the domain of aerial robotics, there are instances of hybrid systems [12]. However, successful developments have been limited and typically involve fewer modes, as seen in works on quadrotor-payload systems [13–15] with only two modes, i.e., suspending cable being loose or taut.

The hybrid nature of the dynamics is widely acknowledged as a significant hurdle in planning, a sentiment echoed in fields like locomotion and manipulation. For walking robots that have discrete contact points with the environment, trajectory generation poses complexities. A common strategy is to predefine contact sequences, e.g., with a separated footstep planning, to avoid the problem’s combinatorial nature [16]. In manipulation tasks, where contacts introduce discrete decision variables, similar planning challenges emerge. Predefined contact sequences are often impractical for contact-rich manipulation tasks, such as in-hand manipulation. Several efforts aim to address this by conducting concurrent searches across different contact modes [17–19]. However,

these approaches frequently face computational challenges and often rely on approximations, e.g., quasi-dynamics, to mitigate these computational issues [19–21].

To tackle the planning challenges stemming from the hybrid nature of the EAMM robot dynamics, we take advantage of the problem’s intrinsic structure. A key insight informs our methodology: the payload at the distal link can be increased simply by locking its immediate connectors. This allows us to sidestep the discrete locking variables and break down the complex planning problem into a series of manageable subproblems to be solved sequentially. We first start with determining the number of links requiring the locking, followed by trajectory generation for each link, taking into account obstacles and related constraints. This strategy constitutes the essence of our algorithm, namely sequential trajectory optimization.

Our methodology substantially reduces the complexity of the problem. In a serial n -link EAMM equipped with m joint lock mechanisms, there could be as many as 2^m distinct configurational dynamics, which would create daunting planning problems if using exact dynamics. Sequential trajectory optimization only requires solving $n + 1$ trajectory optimization problems for single rigid body dynamics, rendering the problem tractable. We validate and demonstrate the advantage of our approach with a number of simulation and experiment studies in various settings, ground-free and ground-bound configurations, and also for both motion-only and manipulation tasks.

II. EXTERNALLY-ACTUATED MODULAR MANIPULATOR

A. System Description

EAMM [9, 10] is a system composed of multiple modular, fully-actuated, and self-sufficient links. See Fig. 2. Each individual link is equipped with an adequate number of distributed rotors to obtain fully-actuated, i.e., able to generate wrenches in arbitrary directions, and self-sufficient, i.e., able to compensate for its own weight. For example, these rotors are arranged in such a way that each link is fully-actuated and can compensate for its own weight similar to the 8-rotor omni-directional aerial robot (ODAR) [22]. The joint with a locking mechanism connecting each sequential link permits roll and pitch motion and can be locked or unlocked using motors, as illustrated in Fig. 1. Other joints are spherical joints, allowing roll, pitch, and yaw relative motion. Further details on the design of the locking mechanism can be found in [11]. Our recently developed system, partially introduced in [10], comprises 4 fully-actuated, self-sufficient links and 2 joint locking mechanisms.

B. Dynamical Model

In Fig. 2, we have an EAMM system with n fully-actuated links and m joint locking systems. Each individual link $\{B_i\}$ is equipped with multiple rotors, enabling it to independently generate arbitrary wrenches, i.e., fully-actuated, and able to compensate for its own weight, i.e., self-sufficient. Here, we denote 1st and n -th links are respectively the base and the distal link of an n -link EAMM system.

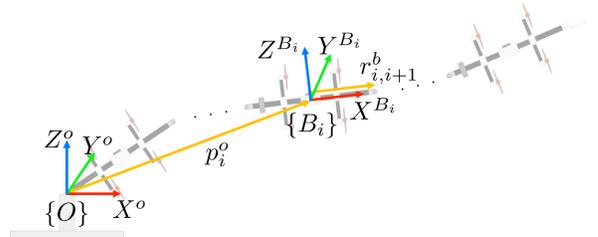


Fig. 2: Illustration of the n -link EAMM system

The relationship between wrenches acting on the link and the rotor thrust can be written by

$$\tau_i = B_i \lambda_i, \quad (1)$$

where $\tau_i \in \mathbb{R}^6$ are i -th link’s generated wrench expressed in i -th link’s body frame, $B_i \in \mathbb{R}^{6 \times 8}$ is the mapping matrix between wrench and rotor, $\lambda_i \in \mathbb{R}^8$ is rotor thrust vector of i -th 8-rotor link, e.g., using 8-rotor ODAR platform [22].

With the generated wrench, we can derive the dynamic equation for the i -th link s.t.

$$M_i(q_i) \ddot{q}_i + C_i(q_i, \dot{q}_i) \dot{q}_i + G_i(q_i) = \tau_i + f_e, \quad (2)$$

where $q_i = (p_i, r_i) \in \mathbb{R}^6$ is the position and Euler angle-based orientation of the i -th link, $M_i(q_i), C_i(q_i, \dot{q}_i), G_i(q_i) \in \mathbb{R}^6$ are the rigid body inertia, Coriolis, and gravity term, $\tau_i \in \mathbb{R}^6$ is the rotor-generated control and $f_e \in \mathbb{R}^6$ is the external wrench from other links or the payload.

With the joint lock mechanism being lightweight and its dynamics negligible, we here formulate the whole-body dynamics using the Lagrange equations as follows:

$$M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) = \tau(u) + A^T(z) \mu + F_e, \quad (3)$$

$$z_i \cdot \dot{q}_i = 0, \quad (4)$$

where $q := [\bar{q}_1, \bar{q}_2, \dots, \bar{q}_n] \in \mathbb{R}^{3n-m}$ is the generalized coordinate with $\bar{q}_i := (\phi, \theta) \in \mathbb{R}^2$ being roll-pitch angles at the joint with locking mechanism and $\bar{q}_i := (\phi, \theta, \psi) \in \mathbb{R}^3$ at $n - m$ spherical joints without locking mechanism, $M(q), C(q, \dot{q}), G(q), F_e, \tau(u)$ are the combined inertia, Coriolis, gravitation, external force and actuation term of the EAMM system, $u = [\lambda_1, \dots, \lambda_{8n}]$ is the vector of all rotor thrusts, $z \in \mathbb{Z}^m$ is set of $z_i \in [0, 1]$ which is state of joint with 1 being locked and 0 being unlocked, i.e., no constraint on combined dynamics, $A(z) \in \mathbb{R}^{2k \times (2n+n-m)}$ is the switching Jacobian matrix, and $\mu \in \mathbb{R}^{2k}$ are the Lagrange multipliers introduced by k active joint locking $z_i \cdot \dot{q}_i = 0$, following the derivation introduced in [23].

C. Problem Formulation

Given an EAMM system design, our objective is to find a trajectory that allows the robot to complete a manipulation task at the end-effector. We can formulate this objective as a constrained optimization problem as follows:

$$\begin{aligned} & \min_{q, \dot{q}, u, z} J(q, \dot{q}, u) \\ & \text{s.t. } M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) = \tau(u) + A^T(z) \mu + F_e, \\ & \quad z_i \cdot \dot{q}_i = 0, y(0) = y_{\text{init}}, y(T) = y_{\text{goal}}, \\ & \quad u \in \mathcal{U}, (q, \dot{q}) \in \mathcal{Q}, z_i \in [0, 1], \end{aligned} \quad (5)$$

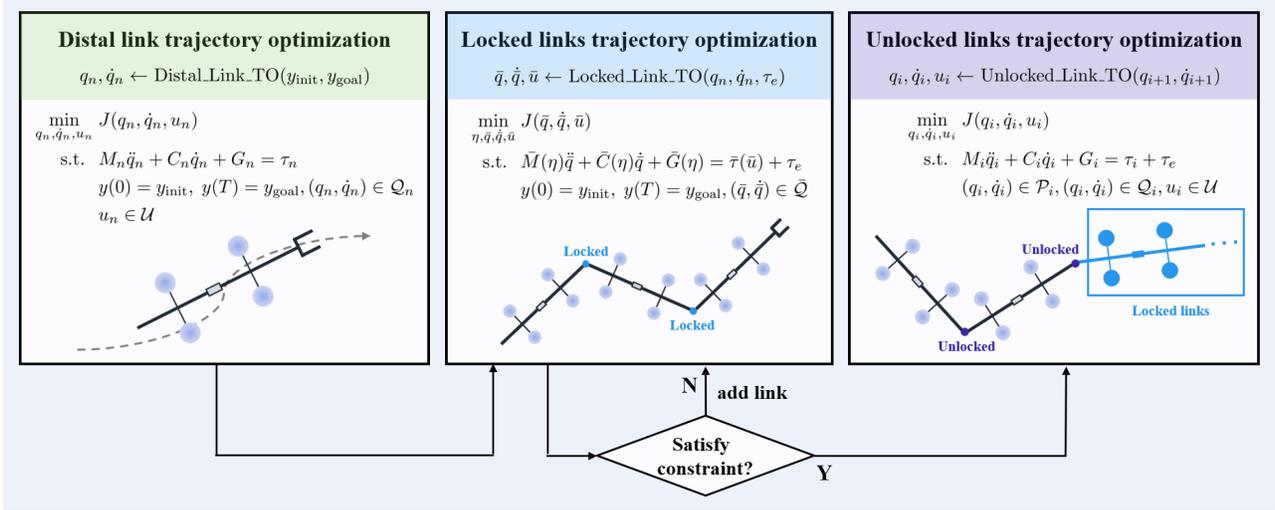


Fig. 3: Framework of sequential trajectory optimization for the EAMM system.

where J is a given objective, e.g., minimizing energy consumption, $y_{\text{init}}, y_{\text{goal}}$ is the initial and goal pose of the tool-tip of the EAMM's distal link, \mathcal{U} is feasible rotor thrust region, and \mathcal{Q} is feasible joint configuration space involved obstacle avoidance and other requirements.

The output of this optimization is the desired trajectory and control for the system, which are subsequently passed to the lower-level tracking controller. We here employ the modular control architecture developed in our previous work [9], where each link can be independently controlled through a decentralized controller.

Our optimization problem is mixed-integer, as it requires making binary integer decisions regarding the state of the joint, which can be either locked ($z_i = 1$) or unlocked ($z_i = 0$). Given that our dynamics and the primary objective are nonlinear, our problem is non-convex in its general form. The key challenge is the combinatorial nature of the dynamics. For a serial n -link EAMM equipped with m joint lock mechanisms, there could be as many as 2^m distinct configurational dynamics, rendering a daunting planning problem. In the subsequent section, we will demonstrate how to leverage the problem's structure to address this issue.

III. SEQUENTIAL TRAJECTORY OPTIMIZATION

Our primary observation regarding the problem's structure is that the distal link, due to external actuation and modular control architecture, is the key factor influencing the payload of the EAMM system. We then can enhance the payload capacity by first locking the joint nearest to the distal link and sequentially adding more links if needed. This observation allows us to separate the discrete decision, regarding the joint locking, from the optimization (5). With this strategy in mind, for a given task, we initially attempt it using only the distal link. If the thrust input required surpasses the permissible thrust limit, we attach another link to the distal link, lock the joint, and reattempt the task with the two links fixedly attached. We continue this process of incorporating links with locked joints until the locked links can together execute the task within the thrust constraint. After following

Algorithm 1 Sequential trajectory optimization

Input: Distal link initial and goal pose, and payload

Output: Trajectory and control of links and locking state

payload_sufficient \leftarrow false

$q_n, \dot{q}_n \leftarrow \text{Distal_Link_TO}(y_{\text{init}}, y_{\text{goal}})$

for $i = n$ to 1 **do**

if payload_sufficient is false **then**

$\bar{q}, \dot{\bar{q}}, \bar{u} \leftarrow \text{Locked_Link_TO}(q_n, \dot{q}_n, \tau_e)$

if $\bar{u} \leq u_{\text{limit}}$ **then**

 payload_sufficient \leftarrow true

$(q_i, \dot{q}_i) \leftarrow \text{Kinematic_Calc}(\bar{q}, \dot{\bar{q}})$

end if

else

$q_i, \dot{q}_i, u_i \leftarrow \text{Unlocked_Link_TO}(q_{i+1}, \dot{q}_{i+1})$

end if

end for

this method, for any remaining links, we include them with their joints in the unlocked state. This increases the EAMM system's degree of freedom, thereby maintaining its dexterity.

Based on the aforementioned approach, we can decompose the trajectory optimization problem for the EAMM with a joint locking system (a mixed-integer problem) into three subproblems: 1) distal link trajectory optimization, 2) locked links trajectory optimization, and 3) unlocked links trajectory optimization. The mixed-integer trajectory optimization of an n -link EAMM can then be found by sequentially solving $n+1$ trajectory optimization problems for a single rigid body. The framework for this sequential trajectory optimization is described in detail in Sec. III-A and illustrated in Fig. 3.

We solve each optimization problem using the direct collocation method paired with trapezoidal integration [24]. The specific dynamic constraints and other requirements are detailed in the following subsections. For the given payload, initial pose, and goal pose of the distal link, the algorithm of sequential trajectory optimization for n -link EAMM is illustrated in Algorithm 1.

A. Decomposed Optimization Problems

As mentioned, the general optimization problem denoted by (5) is decomposed into subproblems. Note that those subproblems are no longer mixed-integer problems.

1) *Distal link trajectory optimization*: In this subproblem, we optimize the distal link trajectory to accomplish the task of moving from the initial pose to the goal pose while respecting the environment with obstacles and the connection to the base of the EAMM system. To obtain an optimized trajectory that minimizes energy consumption, we incorporated the sum of squared thrust values into the cost function. We here only consider the dynamic of the distal link, which is modeled as a simple rigid dynamics (2). Additionally, we do not consider the payload on the optimization problem at hand. Instead, we focus on assessing the feasibility of the distal link's motion for the given task and environmental conditions. The payload is addressed in locked link trajectory optimization for configurations with joint locking states.

The optimization problem then becomes:

$$\begin{aligned} \min_{q_n, \dot{q}_n, u_n} \quad & J(q_n, \dot{q}_n, u_n) \\ \text{s.t.} \quad & M_n \ddot{q}_n + C_n \dot{q}_n + G_n = \tau_n, \\ & y(0) = y_{\text{init}}, y(T) = y_{\text{goal}}, \\ & (q_n, \dot{q}_n) \in \mathcal{Q}_n, u_n \in \mathcal{U}, \end{aligned} \quad (6)$$

where $q_n := (p_n, r_n) \in \mathbb{R}^6$ is the position and Euler angles of the distal link and u_n is the vector of rotor thrusts of the distal link. Note here that we do not have the integer variable in the optimization problem. The admissible search space \mathcal{Q}_n here encompasses both the obstacle constraints and link-base constraints, which are detailed in Sec. III-B.

The subproblem (6) is then a standard trajectory optimization problem and in our current setting, we solve this problem using the direct collocation method with trapezoidal integration [24]. The result q_n, u_n is the initial state and thrust of the distal link. This is just the initial value of the state and control of the distal link, and in the following step, the value of the trajectory and control can be adjusted to take into account the payload.

2) *Locked link trajectory optimization*: Next, we determine the required number of locked links to transport the payload, if present, while adhering to the distal link trajectory and other constraints. In this process, our objective is to minimize energy consumption. Additionally, the multiple locked links are modeled as a single rigid body. In this context, we treat the relative angles η between the locked links as variables for optimization. The problem is then stated as:

$$\begin{aligned} \min_{\eta, \bar{q}, \dot{\bar{q}}, \bar{u}} \quad & J(\bar{q}, \dot{\bar{q}}, \bar{u}) \\ \text{s.t.} \quad & \bar{M}(\eta) \ddot{\bar{q}} + \bar{C}(\eta) \dot{\bar{q}} + \bar{G}(\eta) = \bar{\tau}(\bar{u}) + \tau_e, \\ & y(0) = y_{\text{init}}, y(T) = y_{\text{goal}}, (\bar{q}, \dot{\bar{q}}) \in \bar{\mathcal{Q}}, \end{aligned} \quad (7)$$

where η is the relative angle between locked links, $\bar{q} \in \mathbb{R}^6$ is the position and Euler angle of the rigid body formed by all locked links, \bar{u} is the thrust vector of all locked links,

$\bar{M}, \bar{C}, \bar{G}, \bar{\tau}$ are inertia, Coriolis, gravity, and actuation term adjusted for locked links, $\tau_e \in \mathbb{R}^6$ is the wrench generated by the payload, and $\bar{\mathcal{Q}}$ is the obstacle and connecting to base constraints adjusted for locked links.

In this subproblem, the thrust limit is not incorporated directly into this optimization and the payload is explicitly included in the dynamic, in comparison to (6). We however check the solution of the optimization to see if the thrust for each rotor is feasible. If the thrust is feasible, the group of locked links is capable of manipulating the payload. If the thrust is not feasible, we consequentially add more locked links and repeat the optimization. This approach allows us to find the required locked links for a given payload while reducing the complexity of the optimization problem.

The output of this subproblem is the trajectory and control of the locked links. We then can compute kinematically the trajectory and control of each single link, e.g., $(q_n, u_n), (q_{n-1}, u_{n-1}), \dots, (q_k, u_{n-k})$ and pass to the lower-level controller.

3) *Unlocked link trajectory optimization*: Given the trajectory of the locked and distal links, we now can find the trajectory of other unlocked links consequentially. Here, we want to find the i -th link trajectory and control input, given $(i+1)$ -th link's trajectory. We can then model this as a trajectory optimization problem with an additional constraint ensuring the position and velocity of the i -th link kinematically following the $(i+1)$ -th link in each time step. The problem can be formulated as follows

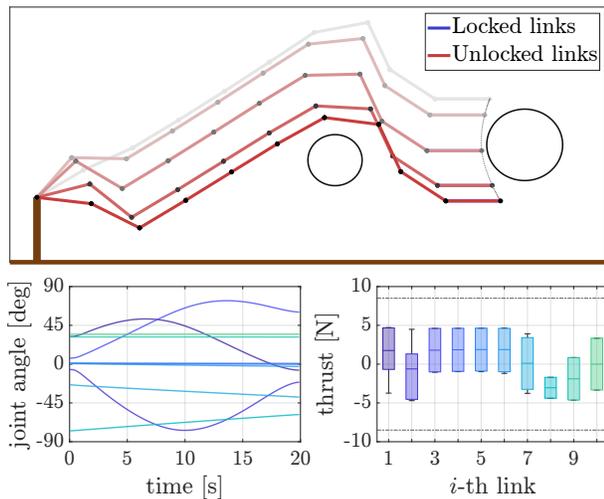
$$\begin{aligned} \min_{q_i, \dot{q}_i, u_i} \quad & J(q_i, \dot{q}_i, u_i) \\ \text{s.t.} \quad & M_i \ddot{q}_i + C_i \dot{q}_i + G_i = \tau_i + \tau_e, \\ & (q_i, \dot{q}_i) \in \mathcal{P}_i, (q_i, \dot{q}_i) \in \mathcal{Q}_i, u_i \in \mathcal{U}, \end{aligned} \quad (8)$$

where $q_i \in \mathbb{R}^6$ is the position and Euler angle of the addressed i -th link, u_i is the thrust vector of the addressed link, \mathcal{P}_i is the kinematic constraint to ensure connecting to the previous $(i+1)$ -th link in each time step, and \mathcal{Q}_i is the obstacle and connecting to base constraints adjusted for the addressed link. Note also that, here we need to add the constraint on the thrust limit of the rotor to ensure feasible control. Again, this is a standard trajectory optimization problem for a rigid body dynamic and can be solved relatively quickly.

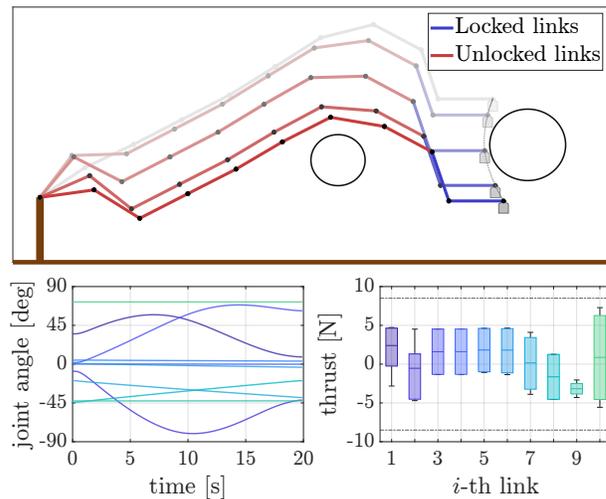
B. Additional Requirements

1) *Obstacle avoidance*: In order to conduct tasks in environments with obstacles, we integrate obstacle avoidance into the trajectory optimization process. In our current research, we only consider static obstacles in the form of ellipses. We select three points for each link and form the distance to the ellipses in order to adjust the constraints for avoiding obstacles.

2) *Link-Base constraints*: As mentioned above, the trajectory optimization for links needs to take into account the constraints to the base. There are two types of bases to consider: 1) In the case of ground-free bases like EAMM attached to a helicopter, there are no additional constraint



(a) 10-link EAMM simulation for motion-only



(b) 10-link EAMM simulation for manipulation

Fig. 4: A 10-link ground-bound EAMM performing tasks while avoiding obstacles.

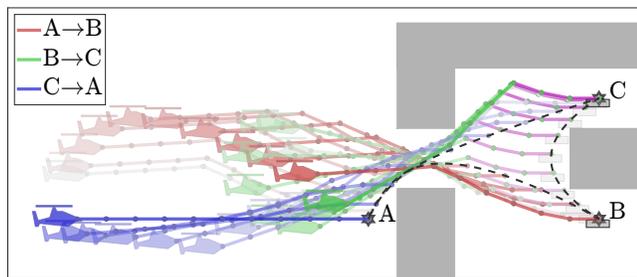


Fig. 5: A 10-link EAMM attached to a helicopter navigating through a maze to transport an object from location B to location C.

requirements. 2) For ground-bound bases, we can further categorize them into attaching to ground mobile and fixed bases. For a mobile base, the distance between the link and the ground remains smaller than the total length of the links suspended between the link and the mobile base. For a fixed base, we imposed distance constraints between the base and the link using distance constraints.

IV. SIMULATION AND EXPERIMENTAL RESULTS

In this section, we present a simulation study of our algorithm and its experimentation on a physical robot. We examine the algorithm under various conditions and settings, encompassing both ground-free scenarios and modifications for ground-bound configurations in the planar plane. A video including our numerical and experimental studies can be found here: <https://youtu.be/NgLBstCAHJQ>.

A. Trajectory Generation for Ground-free EAMM

We first tested our algorithm on a 10-link ground-free EAMM with 9 lockable joints navigating through a maze to retrieve an object. The possible configurational dynamics are 512 modes. The entire sequence is depicted in Fig. 5. Starting at point A, the robot, with all its joints unlocked, maneuvers to the object's location at point B, avoiding obstacles along the way. Subsequently, the robot locks its joints to enhance payload capacity (locked links are marked

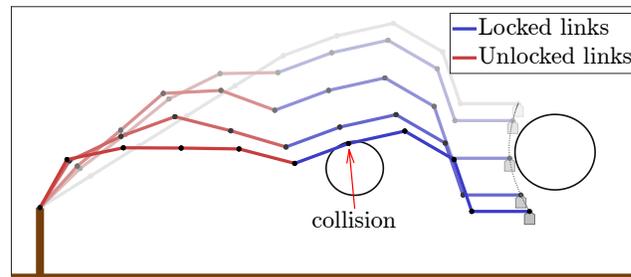


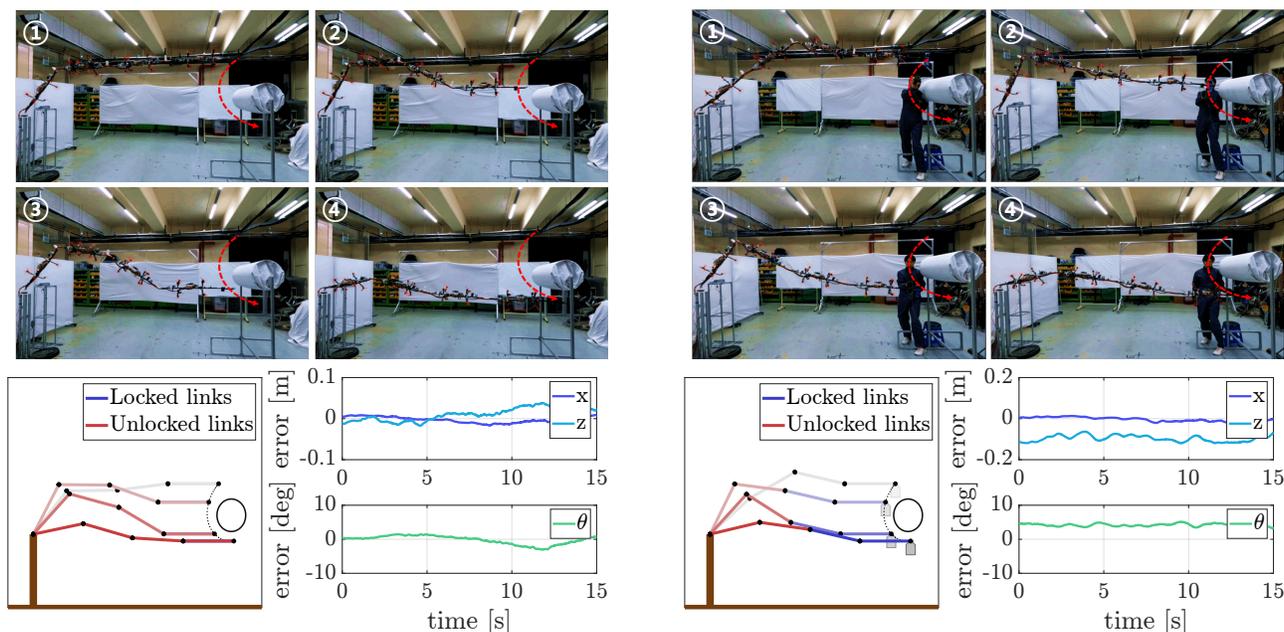
Fig. 6: A 10-link ground-bound EAMM manipulating an object with obstacle collisions

in purple), retrieves the object, and then transports it to the goal at point C. Obstacles in the maze are in gray shapes. We employed the IPOPT solver [25] paired with the CasADi [26] interface. The algorithm returned a viable mission within a computational time of 14.36 seconds for 100-node trajectory optimization and in 54.99 seconds for a 200-node scenario. In practice, we find that introducing additional sample-based pathfinding for the distal link as the algorithm's initialization tends to yield higher-quality outcomes. When initialized with RRT* [27], the total computational time is 13.31 seconds for 100 nodes and 29.24 seconds for 200 nodes, slightly faster due to effective initialization.

B. Trajectory Generation for Ground-bound EAMM

We then explored the application of our algorithm to ground-bound EAMM. In our initial scenario, we looked at a situation where one end of the robot is anchored to the ground, operating in an environment with obstacles. The robot's task was to execute a straightforward point-to-point movement, both with and without carrying an object. Through this study, we demonstrate that the algorithm can ascertain the required number of locking joints and adhere to all established constraints, encompassing robot dynamics, obstacles, payload, and being ground-bound. See Fig. 4.

The primary objective of this section is to understand the impact of the link-base distance constraints, i.e., being



(a) 4-link EAMM experiment for motion-only (b) 4-link EAMM experiment for manipulation
 Fig. 7: Simulation and experiment of a 4-link ground-bound EAMM performing tasks while avoiding obstacles.

anchored to the ground. As previously highlighted, this constraint tends to be overly relaxed. We observed instances where, even if this constraint was met, no viable solution was found. An example of this can be seen in Fig. 6, where the payload constraint and the requisite number of locked joints, particularly in the terminal link, may conflict with the ground-bound constraints. There’s a need for a more balanced approach that explicitly negotiates between the constraints and objectives associated with the distal link in relation to the ground-bound constraint. We aim to address this in our future work. Some potential solutions include: 1) finding a conservative estimate of the constraint that retains the problem’s inherent structure, 2) parameterizing the locking configurations and incorporating them into an external optimization loop, and 3) experimenting with sample-based methods, though they typically yield only asymptotic optimality.

C. Experimental Validation

In our recent developments, we have successfully engineered a 4-link ground-bound EAMM system equipped with 2 lockable joints. Each link measures 1.01 m in length, possesses 8 actuated rotors, and can handle a payload of 0.4 kg. The robot can reach within a spherical space with a radius of 4 m and has a total weight of only 7.18 kg. The low-level controller for the links employs the decentralized control law developed in [9]. This controller receives input data on the position and angular orientation of each link, which are derived from our planning algorithm. The mechanism to lock the link is separately managed by other motors.

To test our system, we developed a matching simulation. We applied our algorithm to this simulation and then transferred its outcomes to the physical robot. Both the simulation and the physical experiments were conducted for two distinct

tasks: simple motion and manipulation of an object weighing 0.72 kg, exceeding the payload capacity of a single link.

Our algorithm found a viable solution. It is also evident that our simulation aligns well with our experimental outcomes. In Fig. 7, we showcase the experiment and the simulation along with the real-time state of the robot’s system. In our current implementation, the low-level controller remains unaware of the object’s weight, leading to a drift in the z-direction as illustrated in Fig. 7 (b). Building on the successful estimation of external forces from our recent research [10], future iterations can incorporate additional payload estimation along with feedforward thrusts at the lower level to overcome this problem.

V. CONCLUSIONS

We have presented a trajectory planning algorithm for EAMMs equipped with joint locking mechanisms. Both numerical and experimental results demonstrate the effectiveness and practicality of our strategy, which leverages the problem’s inherent structure to adequately bypass its combinatorial complexity. Future research avenues include 1) refining our algorithm and its implementation for real-time system control, and 2) investigating the system’s configurational possibilities, especially the interplay between fully-actuated and under-actuated links or the arrangement of rotors within each link, where hardware and design optimization would offer valuable insights.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF2020R1A2C3010039).

REFERENCES

- [1] N. Michael, J. Fink, and V. Kumar, "Cooperative manipulation and transportation with aerial robots," *Autonomous Robots*, vol. 30, pp. 73–86, 2011.
- [2] A. Tagliabue, M. Kamel, R. Siegwart, and J. Nieto, "Robust collaborative object transportation using multiple mavs," *The International Journal of Robotics Research*, vol. 38, no. 9, pp. 1020–1044, 2019.
- [3] Y. S. Sarkisov, M. J. Kim, D. Bicego, D. Tsetserukou, C. Ott, A. Franchi, and K. Kondak, "Development of SAM: cable-suspended aerial manipulator," in *IEEE International Conference on Robotics and Automation*, 2019, pp. 5323–5329.
- [4] D. Sanalitra, H. J. Savino, M. Tognon, J. Cortés, and A. Franchi, "Full-pose manipulation control of a cable-suspended load with multiple uavs under uncertainties," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2185–2191, 2020.
- [5] H. Nguyen, S. Park, J. Park, and D. J. Lee, "A novel robotic platform for aerial manipulation using quadrotors as rotating thrust generators," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 353–369, 2018.
- [6] D. Six, S. Briot, A. Chriette, and P. Martinet, "Dynamic modelling and control of flying parallel robots," *Control Engineering Practice*, vol. 117, p. 104953, 2021.
- [7] M. Zhao, T. Anzai, F. Shi, X. Chen, K. Okada, and M. Inaba, "Design, modeling, and control of an aerial robot dragon: A dual-rotor-embedded multilink robot with the ability of multi-degree-of-freedom aerial transformation," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1176–1183, 2018.
- [8] H. Nguyen and K. Alexis, "Forceful aerial manipulation based on an aerial robotic chain: Hybrid modeling and control," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3711–3719, 2021.
- [9] H. Yang, S. Park, J. Lee, J. Ahn, D. Son, and D. J. Lee, "LASDRA: Large-size aerial skeleton system with distributed rotor actuation," in *IEEE International Conference on Robotics and Automation*, 2018, pp. 7017–7023.
- [10] Y. Kim, H. Lee, J. Lee, and D. J. Lee, "Wrench estimation of modular manipulator with external actuation and joint locking," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2023, pp. 1848–1854.
- [11] H. Lee, Y. Kim, H. Yang, F. Höpflinger, and D. J. Lee, "Development of downsized lasdra with 2-dof joint locking device," in *Aerial Robotic Systems Physically Interacting with the Environment*, 2021, pp. 1–4.
- [12] K. Sreenath and V. Kumar, "Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots," in *Robotics: Science and Systems*, 2013.
- [13] S. Tang and V. Kumar, "Mixed integer quadratic program trajectory generation for a quadrotor with a cable-suspended payload," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 2216–2222.
- [14] P. Foehn, D. Falanga, N. Kuppuswamy, R. Tedrake, and D. Scaramuzza, "Fast trajectory optimization for agile quadrotor maneuvers with a cable-suspended payload," in *Robotics: Science and Systems*, 2017.
- [15] J. Zeng, P. Kotaru, M. W. Mueller, and K. Sreenath, "Differential flatness based path planning with direct collocation on hybrid modes for a quadrotor with a cable-suspended payload," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3074–3081, 2020.
- [16] P. M. Wensing, M. Posa, Y. Hu, A. Escande, N. Mansard, and A. Del Prete, "Optimization-based control for dynamic legged robots," *IEEE Transactions on Robotics*, vol. 40, pp. 43–63, 2024.
- [17] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.
- [18] Z. Manchester, N. Doshi, R. J. Wood, and S. Kuindersma, "Contact-implicit trajectory optimization using variational integrators," *The International Journal of Robotics Research*, vol. 38, no. 12-13, pp. 1463–1476, 2019.
- [19] X. Cheng, E. Huang, Y. Hou, and M. T. Mason, "Contact mode guided motion planning for quasidynamic dexterous manipulation in 3D," in *IEEE International Conference on Robotics and Automation*, 2022, pp. 2730–2736.
- [20] B. Aceituno-Cabezas and A. Rodriguez, "A global quasi-dynamic model for contact-trajectory optimization in manipulation," in *Robotics: Science and Systems*, 2020.
- [21] T. Pang, H. J. T. Suh, L. Yang, and R. Tedrake, "Global planning for contact-rich manipulation via local smoothing of quasi-dynamic contact models," *IEEE Transactions on Robotics*, vol. 39, no. 6, pp. 4691–4711, 2023.
- [22] S. Park, J. Lee, J. Ahn, M. Kim, J. Her, G. Yang, and D. J. Lee, "ODAR: Aerial manipulation platform enabling omnidirectional wrench generation," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 4, pp. 1907–1918, 2018.
- [23] Y. Li, H. Yu, and G. S. Soh, "Hybrid dynamics of nonsmooth mechanical systems: A switching differential-algebraic formulation," in *Symposium on Mechanical Systems and Robotics*, 2022, pp. 178–187.
- [24] M. Kelly, "An introduction to trajectory optimization: How to do your own direct collocation," *SIAM Review*, vol. 59, no. 4, pp. 849–904, 2017.
- [25] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, pp. 25–57, 2006.
- [26] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.
- [27] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.