Whole Body Planning of Mobile Manipulators Leveraging Lie Theory based Optimization

William Smith[1], Siddharth Singh[2], Julia Rudy[3], and Yuxiang Guan[4]

***Abstract*—Mobile manipulators find use in several applications in distinct setups. The current issues with mobile manipulation are the large state space provided by the mobile base and the challenge of modeling high degree of freedom systems. In this work, we propose a Lie theory and optimization based approach for whole body planning of mobile manipulators to address these issues. Existing kinematics based approaches are unable to converge to an optimized joint state due to a seemingly infinite state space provided by the unconstrained motion of the base. We propose using a Lie theory based optimization framework to find the inverse kinematic constraints by converting the kinematic model, created using screw coordinates, between its Lie group and vector representation. An optimization function is devised to solve for the desired joint states of the entire mobile manipulator while including collision check constraints. This allows the motion of the mobile base and manipulator to be planned and applied in unison. The optimization framework also allows other desired conditions to be considered without large changes to the structure of the method. The performance of the whole body state planner is validated on multiple mobile manipulators in simulation with both holonomic and non-holonomic mobile bases. Our solver is available with further derivations and results at https://github.com/peleito/slithers**

***Index Terms*—mobile manipulation, lie theory, optimization, trajectory planning, motion planning**

I. Introduction

Mobile manipulation is a fundamental task in robotics that has valuable applications in various fields. Mobile manipulators were first used for space exploration on the surface of celestial bodies [1] where the robots were teleoperated. The importance of space robotics to perform in remote locations has increased research of online methods without the use of an operator [2]. In recent decades, with the proliferation of robotic systems, mobile manipulators have found use in several domains such as construction [3], agriculture [4], additive manufacturing [5], telehealth [6], mapping [7, 8], etc. The presence of mobile manipulators is also growing in the manufacturing setting as industries switch to a smart factory system [9] with a focus on reliability [10] and scalability [11]. New companies have even been formed to push mobile manipulators to low level consumers in offices or homes [12]. Due to the many use cases and immense potential of mobile manipulators, it is imperative to provide solutions for state planning to aid in its practical use.

A common problem for all mobile manipulation tasks is maneuvering the mobile base and the manipulator in unison to complete a task. Attempts have been made to integrate a robotic manipulator with a mobile platform, but the motion of two robots is typically controlled separately [13] leading to sporadic movement of the entire robotic system. This paper proposes the motion of a robotic manipulator and mobile platform be controlled in unison, increasing the usability of mobile manipulators in practical applications.

A. Previous Work

The existing methods of state planning for mobile manipulators can be broadly classified into two approaches. The first approach is kinematic based planning methods that solve all of the joint states for the robot to follow. The states are then connected with conventional planning methods and interpolated to create a finely sampled path in state space. In the second approach, the problem can be separated into the motion of the mobile base and the motion of the manipulator. This allows for individualized methods to be used for planning the states of the mobile base and manipulator separately and then synchronizing the solutions.

For the first approaches, many kinematic based planning methods start by finding the optimal desired state in state space based on a given pose goal. The path is then converted into joint states using inverse kinematics [14] or optimization based inverse kinematics [15, 16]. Due to the under constrained nature of inverse kinematics, optimization based approaches [17, 18] have been pursued when the joint motions are fully constrained. The states are then connected using a planning algorithm, such as rapidly exploring random trees (RRT) [19] and its variants. Many different techniques have been used to find the optimal desired states including evolutionary algorithms [20] and constrained inverse kinematics [21, 22]. Other techniques have been implemented in MoveIt [23], an open-source framework for manipulator motion planning, allowing the user to pick and choose the best methods along each step. These formulations emphasize the final state of the mobile manipulator and less concern about the path followed along the way.

In the second approaches, the motion of mobile manipulators and bases are separated, combining traditional manipulator state planning techniques with classical mobile robot path planning methods to reduce the complexity of the planning problem. Simply the mobile base is maneuvered towards

[1]Analytical and Spatial Intelligence Research Team, Applied Research Associates, Raleigh NC, 27612, USA wbs3ra@virginia.edu

[2]Department of Mechanical and Aerospace Engineering, University of Virginia, Charlottesville VA, 22903, USA sks4zk@virginia.edu

[3]Department of Electrical and Computer Engineering, University of Virginia, Charlottesville VA, 22903, USA jr7yj@virginia.edu

[4]Department of Mechanical Engineering, The University of Texas at Dallas, Richardson TX, 75080, USA yuxiang.guan@utdallas.edu

regions to maximize the manipulability for the respective task [24, 25]. This leads to a discretized path that has the base stop frequently to allow for the manipulator to actuate [26, 27]. Many approaches aim to determine the optimal positions for the mobile platform to separate the motion of the base and manipulator [28]. The optimal base positions are typically determined by trying to minimize joint motion [29, 30] and minimize unnecessary base motion [25]. These methods then make use of conventional inverse kinematics [31] to actuate the manipulator. Since these techniques mostly focus on moving from one state to another and do not consider the continuous motion along the path, it is difficult for the robot to smoothly follow long paths.

The two different approaches are appropriate for planning mobile manipulator states with low degrees of freedom but have limited scalability to higher degrees of freedom. Consequently, these methods are not suited for problems that require mobile manipulation in large work environments, such as construction [32] or agriculture [33]. For such systems, the motion of the base is often planned separately due to the large workspace containing many state solutions to any given desired end effector pose. The separated motion will increase wasted time and disrupt smooth end effector motion as the robot stops to allow for proper positioning of the manipulator. Therefore a state planner for a mobile manipulator with continuous and unified motion of the base and manipulator is desired to improve the performance of maneuvering in unrestricted environments.

B. Our Contributions

This paper presents a Lie theory based optimization approach for whole body planning of mobile manipulators. Given a sequence of desired end effector poses, the proposed method solves for the joint values incrementally using kinematic constraints found through Lie theory. The joint state values are estimated by solving a constrained optimization problem which yields the sequence of states given the current pose and desired pose. The cost function is modified to ensure the smoothness of the joint states. In this study, our contribution is three-fold: 1) firstly, we successfully developed a new whole body kinematic planner given unrestricted motion in high degrees of freedom, 2) the proposed method is generalizable and can be easily adjusted for mobile manipulators of differing degrees of freedom, 3) and lastly the proposed method ensuring smoothness in the joint states with unified motion between the base and manipulator by minimizing jerk in the motion trajectory.

Additionally the proposed method also seamlessly integrates self-collision check to ensure feasibility. The proposed optimization framework also allows to choose between distinct constraint types such as minimizing energy, time or jerk making highlighting the tractability of the proposed approach. The proposed method is validated qualitatively and quantitatively in a simulated experiment on both holonomic and non-holonomic mobile bases. A comprehensive analysis is performed on the proposed method by testing different robotic setups against different desired end effector poses. In the validation step, it is found that our method can follow all the given paths with significantly high accuracy and considerable smoothness.

This paper is organized as follows. Section II describes the forward kinematics of a manipulator using screw coordinates and Lie theory. Section III presents our proposed approach, the optimization based state planner for mobile manipulators. Experimental validation is presented in Section IV, and Section V summarizes conclusions and ongoing work.

II. Background: Manipulator Kinematics

Kinematics describes the motion of bodies without considering the forces or moments that cause the motion [34]. It is essential to formulate suitable kinematic models for robotic systems to analyze the behavior of robot manipulators. The robot kinematics are composed of forward kinematics [35], finding the end effector state given joint states, and inverse kinematics, finding the joint states given the end effector state. To develop a kinematic model of the mobile manipulator, the physical parameters of the robot must be measured. The Denavit-Hartenberg (DH) [36] method utilizes four parameters and is the most common method for deriving the homogeneous transformation matrices of a robotic manipulator. Even though the process for determining the kinematics of a manipulator is simplified through the DH method, the process requires that a strict convention be followed when assigning body fixed frames to all of the joints.

Conventional kinematic models are constructed by analyzing the multi-link system and assembling a product of transformation matrices, which is unsuitable for high degree of freedom mobile manipulation. By contrast, Lie theory's generic format makes it easier to model manipulator kinematics and dynamics for better planning and control. Since a mobile manipulator consists of both a manipulator and a mobile base, the forward kinematics must be calculated for both the base and manipulator. In this section, Lie theory based forward kinematics are shown for constructing the kinematic model of a manipulator.

A. Screw Coordinates

Conventionally, homogeneous transformations found using DH parameters are used to represent the relationship between two robot links. Instead, screw coordinates can be used to define the motion for all of the joints in a robotic manipulator. Screw theory [37] helps define the position and motion of rigid bodies with respect to a fixed reference frame by classifying all motion as a screw coordinate system which is combination of rotation and translation. This allows for a generic method to be applied to all motions regardless of the degrees of freedom associated with the motion. Body fixed frames are not necessary to be defined as only one frame representing the fixed reference frame is necessary. A twist, $\mathbf{S} \in \mathbb{R}^6$, is a screw composed of two three dimensional vectors. It represents the rotational, $\mathbf{S}_\omega$, and translational, $\mathbf{S}_v$, motion about an axis as given by

$$\mathbf{S} = \begin{Bmatrix} \boldsymbol{\omega} \\ \mathbf{v} + \mathbf{d} \times \boldsymbol{\omega} \end{Bmatrix}, \tag{1}$$

where $\boldsymbol{\omega} \in \mathbb{R}^3$ is the angular motion, $\mathbf{v} \in \mathbb{R}^3$ is the linear motion, and $\mathbf{d} \in \mathbb{R}^3$ is a translation from the reference frame to the joint. Since screw joints can be used to represent all of the joints of a mobile manipulator, screw theory can be used to represent the kinematic relationships of the entire robot.

Screw theory is useful for forward and inverse kinematics since revolute and prismatic joints are represented in the same format. Since most actuators only provide one degree of freedom, many systems can be divided into a set of revolute and prismatic joints corresponding to the total motion. In a revolute joint ($\mathbf{v} = \mathbf{0}$), the twist element, $\mathbf{S}_\omega$, is defined by rotational motion and translational vectors as given by

$$\mathbf{S}_\omega = \begin{Bmatrix} \boldsymbol{\omega} \\ \mathbf{d} \times \boldsymbol{\omega} \end{Bmatrix}. \tag{2}$$

A prismatic joint ($\boldsymbol{\omega} = \mathbf{0}$) has a twist element, $\mathbf{S}_v$, defined by the linear motion vector as given by

$$\mathbf{S}_v = \begin{Bmatrix} \mathbf{0} \\ \mathbf{v} \end{Bmatrix}. \tag{3}$$

The defined screw coordinates can then be used to compute the forward and inverse kinematics from the fundamentals of Lie theory using the product of exponentials.

B. Lie Groups and Lie Algebra

Lie theory is the foundation that relates the Lie group and its corresponding Lie algebra through the exponential map. Lie groups are special smooth manifolds commonly found in the field of robotics [38]. Typical groups in robotics are for rotations defined by the special orthogonal group, $SO(n)$, and rigid motion defined by the special Euclidean group, $SE(n)$. Through the matrix exponential, the motion at a given time can transform the current pose into the next pose, as seen in Figure 1. The matrix exponential directly maps the motion from the Lie algebra to the Lie group enabling fundamental mathematical operations to be performed in the tangent space and then mapped back to the group. One advantage of performing planning on the Lie algebra is the ease of finding the shortest path between points as compared to solving for a manifold. On the Lie algebra, the shortest path is the line connecting the two points which represents the geodesic when mapped back to the Lie group. In the case of a three dimensional rigid body transformation, $[\mathbf{P}] \in SE(3)$, any element in the Lie group can be defined by a rotation matrix, $[\mathbf{R}]$, and translation vector, $\mathbf{t}$, as given by

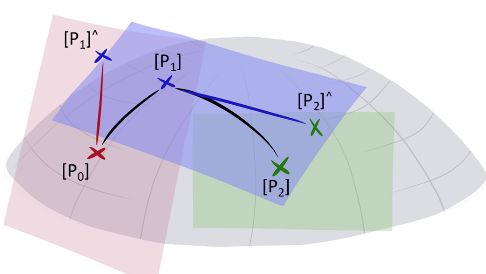

Fig. 1. Illustrative representation of using Lie theory to move across the Lie group using the corresponding Lie algebra. The motion in between each point on the Lie group (manifold) can be represented as a motion on the corresponding Lie algebra (tangent surface).

$$[\mathbf{P}] = \begin{bmatrix} [\mathbf{R}] & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{4}$$

The corresponding Lie algebra can be found by creating a tangent space on the group manifold at the group identity. Due to the properties of a Lie group, the tangent space at any point on the manifold is identical and therefore the Lie algebra can be used to represent the tangent space of any point on the manifold. The Lie algebra, $[\mathbf{S}]^\wedge \in \mathfrak{se}(3)$, for a homogeneous transformation is defined by the twist governing the motion and given by

$$[\mathbf{S}]^\wedge = \begin{bmatrix} [\boldsymbol{\omega}]_\times & \mathbf{v} + \mathbf{d} \times \boldsymbol{\omega} \\ \mathbf{0} & 0 \end{bmatrix}, \tag{5}$$

where $[\cdot]_\times$ indicates a skew symmetric matrix.

Lie theory is especially helpful for forward kinematics of robotic manipulators as joints are broken down into purely revolute and prismatic joints. The matrix exponential is used to map $[\mathbf{S}(q)]^\wedge$ onto the $SE(3)$ manifold changing the pose from $[\mathbf{P}_0]$ to $[\mathbf{P}]$ as given by

$$[\mathbf{P}] = e^{[\mathbf{S}(\mathbf{q})]^\wedge}[\mathbf{P}_0], \tag{6}$$

where $[\mathbf{P}_0]$ is the configuration state at the identity and S is a function of q, the joint state. Since the twist coordinates are defined with respect to a fixed reference frame, right hand multiplying $[\mathbf{P}_0]$ ensures $[\mathbf{S}(q)]^\wedge$ is time invariant. When there is a kinematic chain of rigid links as in a manipulator, the product of exponentials is used to solve for the forward kinematics along the entire chain as given by

$$[\mathbf{P}_n^0] = \prod_{i \in n} \left(e^{[\mathbf{S}_i(q_i)]^\wedge} \right) [\mathbf{P}_0], \tag{7}$$

where n is the number of links in the robotic system. It can be seen that the forward kinematics takes a similar form as compared to using DH parameters but allows for all motion, including the motion of the mobile base, to be modeled in the same manner.

III. Proposed Method: Whole Body Planning using Lie Theory Kinematics

Figure 2 shows the basic schematic for the proposed state planning method. Let us assume the desired path of the end effector is specified and given by $\mathcal{P}^*_{0:T} = \{[\mathbf{P}_n^w]^*(0), [\mathbf{P}_n^w]^*(\Delta t), \cdots, [\mathbf{P}_n^w]^*(T)\}$ where $[\mathbf{P}_n^w](t) \in SE(3)$ is the pose of the end effector at any given time. The time sampling duration is represented by Δt and the total time duration is represented by T. At any given time the state of the robot is defined as $\mathbf{x}_r = [\mathbf{q}, \dot{\mathbf{q}}, \mathbf{t}_r, \mathbf{v}_r, \boldsymbol{\theta}_r, \boldsymbol{\omega}_r]^\top$. Here

$\mathbf{q} = [\theta_1, \theta_2, \cdots, \theta_n] \in \mathbb{R}^n$ represents the state vector of the manipulator in joint space and consequently, $\dot{\mathbf{q}}$ represents the joint velocity vector. The mobile base position in the world coordinates is represented by $\mathbf{t} \in \mathbb{R}^3$ and $\mathbf{v} \in \mathbb{R}^3$ represents the mobile base velocity. The global orientation and global angular velocity of the mobile base are represented by $\boldsymbol{\theta}_r$ and $\boldsymbol{\omega}_r$, respectively. The current state of the robot is assumed to be known with minimal uncertainty from joint encoders and state estimation of the base. Onboard motion controllers will then take the desired state to determine the low level input based on the dynamical model. After the elapsed time, the desired states will be computed again until all of the desired poses are reached.

A proper inverse kinematics solver should be grounded in a strong forward kinematic model of the robot. Since the inverse kinematics of a mobile manipulator are rank deficient, the forward kinematics can be used as the basis for an optimization problem. Analytical and numerical methods are the two main techniques used to solve inverse kinematics problems. The joint variables can be calculated analytically based on given configuration data but there is not always a solution due to the rank of the resulting system of equations. Therefore, a numerical method is pursued to estimate the joint variables required to achieve the desired state. Several optimization methods could be utilized in state planning, such as $minimum\ time$, $minimum\ energy$, and $minimum\ jerk$. In the following subsection, an optimization method to achieve $minimum\ jerk$ is proposed to achieve a desired pose while maintaining the smooth motion of the base and joints.

A. Inverse Kinematic Constraints Using Lie Theory

The first step in state planning is to use inverse kinematics to determine the required joint states to achieve the desired pose goals. Lie theory and screw coordinates are useful for computing the inverse kinematics since it uses the change in states which is naturally suited for executing planned states. It also takes advantage of connecting poses along the geodesic of the manifold, reducing major unnecessary motion. The transition from forward kinematics to inverse kinematics is intuitive since the state variables can be easily separated from the pose variables.

First, the forward kinematics of the robotic system are represented using the product of exponentials as seen in Equation 7, but are adapted to consider a time series of motion by introducing time steps denoted by the subscript k. The screw coordinates and configuration state are defined with respect to a body fixed frame on the base of the robot to ensure they are time invariant. The entire pose is then projected into the world frame through the transformation from the base frame to the world frame. The forward kinematics to the next pose, $[\mathbf{P}^w_{n,k+1}]$, are now given by

$$[\mathbf{P}^w_{n,k+1}] = [\mathbf{P}^w_{r,k}] e^{[\mathbf{S}_r(\mathbf{v}_{r,k+1}, \boldsymbol{\omega}_{r,k+1})]^\wedge} * \prod_{i \in n} \left(e^{[\mathbf{S}_i(q_{i,k+1})]^\wedge} \right) [\mathbf{P}_0], \quad (8)$$

where $[\mathbf{P}^w_{r,k}]$ is the transformation from the base frame of the robot to the world frame, $\mathbf{S}_r$ is the screw coordinates for the robotic base, and n is the number of links in the robotic manipulator. In the case of a mobile manipulator, the Lie algebra, $[\mathbf{S}_r(\mathbf{v}_{r,k+1}, \boldsymbol{\omega}_{r,k+1})]^\wedge$, represents the degrees of freedom provided by the mobile platform. Common mobile platform configurations provide one to three degrees of freedom.

The inverse kinematics can be solved by isolating the joint states from the poses to yield

$$[\mathbf{P}^w_{r,k}]^{-1}[\mathbf{P}^w_{n,k+1}][\mathbf{P}_0]^{-1} = e^{[\mathbf{S}_r(\mathbf{v}_{r,k+1}, \boldsymbol{\omega}_{r,k+1})]^\wedge} \prod_{i \in n} \left(e^{[\mathbf{S}_i(q_{i,k+1})]^\wedge} \right). \quad (9)$$

The resulting system is complex to solve and difficult to isolate the state variables using basic linear algebra. Therefore, the matrices are mapped to vector space, $\mathbb{R}^6$, using the Log operator as given by

$$\boldsymbol{\tau} = \mathrm{Log}\left([\mathbf{P}]\right) = \log\left([\mathbf{P}]\right)^\vee, \quad (10)$$

where $\boldsymbol{\tau} \in \mathbb{R}^6$ is the screw coordinate representation comprising the rotation and translation between poses. The Log operator maps between the manifold and vector space. The entities on the manifold are first mapped to the tangent space with the matrix logarithm, $\log(\cdot)$. They are then mapped from the tangent space to the vector space with the vee operator, $(\cdot)^\vee$. Therefore the inverse kinematic constraints in vector form are denoted by

$$\mathrm{Log}\left([\mathbf{P}^w_{r,k}]^{-1}[\mathbf{P}^w_{n,k+1}][\mathbf{P}_0]^{-1}\right) = \mathrm{Log}\left(e^{[\mathbf{S}_r(\mathbf{v}_{r,k+1}, \boldsymbol{\omega}_{r,k+1})]^\wedge} \prod_{i \in n} \left(e^{[\mathbf{S}_i(q_{i,k+1})]^\wedge} \right) \right). \quad (11)$$

For simplification purposes let

$$\boldsymbol{\tau}_{poses}\left([\mathbf{P}^w_{n,k+1}]\right) = \mathrm{Log}\left([\mathbf{P}^w_{r,k}]^{-1}[\mathbf{P}^w_{n,k+1}][\mathbf{P}_0]^{-1}\right), \quad (12)$$

where $\boldsymbol{\tau}_{poses}\left([\mathbf{P}^w_{n,k+1}]\right)$ represents the pose transformations and let

$$\boldsymbol{\tau}_{joints}\left(\mathbf{u}_{k+1}\right) = \mathrm{Log}\left(e^{[\mathbf{S}_r(\mathbf{v}_{r,k+1}, \boldsymbol{\omega}_{r,k+1})]^\wedge} \prod_{i \in n} \left(e^{[\mathbf{S}_i(q_{i,k+1})]^\wedge} \right) \right), \quad (13)$$

where $\boldsymbol{\tau}_{joints}\left(\mathbf{u}_{k+1}\right)$ represents the joint transformations in vector form and $\mathbf{u}_{k+1} = \{\mathbf{q}_{k+1}, \mathbf{v}_{r,k+1}, \boldsymbol{\omega}_{r,k+1}\}$ is the input vector. The resulting augmented matrix made up of the two vectors cannot be solved using elementary linear algebra because it is nonlinear. Analytical methods are not pursued since the matrix is rank deficient with free variables, so a numerical optimization technique will be used to solve for the joint states.

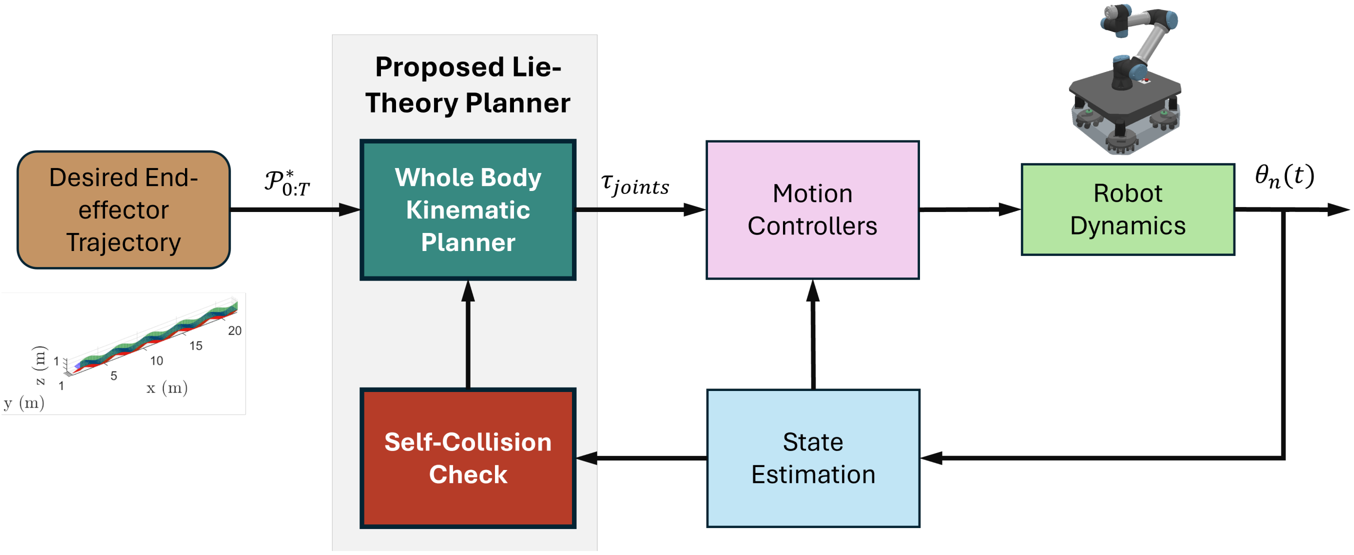

Fig. 2. The framework of the proposed method takes in a set of desired end effector poses, computes the desired state based on an objective function and executes the motion to achieve the desired states.

B. Optimization Based Inverse Kinematics

The optimization problem is formulated to solve for the joint variables with the main constraint derived from the inverse kinematics in Equation 11. The objective function can be created and adapted for the specific task of the robot with the addition of other constraints. Since different tasks expect different performance from the mobile manipulator, it is best to start with the generic formulation. The basic formulation of the optimization problem is given by

$$\begin{aligned} \underset{\mathbf{u}^*_{k+1}}{\text{argmin}} \quad & \text{desired constraints} \\ \text{s.t.} \quad & \text{required constraints} \end{aligned} \tag{14}$$

where $\mathbf{u}^*_{k+1}$ is the optimization variable representing the next desired state of the mobile manipulator, 'desired constraints' are constraints that can be afforded with an associated cost λ, and 'required constraints' are constraints that can not be afforded. The state consists of the controllable parameters for the mobile robot including the linear and angular velocities of the mobile base, and the joint values of the manipulator.

As the optimization problem is made more specific for the task, the set of available states is reduced. The simplest formulation is minimizing the pose error between the next point in the path and the end effector by reformulating Equation 11 as a difference in the objective function without violating joint constraints. The poses can be achieved by setting the pose error as a 'desired constraint', which minimizes the predicted pose error between the forward kinematics of the end effector and the desired pose. Joint constraints are 'required constraints' because the mobile manipulator can not physically occupy positions outside of the minimum and maximum joint values. For a simple task with no contact forces, the states can be planned to follow the poses in $\mathcal{P}^*_{0:T}$ by

$$\begin{aligned} \underset{\mathbf{u}^*_{k+1}}{\text{argmin}} \quad & \left\| \boldsymbol{\tau}_{joints}(\mathbf{u}^*_{k+1}) - \boldsymbol{\tau}_{poses}\left([\mathbf{P}^w_{n,k+1}]\right) \right\|_2 \\ \text{s.t.} \quad & \mathbf{u}^*_{k+1} \geq \mathbf{u}_{\min} \\ & \mathbf{u}^*_{k+1} \leq \mathbf{u}_{\max}. \end{aligned} \tag{15}$$

The 'required constraints' are simply the joint limits given by $\mathbf{u}_{\min}$ and $\mathbf{u}_{\max}$. The current formulation can guarantee state smoothness between individual points, but to achieve continuous operation other constraints must be considered. An issue with such a basic formulation is the state smoothness of the joint solutions across the entire path, which is not currently considered at all. The pose error is likely to approach zero from the optimized joint values but the motion will not necessarily be feasible with many self collisions and joint discontinuities present.

Self collisions must be prevented to ensure the manipulator can move from state to state without obstruction. The redundancy in state motion and preplanned end effector path allows for collisions to be avoided by ensuring the desired state is collision free. Collision detection between two bodies [39] has been well studied with a focus on manipulators [40]. Given the joint states, the collision detection algorithm will return the collision status of the mobile manipulator as given by

$$\text{detectSelfCollision}(\mathbf{q}) = \begin{cases} 1 & \text{if collision} \\ 0 & \text{if no collision} \end{cases} \tag{16}$$

where detectSelfCollision is the self collision detection algorithm [41] and $\mathbf{q}$ is the state variable for the manipulator. To improve the speed and performance capabilities for optimizing, an ellipsoid approximation is used for the collision geometry, and only manipulator joints affecting possible collisions are checked. The self collision detection will be a 'required constraint' in the final optimization function since the motion cannot be executed if the desired state leads to a collision.

To combat the smoothness issues present in the basic formulation, several constraints can be added as 'desired constraints' to make the joint states more continuous. Since the time between points is given and the smoothness is of high priority, an optimization method to achieve *minimum jerk* is proposed to plan states for following a path. If *minimum jerk* is achieved along each joint, the robot will maintain smooth motion of the mobile platform and manipulator. Smoothness is ensured by adding two 'desired constraints' to the problem, through minimizing the motion of each joint and minimizing the jerk of each joint. Minimizing the motion of each joint increases the continuity of the computed states and ensures the robot does not make any unnecessary movements between poses. The jerk of the joints, $\dddot{\mathbf{u}}^*_{k+1}$, ensures the joints transition smoothly from one state to another along the length of the path. The objective function is reformulated to achieve smooth state planning as follows

$$\begin{aligned} \underset{\mathbf{u}^*_{k+1}}{\operatorname{argmin}} \quad & \lambda_e \left\| \boldsymbol{\tau}_{joints}(\mathbf{u}^*_{k+1}) - \boldsymbol{\tau}_{poses}\left([\mathbf{P}^w_{n,k+1}]\right) \right\|_2 \\ & + \left\| \boldsymbol{\lambda}_v \odot \left(\mathbf{u}^*_{k+1} - \mathbf{u}_k\right) \right\|_2 + \lambda_j \left\| \dddot{\mathbf{u}}^*_{k+1} \right\|_2 \\ \text{s.t.} \quad & \text{detectSelfCollision}(\mathbf{u}^*_{k+1}) = 0 \\ & \mathbf{u}^*_{k+1} \geq \mathbf{u}_{\min} \\ & \mathbf{u}^*_{k+1} \leq \mathbf{u}_{\max}, \end{aligned} \tag{17}$$

where $[\mathbf{A}] \odot [\mathbf{B}]$ is the Hadamard product and $\mathbf{u}_k$ is the current configuration state. The other 'desired constraints' are used for state smoothing, so the path is feasible for low level controllers to follow. The 'required constraints' simply remain the joint limits, given by $\mathbf{u}_{\min}$ and $\mathbf{u}_{\max}$, with the addition of the self collision detector. The joint state continuity is controlled by $\left\| \boldsymbol{\lambda}_v \odot \mathbf{u}^*_{k+1} \right\|_2$ through limiting the motion of each joint in a weighted fashion. The jerk of the system, $\dddot{\mathbf{u}}^*_{k+1}$, is approximated by backwards finite differences as given by

$$\dddot{\mathbf{u}}^*_{k+1} \approx \frac{\frac{5}{2}\mathbf{u}^*_{k+1} - 9\mathbf{u}_k + 12\mathbf{u}_{k-1} - 7\mathbf{u}_{k-2} + \frac{3}{2}\mathbf{u}_{k-3}}{\Delta t^3}, \tag{18}$$

where Δt is the duration of each time step and ensures the joint motion is not sporadic compared to the previous states. It is to be noted that λ_e, $\boldsymbol{\lambda}_v$, and λ_j are scaling factors representing the relative importance.

The goal pose is updated and the desired states are executed by low-level controllers to achieve the necessary motion to reach the desired states. The next states are to be solved in a step wise manner for the entire length of the path.

IV. Experimental Validation

The proposed method of Lie theory based optimization state planning was tested on simulated mobile manipulators. The experiment was conducted to determine the feasibility of the proposed method as a state planner by ensuring success on varying path types and various robotic systems. A successfully planned state has high pose accuracy and no violated constraints. It was assumed the system was fully controllable and observable with minimal sensor noise. The planning performance of the proposed method was validated by accurately and smoothly following various end effector paths. The goal was for all of the joint states for the current time step to be planned smoothly and efficiently to achieve the next desired pose. The experimental setup ensured the proposed method is universal with tests conducted on various robotic setups and complex paths.

The proposed method was experimentally validated by having the robot follow various paths for the end effector with different robot platforms. The robot was initialized at a different initial pose than the starting pose of the path, so the robot had to first travel to the path before continuing to the next point. The remaining states would then be solved in a step wise manner for the entire length of the path. Two mobile manipulators with different base configurations were used for experimentation. The robots tested were an industrial collaborative manipulator on a non-holonomic mobile platform and the same manipulator on a holonomic mobile platform, as seen in Figure 3. The manipulator was a six degree of freedom industrial collaborative manipulator, the non-holonomic platform was a differential drive mobile robot, and the holonomic platform was an x-drive robot. Figure 4 shows the three test paths that were used, namely, vertical helix, sine wave, and horizontal helix. The test paths used have non-zero derivatives to test the continuity of states planned for differing path configurations. Table I lists the parameters used for the experiment, which were chosen to achieve the goal with feasible and continuous states.

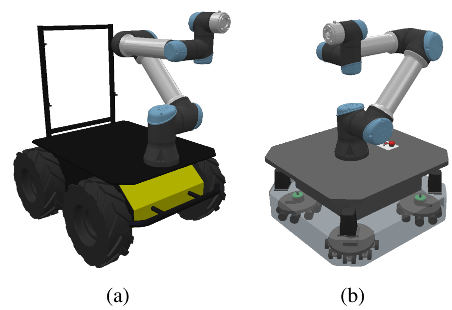

Fig. 3. Mobile manipulators consisting of a six degree of freedom industrial robotic manipulator mounted on a (a) non-holonomic mobile platform (Clearpath Robotics Husky) and (b) holonomic mobile platform.

The results of the experiment show that the proposed method is a successful inverse kinematic solver, suitable for a variety of robot and path configurations. The error for

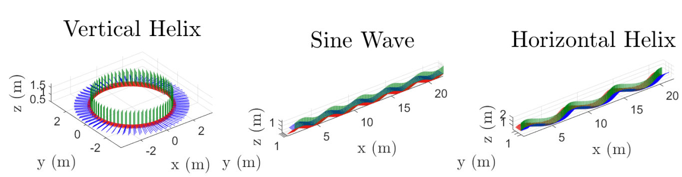

Fig. 4. Test paths used for the simulated experiment with the colored axes representing the desired pose of the end effector. The red, blue, and green axes represent the x, y, and z axes, respectively.

TABLE I
PARAMETERS USED FOR THE STATE PLANNER DURING THE FULL SIMULATED EXPERIMENT.

Parameter	Value
Δt (s)	0.1
T (s)	20.0
λ_e	25.0
λ_j	0.001
$\boldsymbol{\lambda}_v$	[1.0, 1.0, 0.25, 0.25, 0.1, 0.1, 0.1, 0.1]

the different test paths is shown in Figure 5, with both the position and orientation error approaching zero for both robot configurations after the mobile manipulator converges with the path. The average position and orientation root mean square

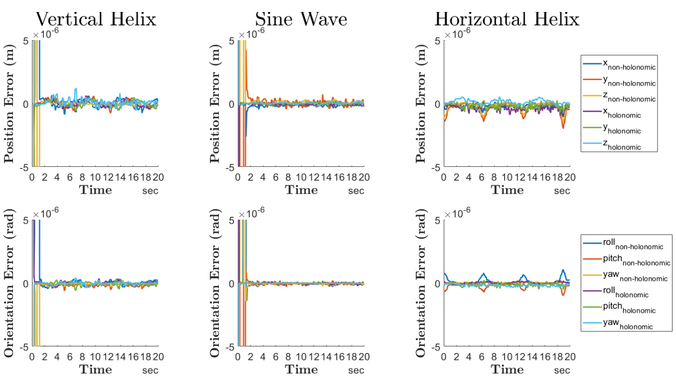

Fig. 5. The error of the end effector on each of the different paths for both the position and orientation when using a non-holonomic and holonomic mobile platform.

error (RMSE) for all of the paths across all of the time steps are shown to be 0.0243 meters and 0.0573 radians, respectively. A majority of the error comes from the initial two seconds as the robot converges with the desired path. The computed linear and angular velocities of the non-holonomic base are shown in Figure 6. The optimized joint states of

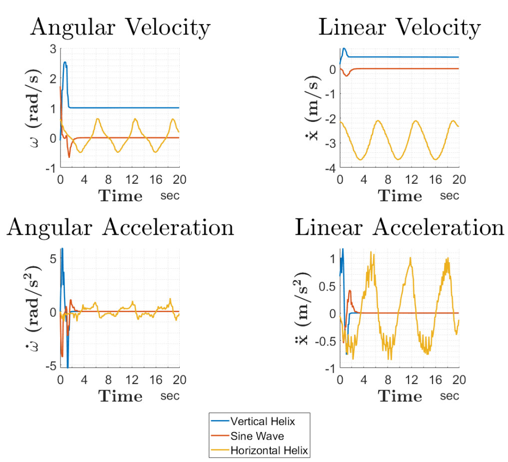

Fig. 6. The computed angular and linear velocity for the mobile manipulator with a non-holonomic base.

the manipulator on the non-holonomic base are shown in Figure 7. All of the computed inverse kinematic variables are

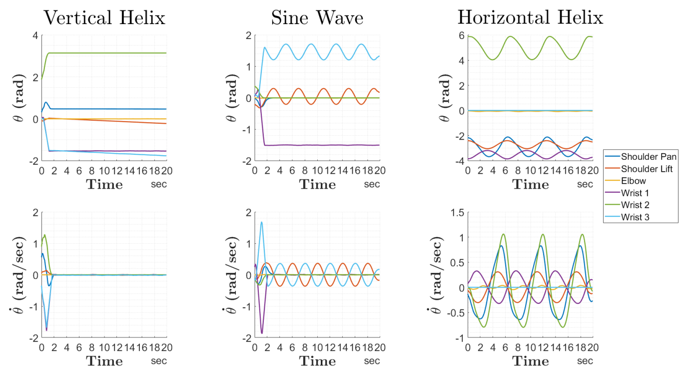

Fig. 7. The computed joint states for the robotic manipulator mounted on the non-holonomic base.

observed to be smooth and within reasonable bounds for a low level controller to execute. The mobile manipulator with a holonomic base performed similarly by accurately achieving the poses, as seen in Figure 5. The holonomic base exhibits smooth base and joint solutions, as seen in Figures 8 and 9, respectively. The main metrics required for performance

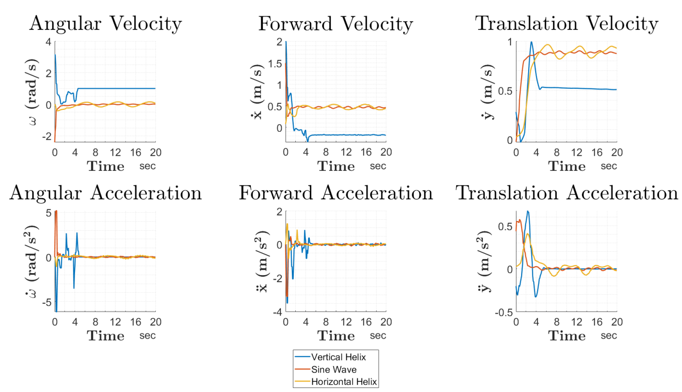

Fig. 8. The computed angular and linear velocities for the mobile manipulator with a holonomic base. The forward and translational velocities represent the velocities in the x and y direction with respect to the base frame, respectively.

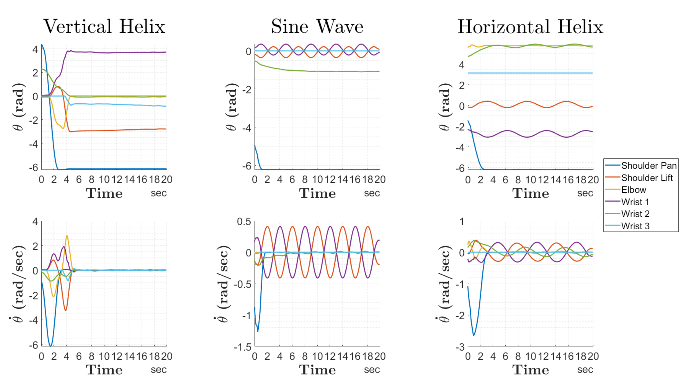

Fig. 9. The computed joint states for the robotic manipulator mounted on the holonomic base.

validation on each of the paths are shown in Table II for the

TABLE II

SUMMARIZED METRICS AND RESULTS FROM THE SIMULATED EXPERIMENT WHEN TESTING THE STATE PLANNER ON THE MOBILE MANIPULATOR WITH A NON-HOLONOMIC AND HOLONOMIC BASE.

Mobile Base Configuration →	Non-holonomic			Holonomic		
Trajectories ↓	Vertical Helix	Sine Wave	Horizontal Helix	Vertical Helix	Sine Wave	Horizontal Helix
Position RMSE (m)	0.0415	0.0299	0.0031	0.0472	0.0147	0.0094
Rotation RMSE (rad)	0.0214	0.0159	0.0016	0.0199	0.2841	0.0011
Max Forward Velocity (m/s)	0.8317	0.3004	3.6881	2.0000	1.4923	0.5514
Max Translation Velocity (m/s)	-	-	-	0.9933	0.9028	0.9650
Max Angular Velocity (rad/s)	2.5299	1.7195	0.6449	3.1416	2.4128	0.4185
Max Forward Acceleration (m/s^2)	1.1774	0.5491	1.1238	3.4888	3.1030	1.2341
Max Translation Acceleration (m/s^2)	-	-	-	0.6746	0.5749	0.4107
Max Ang Accel (rad/s^2)	5.8892	4.2186	1.2413	6.1535	5.1969	1.0681
Max Forward Jerk (m/s^3)	5.8309	1.5568	3.4703	24.6442	15.3703	11.4400
Max Translation Jerk (m/s^3)	-	-	-	1.1581	0.6296	0.3900
Max Angular Jerk (rad/s^3)	23.7865	19.4592	6.7439	36.1791	25.6212	3.4932
Max Joint Velocity (rad/s)	1.7848	1.8732	1.0609	6.1663	1.2688	2.6576
Max Joint Acceleration (rad/s^2)	1.9905	2.5079	1.1399	5.4608	1.4042	1.8988
Max Joint Jerk (rad/s^3)	10.0259	7.9320	2.6727	9.6929	2.7737	3.4553

non-holonomic and holonomic bases. The maximum values shown in the table, mostly come from the initial motion of the robot to achieve the first state and are otherwise significantly less. The metrics and figures show the inverse kinematic planner can generate smooth and feasible state curves for a mobile manipulator. Since the state and its derivatives do not violate the bounds of the robot and the jerk is low enough for control, it shows that the entire states are feasible to execute.

Using Lie theory to formulate the objective function allowed for the high performance of the proposed method. Due to the kinematic constraints in the product of exponentials, the steady state error tends towards zero and the motion is smooth in between points. Since the objective function also includes costs associated with sporadic motion, the state solutions are smooth enough between desired points to be executed. The low error and smooth states show the benefits of using an optimization based approach to determine the desired states for the mobile manipulator. The performance of the proposed method on different mobile manipulators shows the method is generalizable for different robotic bases and manipulators.

V. CONCLUSIONS

This paper presented a method for whole body motion planning of mobile manipulators using Lie theory and constrained optimization to generate the proper states from inverse kinematics. The input for the proposed method is a path of poses for the end effector to follow as efficiently as possible.

In this paper, it was found that the inverse kinematics planner can generate smooth and feasible state curves. The RMSE of the end effector pose compared to the desired pose is approaching zero upon reaching steady state. The base and joint velocities and accelerations are all within reasonable bounds and have no major discontinuities. Due to the evaluated performance, the proposed method is found to be a suitable state planner for mobile manipulators. The results clearly show the capabilities of the proposed approach to plan the motion for a mobile manipulator to follow and achieve the desired end effector trajectory.

Future work includes testing the state planning performance on real world robots with noise and disturbances. The addition of position based force control using estimated deformations is being researched and pursued. Evaluation for planning with higher degrees of freedom systems, including humanoids, is sought to present in the upcoming opportunities including conferences and journals.

REFERENCES

[1] P. Putz, “Space robotics in Europe: A survey,” *Robotics and Autonomous Systems*, vol. 23, no. 1, pp. 3–16, Mar 1998.

[2] A. Orsula, S. Bøgh, M. Olivares-Mendez, and C. Martinez, “Learning to Grasp on the Moon from 3D Octree Observations with Deep Reinforcement Learning,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 4112–4119.

[3] P. Štibinger, G. Broughton, F. Majer, Z. Rozsypálek, A. Wang, K. Jindal, A. Zhou, D. Thakur, G. Loianno, T. Krajník, and M. Saska, “Mobile Manipulator for Autonomous Localization, Grasping and Precise Placement of Construction Material in a Semi-Structured Environment,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2595–2602, 2021.

[4] I. Vatavuk, G. Vasiljević, and Z. Kovačić, “Task Space Model Predictive Control for Vineyard Spraying with a Mobile Manipulator,” *Agriculture*, vol. 12, no. 3, 2022.

[5] K. Dörfler, G. Dielemans, L. Lachmayer, T. Recker, A. Raatz, D. Lowke, and M. Gerke, “Additive Manufacturing using mobile robots: Opportunities and challenges for building construction,” *Cement and Concrete Research*, vol. 158, p. 106772, 2022.

[6] Z. Li, P. Moran, Q. Dong, R. J. Shaw, and K. Hauser, “Development of a tele-nursing mobile manipulator for remote care-giving in quarantine areas,” in *2017 IEEE*

International Conference on Robotics and Automation (ICRA), 2017, pp. 3581–3586.

[7] W. Smith, Y. Qin, T. Furukawa, and G. Dissanayake, “Autonomous Robotic Map Refinement for Targeted Resolution and Local Accuracy,” in *2022 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2022, pp. 130–137.

[8] W. Smith, Y. Qin, S. Singh, H. Burke, T. Furukawa, and G. Dissanayake, “A Multistage Framework for Autonomous Robotic Mapping with Targeted Metrics,” *Robotics*, vol. 12, no. 2, 2023.

[9] M. M. Mabkhot, A. M. Al-Ahmari, B. Salah, and H. Alkhalefah, “Requirements of the Smart Factory System: A Survey and Perspective,” *Machines*, vol. 6, no. 2, p. 23, Jun 2018, number: 2 Publisher: Multidisciplinary Digital Publishing Institute.

[10] M. Javaid, A. Haleem, R. P. Singh, and R. Suman, “Substantial capabilities of robotics in enhancing industry 4.0 implementation,” *Cognitive Robotics*, vol. 1, pp. 58–75, Jan 2021.

[11] M. Pirani, A. Bonci, and S. Longhi, “A scalable production efficiency tool for the robotic cloud in the fractal factory,” in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016, pp. 6847–6852.

[12] D. Kalashnikov, A. Irpan, P. P. Sampedro, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, and S. Levine, “Deep reinforcement learning for robotic manipulation,” US Patent US20 210 237 266A1, Aug, 2021.

[13] C. V. Braun, J. Ortiz-Haro, M. Toussaint, and O. S. Oguz, “RHH-LGP: Receding Horizon And Heuristics-Based Logic-Geometric Programming For Task And Motion Planning,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 13 761–13 768.

[14] I. W. Muzan, T. Faisal, H. M. A. A. Al-Assadi, and M. Iwan, “Implementation of Industrial Robot for Painting Applications,” *Procedia Engineering*, vol. 41, pp. 1329–1335, Jan 2012.

[15] M. Schappler, S. Tappe, and T. Ortmaier, “Resolution of Functional Redundancy for 3T2R Robot Tasks using Two Sets of Reciprocal Euler Angles,” in *Advances in Mechanism and Machine Science*, T. Uhl, Ed. Cham: Springer International Publishing, 2019, pp. 1701–1710.

[16] G. Trigatti, P. Boscariol, L. Scalera, D. Pillan, and A. Gasparetto, “A new path-constrained trajectory planning strategy for spray painting robots - rev.1,” *The International Journal of Advanced Manufacturing Technology*, vol. 98, no. 9, pp. 2287–2296, Oct 2018.

[17] F. Marić, M. Giamou, S. Khoubyarian, I. Petrović, and J. Kelly, “Inverse Kinematics for Serial Kinematic Chains via Sum of Squares Optimization,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 7101–7107.

[18] D. Rakita, B. Mutlu, and M. Gleicher, “RelaxedIK: Real-time Synthesis of Accurate and Feasible Robot Arm Motion,” in *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018.

[19] S. LaValle and J. Kuffner, “Randomized kinodynamic planning,” in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 1, May 1999, pp. 473–479 vol.1, iSSN: 1050-4729.

[20] D. Berenson, J. Kuffner, and H. Choset, “An optimization approach to planning for mobile manipulation,” in *2008 IEEE International Conference on Robotics and Automation*, May 2008, pp. 1187–1192, iSSN: 1050-4729.

[21] T. Pardi, V. Maddali, V. Ortenzi, R. Stolkin, and N. Marturi, “Path planning for mobile manipulator robots under non-holonomic and task constraints,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2020, pp. 6749–6756, iSSN: 2153-0866.

[22] S. Moe, G. Antonelli, A. R. Teel, K. Y. Pettersen, and J. Schrimpf, “Set-based tasks within the singularity-robust multiple task-priority inverse kinematics framework: General formulation, stability analysis, and experimental results,” *Front. Robot. AI*, vol. 3, Apr 2016.

[23] D. Coleman, I. Sucan, S. Chitta, and N. Correll, “Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study, shorttitle = Reducing the Barrier to Entry of Complex Robotic Software,” Apr 2014, arXiv:1404.3785 [cs].

[24] G. Colucci, A. Botta, L. Tagliavini, P. Cavallone, L. Baglieri, and G. Quaglia, “Kinematic Modeling and Motion Planning of the Mobile Manipulator Agri.Q for Precision Agriculture,” *Machines*, vol. 10, no. 5, 2022.

[25] F. Paus, P. Kaiser, N. Vahrenkamp, and T. Asfour, “A combined approach for robot placement and coverage path planning for mobile manipulation,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep 2017, pp. 6285–6292, iSSN: 2153-0866.

[26] R. K. Megalingam, V. Prithvi Darla, and C. S. Kumar Nimmala, “Autonomous Wall Painting Robot,” in *2020 International Conference for Emerging Technology (INCET)*, Jun 2020, pp. 1–6.

[27] A. Muneer and Z. Dairabayev, “Design and implementation of automatic painting mobile robot,” *IAES International Journal of Robotics and Automation (IJRA)*, vol. 10, no. 1, pp. 68–74, Mar 2021, number: 1.

[28] Q. Yu, G. Wang, X. Hua, S. Zhang, L. Song, J. Zhang, and K. Chen, “Base position optimization for mobile painting robot manipulators with multiple constraints,” *Robotics and Computer-Integrated Manufacturing*, vol. 54, pp. 56–64, Dec 2018.

[29] A. Sarker, A. Sinha, and N. Chakraborty, “On Screw Linear Interpolation for Point-to-Point Path Planning,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 9480–9487.

[30] S. Ren, Y. Xie, X. Yang, J. Xu, G. Wang, and K. Chen, “A Method for Optimizing the Base Position of Mobile

Painting Manipulators," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, pp. 370–375, Jan 2017, conference Name: IEEE Transactions on Automation Science and Engineering.

[31] P. Trutman, M. S. E. Din, D. Henrion, and T. Pajdla, "Globally Optimal Solution to Inverse Kinematics of 7DOF Serial Manipulator," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6012–6019, 2022.

[32] H. Ardiny, S. Witwicki, and F. Mondada, "Construction automation with autonomous mobile robots: A review," in *2015 3rd RSI International Conference on Robotics and Mechatronics (ICROM)*, 2015, pp. 418–424.

[33] K. G. Fue, W. M. Porter, E. M. Barnes, and G. C. Rains, "An Extensive Review of Mobile Agricultural Robotics for Field Operations: Focus on Cotton Harvesting," *AgriEngineering*, vol. 2, no. 1, pp. 150–174, 2020. [Online]. Available: https://www.mdpi.com/2624-7402/2/1/10

[34] S. Kucuk and Z. Bingul, "Robot Kinematics: Forward and Inverse Kinematics," in *Industrial Robotics: Theory, Modelling and Control*, S. Cubero, Ed. Pro Literatur Verlag, Germany / ARS, Austria, Dec 2006.

[35] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. Wiley, Nov 2005, google-Books-ID: cPhvxwEACAAJ.

[36] R. Hartenberg and J. Danavit, *Kinematic synthesis of linkages*. New York: McGraw-Hill, 1964.

[37] R. S. Ball, "The theory of screws: A study in the dynamics of a rigid body," *Mathematische Annalen*, vol. 9, no. 4, pp. 541–553, Dec 1876.

[38] J. Solà, J. Deray, and D. Atchuthan, "A micro Lie theory for state estimation in robotics," 2018.

[39] E. Gilbert, D. Johnson, and S. Keerthi, "A fast procedure for computing the distance between complex objects in three-dimensional space," *IEEE Journal on Robotics and Automation*, vol. 4, no. 2, pp. 193–203, 1988.

[40] L. Qiao, X. Luo, Q. Luo, M. Li, and J. Jiang, "Optimizing Kinematic Modeling and Self-Collision Detection of a Mobile Manipulator Robot by Considering the Actual Physical Structure," *Applied Sciences*, vol. 11, no. 22, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/22/10591

[41] T. M. Inc., "Robotics system toolbox version: 4.1 (r2022b)," Natick, Massachusetts, United States, 2022. [Online]. Available: https://www.mathworks.com