Pick-and-Place Task using Wheeled Mobile Manipulator - A Control Design Perspective

Conference Paper · September 2020

DOI: 10.1109/ICCIT-144147971.2020.9213717

3 authors:

Muhammad Affan
NED University of Engineering and Technology, Karachi
15 PUBLICATIONS 46 CITATIONS

Syed Umaid Ahmed
National University of Computer and Emerging Sciences
17 PUBLICATIONS 44 CITATIONS

Riaz Uddin
NED University of Engineering and Technology, Karachi
50 PUBLICATIONS 279 CITATIONS

2020 International Conference on Computing and Information Technology, University of Tabuk, Kingdom of Saudi Arabia.
Volume: 02, Issue: ICCIT- 1441, Page No.: 34 – 39, 9th & 10th Sep. 2020

Pick-and-Place Task using Wheeled Mobile Manipulator - A Control Design Perspective

Muhammad Affan
Haptics, Human Robotics and Condition Monitoring Lab (affiliated with NCRA)
NED Unversity of Engineering and Technology
Karachi, Pakistan
engineer.affan@gmail.com

Syed Umaid Ahmed
Department of Mechanical Engineering
NED Unversity of Engineering and Technology
Karachi, Pakistan
syedumaidahmed96@gmail.com

Riaz Uddin
Haptics, Human Robotics and Condition Monitoring Lab (affiliated with NCRA)
NED Unversity of Engineering and Technology
Karachi, Pakistan
riazuddin@neduet.edu.pk

***Abstract*— This paper is aimed towards students and roboticists with an objective to provide the concise theoretical and applied knowledge necessary for the control design of mobile manipulator. For this purpose, topics such as kinematics, motion planning and control theory are explored. Moreover, this knowledge is integrated from the perspective of Mecanum wheeled 5-R mobile manipulator for the pick-and-place task of cube. Finally, the paper shows the successful application of the accumulated knowledge in simulated V-REP environment.**

Keywords—robot, mobile manipulator, pick-and-place, kinematics, feedback control

I. Introduction

In today's world, the field of robotics is making its way in every aspect of human life. New applications are being extensively explored to even further assist industrial, service, healthcare, education, space and agricultural sectors [1]–[3]. Yet, all these 'intelligent' robotic applications rely fundamentally on locomotive or manipulative abilities or the combination of both in the form of mobile manipulation. Specifically, human-like tasks are most effectively carried out by mobile manipulators because such mobile platform have the extended workspace and better provision of new navigation strategies. Moreover, added dexterity of robotic manipulator placed over mobile platform provides several operational functionalities. Although, many mobile manipulation categories have been proposed over the years but majorly they are categorized based on mode or type of locomotion [4], [5] such as legged walking robots, wheeled mobile platform, sub-sea robots, free-flying space robotic systems, etc. However, accurate modelling is fundamental among all mentioned robotic control tasks. Although, wheeled mobile manipulation is considered simplest with respect to modeling among its counter-part approaches but, rolling without slipping (r.w.s) consideration of the robotic wheels incorporates non-holonomic constraints. The study and modelling of the non-holonomic constraints are carried out in [6], [7]. Similarly, underactuated systems and non-linear controllability for wheeled manipulators could be referred from [8], [9]. For detailed study on mechanical aspects of robotic design, topics on 'robotic mobility' from [9] could be referred. Based on the mechanical analysis, controller of the robotic system is engineered with due consideration of stability, error-tolerance and construability [9]. Well-established yet evolving areas of linear (e.g. [10]) and non-linear (e.g. [8], [11]) control research forms the basis of modern control of robotics. Originating from 1970's as torque controller, the controllers' research for research is being shifted towards the non-linear, robust, adaptive and intelligent control methodologies. The book by Lynch [9] and Spong [12] could be referred for developing fundamental knowledge of control engineering practices in the field of robotics.

Motion planning is also highly relevant and interlink study that ensures the most optimized movement of mobile manipulator between initial and final state with respect to constraints, if any. The foundational earliest development in this arena was A$*$ search [13] which in turn was the improvement of Dijkstra's algorithm [14]. Since then, there has been tremendous research in this direction, leading to the two main division of sample base motion planning namely, probabilistic roadmaps (PRMs) [15] and rapidly exploring random trees (RRTs) [16], [17]. The subsequent research in this direction is summarized in [18], [19].

Research Institutes all over the world, along with extensive research contribution in different areas of robotics especially in mobile manipulator, have also shown great interest in developing their own experimental mobile manipulators based on commercially available robotic bases and arms. These are: Cody of Georgia Institute of Technology [20], POLAR of Cornell University [21], UMass Mobile Manipulator [22], Willow Garage's PR2, MR ROAM of Southwest Research Institute (SwRI) and the Fraunhofer IPA's Care-O-bot 4. This development is crucial as it forms the basis of new research by providing experimental platform to validate new research methods. Moreover, it provides an opportunity to researchers to commercialize their own product/services for professional, domestic and industrial robotic applications. Currently, the mobile manipulators are most widely utilized in industries for different kinds of collaborative and routine task. However, such industrial tasks require higher precision and efficiency in terms of speed, maneuverability, power consumption, robustness etc. [23].

From above discussion, it evident that there is a persistent need of improvements in mobile manipulator in all the mentioned areas, especially, the control design [24]. Therefore, this paper explores an important wheeled manipulation task (pick-and-place of object) in an indoor simulated environment from the control engineering perspective. To be precise, this paper provides the

978-1-7281-2680-7/20/$31.00 ©2020 IEEE

fundamental understanding of robot's kinematics, dynamics, motion planning and control design procedure, and their unified usage on KUKA's youBot for the pick-and-place task in V-REP environment. The authors aim to extend this study for the indigenous development of wheeled mobile manipulator for an automated pick-and-place task in warehouse.

This paper is distributed into four sections. The section I presents the introduction followed by section II which presents the methodology. Then section III presents the discussion and results while section IV concludes the paper.

II. THEORETICAL BACKGROUND

A. Dynamic simulations

Dynamic simulations fundamentally utilizes computational tools and programs for the purpose of modeling and analyzing the time-varying characteristics of systems. The computational tools/programs having such capability are called dynamic simulators. These systematic characteristics are typically depicted by ordinary differential equations or partial differential equations. However, inclusion of real-world constraints in mathematical model, turn equation non-linear thereby, requiring discrete numerical solution to the given equations. This is done through traversing all time intervals one-by-one in fixed or adaptive step mode based on error tolerance while accumulating area under the differential curves. Dynamic simulations have been extensively used since decades and even now, their applications are expanding namely but not limited to process control, mechatronic/robotic systems, computational fluid dynamics, power systems analysis, weather forecasting, climatic changes, flight simulations etc. In case of robotic applications, physics/game engines are employed in for simulating robotic dynamic interactions between bodies including force/torque responses (dynamic collision/contact) and articulated motion involving joints. Robotic simulator can be effectively used to design and verify algorithms for robotic platforms without needing the actual physical systems. Later, the simulated work could be applied to physical system with no or little modification. There are various dynamic simulators available for robotic systems with different features e.g. V-REP, MuJoCo, Gazebo, Webots and many others. Nevertheless, the goal of every robotic simulator is to accurately realize collision, restitution, contacts, friction etc. by integrating and advancing continuously to next timestep in simulation, all within the limited computational resources.

In our case, robotic simulation has been carried out in V-REP simulator with ODE as physics engine. V-REP simulator provides flexible integrability with other environments/plugins, rich set of features and the possibility to choose physics engine from Bullet, ODE, Vortex, and Newton based on their application.

B. Kinematics of the wheeled mobile manipulator

For developing better understanding of the kinematics involved in wheeled mobile manipulation, the task of manipulation and locomotion are separately discussed followed by the integration of presented knowledge.

1) Robotic Manipulation: Object's manipulation such as pick, place, move, damage or at least influencing an object through end-effector (hand) of robotic arm is referred to as robotic manipulation. In most cases, an end-effector is a gripper with two fingers (open/close state) rigidly attached to the last link of robotic arm. A reference frame could be assigned to each robotic link however, two reference frames namely, fixed reference frame {s} attached to the manipulator's base and the base or end-effector frame {'b' or 'e'] attached to the last link of robotic arm are of utmost importance in kinematic calculations. These frames are utilized in obtaining position and orientation of {b} in {s}frame from joint angles θ (forward kinematics) or vice versa (inverse kinematics). Denavit–Hartenberg (D-H) convention, introduced in [25], based forward kinematic calculation is relatively common in technical literature while the other being the Product of Exponentials (PoE), originally introduced by Brockett in [26]. In our work, we have adopted PoE formula for forward kinematic (in spatial form) and can be given as (1).

$$T(\theta) = e^{[S_1]\theta_1} \cdots e^{[S_n]\theta_n} M;\ i = 1,2,3 \cdots n \tag{1}$$

Where, $[S_i]$ represents the counter clockwise motion of screw axis of joint 'i' (se{3} matrix form in fixed-frame {s} coordinates), θ_i refers to the associated variables with joint 'i', M indicates the end-effector's frame {b} position and orientation at home(zero) configuration (SE{3}) while T is the Transformation matrix respectively. This could also be represented in body frame as (2).

$$T(\theta) = M e^{[B_1]\theta_1} \cdots e^{[B_n]\theta_n} \tag{2}$$

Where, B_i is the screw axis associated with joint 'i' in body reference-frame {b} at home (zero) configuration of robot as shown in (3).

$$B_i = [Adj_{M^{-1}}] S_i;\ i = 1,2,3 \cdots n \tag{3}$$

As evident, the taken PoE form serves many advantages over conventional D-H parameters. It eliminates the need of defining all individual link frames, revolute and prismatic joint are uniformly treated, and screw representation of joint axes have better geometrical interpretability [9]. Moreover, PoE form enables simplified Jacobian derivation as each joint in screw representation acts as columns. Jacobian matrix maps joint velocities to spatial or body twist as shown in (4) and (5) respectively.

$$\mathcal{V}_s = J_s(\theta)\dot{\theta} \tag{4}$$

$$\mathcal{V}_b = J_b(\theta)\dot{\theta} \tag{5}$$

Where, J_s and J_b are spatial and body twist whose i^{th} column could be computed as shown in (6) and (7) respectively.

$$J_{si}(\theta) = Ad_{e^{[S_1]\theta_1} \cdots e^{[S_{i-1}]\theta_{i-1}}} S_i(\theta_i) \tag{6}$$

$$J_{bi}(\theta) = Ad_{e^{-[B_n]\theta_n} \cdots e^{-[B_{i+1}]\theta_{i+1}}} B_i(\theta_i) \tag{7}$$

Jacobian matrix (body or spatial-frame coordinates) along with wrenches imposed on end-effecter (Ƒ) could be utilized to calculate joint torques/forces (τ) as shown in (8).

$$\tau = J_b^T(\theta)\mathcal{F}_b = J_s^T(\theta)\mathcal{F}_s \tag{8}$$

The Jacobian matrices are also utilized in computation of inverse kinematics through Jacobian inverse method as shown in (9).

$$\theta^{i+1} = \theta^{i} + J^{\dagger}(\theta^{i})\nu \tag{9}$$

Where $J^{\dagger}$ is the pseudoinverse of the Jacobian J(θ), ν is twist and θ is the solution of inverse kinematics for desired end-effector configuration X ∈ SE(3) of any spatial open chain robot with forward kinematics T(θ) satisfying (10).

$$X = T(\theta) \tag{10}$$

2) Wheeled Locomotion: Wheeled Locomotion refers to the mechanism of traversing from one place to another in context of mobile robots. Let the chassis configuration of wheeled mobile robot in plane can be represented as (11) and associated planar twist as (12).

$$q = (\emptyset, x, y) \tag{11}$$

$$\nu_b = \begin{vmatrix} \omega_{bz} \\ \nu_{bx} \\ \nu_{by} \end{vmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\emptyset & \sin\emptyset \\ 0 & -\sin\emptyset & \cos\emptyset \end{bmatrix} \begin{bmatrix} \dot{\emptyset} \\ \dot{x} \\ \dot{y} \end{bmatrix} \tag{12}$$

Chassis of mobile robots are subjected to single nonintegrable Pfaffian velocity constraints termed as nonholonomic constraints (13) [6]. However, no such constraints are imposed on omnidirectional robots' (Mecanum or omni wheels) i.e. our taken approach.

$$\begin{aligned} A(q)\dot{q} &= [0 \quad \sin(\emptyset) \quad -\cos(\emptyset)]\dot{q} \\ &= \dot{x}\sin(\emptyset) - \dot{y}\cos(\emptyset) = 0 \end{aligned} \tag{13}$$

The wheels' velocity *u* of omnidirectional mobile robot (with 3 or higher number of wheels) could be obtained by either $\dot{q}$ through *H(Φ)* or ν_b through *H(0)* matrix of order = (number of wheels, 3) or even vice versa as shown in (14) and (15) respectively.

$$u = H(\emptyset)\dot{q} = H(0)\nu_b \tag{14}$$

$$\nu_b = H^{\dagger}(0)u \tag{15}$$

Equation (15) can be used to provide estimate of new chassis configuration *(q)* based on the change in chassis configuration with respect to initial/previous configuration point through odometry.

3) Wheeled mobile manipulation: A wheeled mobile manipulator with '*w*' _number of wheels and robotic arm of 'j' joint coordinated motion can be accomplished through the following formulae:

$$\nu_e = J_e(\theta)\begin{bmatrix} u \\ \dot{\theta} \end{bmatrix} = [J_{base}(\theta) \quad J_{arm}(\theta)]\begin{bmatrix} u \\ \dot{\theta} \end{bmatrix} \tag{16}$$

Where $J_{arm}(\theta)$ is the manipulator Jacobean matrix as previously described whereas, $J_{base}(\theta)$ transforms wheel velocities to the velocity of end-effector as shown in (17).

$$J_{base}(\theta) = \left[Ad_{T_{0e}^{-1}(\theta)T_{b0}^{-1}(\theta)}\right]F_6 \tag{17}$$

Where {e}, {b} and {0} are end-effector, mobile robot's base and manipulator origin reference frames while F_6 is the pseudoinverse of *H(0)* with rows of zeros stack above and below the actual matrix [9].

C. Trajectory Generation

A trajectory *θ(t)* is pre-computed or online sequentially computed joints/end-effector positions and velocities that robot follows while performing a specified task. A good trajectory function follows smoothly over time while considering all joint limits, if specified [9]. Path *θ(s(t))* or *θ(s)* is a more frequently used term in context of robotic trajectory planning that provides geometric interpretation of robot's joint configurations *Φ* with the help of time scaling factor '*s*'. These two terminologies can be mathematically written as (18) and (19).

$$\theta(t): [0, T] \rightarrow \phi \tag{18}$$

$$\theta(s): [0,1] \rightarrow \phi \tag{19}$$

Where, '*s*' assigns robot's configuration values *Φ* to all instants of motion/trajectory time period as shown in (20).

$$t \in [0, T], s: [0, T] \rightarrow [0,1] \tag{20}$$

The trajectory and associated scaling need to be twice differentiable to obtain robot's motion dynamics (joint velocities and accelerations) as shown in (21) and (22).

$$\dot{\theta} = \frac{d\theta}{ds}\dot{s} \tag{21}$$

$$\ddot{\theta} = \frac{d\theta}{ds}\ddot{s} + \frac{d^2\theta}{ds^2}\dot{s}^2 \tag{22}$$

Typically, robotic motion can be carried out in point-to-point straight line (joint space and task space), polynomial via reference points or via optimal-time path with given consideration to actuator limits [12]. The straight point-to-point/rest-to-rest approach is considered for this control design of manipulation task. It is important to note that the trajectory of robotic motion essentially depends upon the representation/nature of straight-line path and type of time scaling technique used for the given motion. Typically, joint-space and cartesian-space representation are used for straight-line that could be mathematically expressed in (23) and (24) respectively.

$$\theta(s) = \theta_{start} + s(\theta_{end} - \theta_{start}), s \in [0,1] \tag{23}$$

$$X(s) = X_{start} + s(X_{end} - X_{start}), s \in [0,1] \tag{24}$$

Where, θ_{start}, θ_{end}, X_{start} and X_{end} are start and end configurations in joint and task space respectively. The (2) is defined in minimum set of coordinates however, if one wishes to use SE(3) representation then screw motion could be taken as shown in (25) considering start and end configurations are in s-frame considering $s \in [0,1]$.

$$X(s) = X_{start,s}exp\left(\log\left(X_{s,start}^{-1}X_{s,end}\right)s\right) \tag{25}$$

The estimated straight-line path in both joint or cartesian space is then mapped over the time horizon accordingly while respecting pre-specified velocity and acceleration constraints of given robot. The time-mapping could be polynomial, trapezoidal or S-curve [9]. This control design has used time scaling of 5th order polynomial form as shown in (26).

$$s(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad (26)$$

Subjected to following constraints shown in (27) and (28).

$$s(0) = \dot{s}(0) = \ddot{s}(0) = \dot{s}(T) = \ddot{s}(T) = 0 \quad (27)$$

$$s(T) = 1 \quad (28)$$

Equation (1) can be evaluated against above constraints to find unknown coefficients.

D. Control Engineering

The control design enables robotic motion or force (or even both) control as per the given task or environment. The typical objectives are motion/force control, hybrid force-motion control and impedance control through feedforward/feedback with respect to the mechanical constraints imposed. A simplified block diagram of robotic controller is shown in Fig. 1.

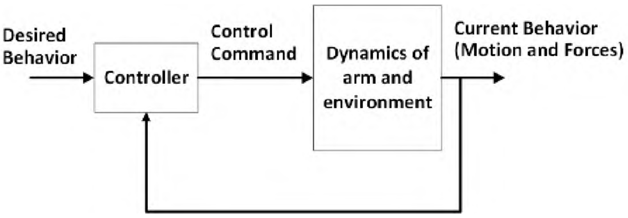

Fig. 1. Simplified block diagram of robot's control loop

Some controller works by minimizing the difference between the reference value and the actual value of the controlled system. For instance, in case of single-joint robot, the reference and actual values could be desired joint-position [$\theta_d(t)$] and the actual joint-position [$\theta(t)$] respectively, written as (29).

$$\theta_e(t) = \theta_d(t) - \theta(t) \quad (29)$$

If above equation is formulated as differential equation (error evolution over time), it is called error dynamics of the controller. The performance of this controller could be characterized by analyzing the error response (when provided with initial non-zero reference) in the given system such as overshoot, 2% or 5% settling time and steady-state error.

Similarly, the above idea could be extended to multi-joint system, and for the control of other input variables such as velocity, force, impedance etc. [9]. Contrary to above difference based closed-loop feedback controller, the open-loop feedforward controller is also used in control design of robots. The combination of feedforward and feedback controller also combines their advantages. Therefore, the task-space based feedforward plus PI Feedback motion controller has been adopted in the control design of discussed wheeled manipulator whose twist computation is govern by the (30).

$$v_b(t) = [Ad_{X^{-1}X_d}]v_d(t) + K_p X_e(t) + K_i \int_0^t X_e(t)dt \quad (30)$$

Whereas, (31)-(33) shows the controller gain and feedforward term.

$$K_p = k_p I \quad (31)$$

$$K_i = k_i I \quad (32)$$

$$[X_e] = \log(X^{-1}X_d) \quad (33)$$

Similarly, a task-space based feedforward plus PI Feedback force control is used as shown in (34).

$$\tau = \tilde{g}(\theta) + J^T(\theta)\big(F_d + K_{fp}F_e + K_{fi}\int F_e(t)dt - K_{damp}v\big) \quad (34)$$

Whereas, $F_e = F_d - F_{tip}$ while F_{tip} is wrench applied by end-effector to environment, $\tilde{g}(\theta)$ gravitational torques, F_d is desired wrench, v is actual velocity, K_{damp} is damping positive-definite matrix, K_{fp} and K_{fi} positive-definite proportional and integral gain matrices.

III. Methodology

Fig. 2 summarizes the above discussion by illustrating the flow of control in the given wheeled manipulation. Initially, the 12-configuration variables of chassis, arm and wheels (for our case, 3-axes coordinates, 5-joint and 4-wheel angles), 9-control variables of wheel and joint angular rate, sampling time, constraints and other configuration matrices for kinematic computations are initialized. Afterwards, the reference trajectory is pre-computed for the end-effector (from start to end of the given pick-and-place task of cube). The reference trajectory is computed based on the controller/servo's operational frequency, cube's initial and desired configuration. This reference trajectory is provided to the selected PI plus feedforward controller of the wheeled mobile manipulator. The feedback controller evaluates the current coordinated state with the help of (29). The gripper's twist i.e. the output of (30), expressed in end-effector frame {e} is used to compute the wheel and joint control velocities through inverse Jacobean matrix as shown in (17). These commanded angular rates are used to compute next state of the system with which reference trajectory is corrected (generated again), and this loop continues until the robot reached the goal state.

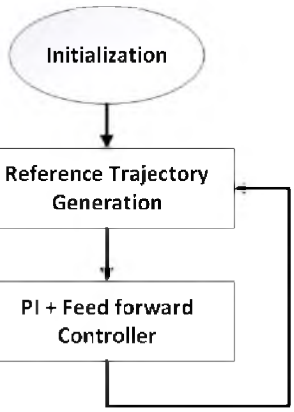

Fig. 2. The control system flowchart of designed wheeled manipulator

Fig. 3 shows the snapshots of simulated KUKA youBot in V-REP environment successfully performing the pick-and-place task of cube.

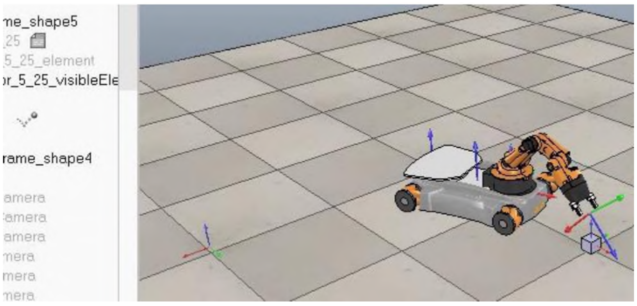

Fig. 3-a. Gripper move from home configuration to the top of cube

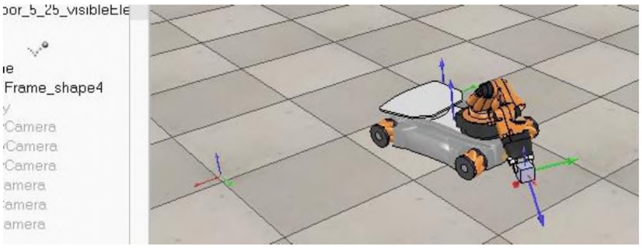

Fig. 3-b. Gripper manipulating the cube from its initial to final position

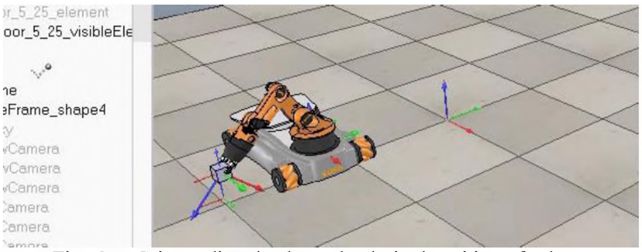

Fig. 3-c. Gripper directly above the desired position of cube.

Fig. 3-d. Gripper placing the cube at the desired location

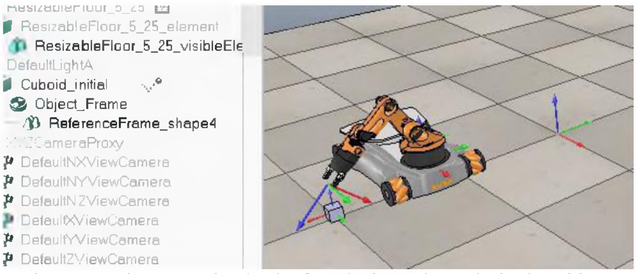

Fig. 3-e. Gripper moving back after placing cube at desired position

IV. Conclusion

In this paper, we have explored the object manipulation task by using the over-actuated Mecanum wheeled mobile platform (KUKA youBot) in the V-REP simulated environment. Specifically, this work has developed the fundamental understanding of kinematic models and applied control engineering practices towards the given pick-and-place task. The integration of knowledge towards the given application is successfully validated as evident from simulation results. In future, this work will be extended to the actual implementation of the wheeled mobile manipulator for local warehouses.

References

[1] F. Rubio, F. Valero, and C. Llopis-Albert, “A review of mobile robots: Concepts, methods, theoretical framework, and applications,” *Int. J. Adv. Robot. Syst.*, vol. 16, no. 2, pp. 1–22, 2019.

[2] M. Affan, S. U. Ahmed, A. Isfand Yar Manek, and R. Uddin, “Design and Implementation of the Washout Filter for the Stewart-Gough Motion Platform,” in *Proceedings of 2019 International Conference on Computational Intelligence and Knowledge Economy, ICCIKE 2019*, 2019, pp. 415–419.

[3] L. Royakkers and R. van Est, “A Literature Review on New Robotics: Automation from Love to War,” *Int. J. Soc. Robot.*, vol. 7, no. 5, pp. 549–570, 2015.

[4] T. Arai, “Robots with integrated locomotion and manipulation and their future,” in *IEEE International Conference on Intelligent Robots and Systems*, 1996, vol. 2, pp. 541–545.

[5] B. Bayle, J. Y. Fourquet, and M. Renaud, “From manipulation to wheeled mobile manipulation: Analogies and differences,” *IFAC Proc. Vol.*, vol. 36, no. 17, pp. 97–101, 2003.

[6] G. Campion, G. Bastin, and B. D’Andréa-Novel, “Structural properties and classification of kinematic and dynamic models of wheeled mobile robots,” *IEEE Trans. Robot. Autom.*, vol. 12, no. 1, pp. 47–62, 1996.

[7] Y. Chang, S. Ma, H. Wang, and D. Tan, “A kinematic modeling method for a wheeled mobile robot,” in *2009 IEEE International Conference on Mechatronics and Automation, ICMA 2009*, 2009, pp. 1100–1105.

[8] A. Isidori, *Nonlinear Control Systems*. Springer, 2006.

[9] Kevin M. Lynch and Frank C. Park, *Modern Robotics: Mechanics, Planning and Control*, no. May. 2017.

[10] G. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback control of dynamic systems*, 6th ed., vol. 55–2. Pearson, 1994.

[11] V. Jurdjevic, *Geometric Control Theory*. Cambridge: Cambridge University Press, 1996.

[12] M. W. Spong, S. Hutchinson, and M. Vidyasagar, “Robot modeling and control,” *IEEE Control Syst.*, vol. 26, no. 6, pp. 113–115, 2006.

[13] P. E. Hart, N. J. Nilsson, and B. Raphael, “Formal Basis for the Heuristic Determination of Minimum Cost Paths,” *Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, 1968.

[14] E. W. Drjksra, “A Note on Two Problems in Connexion with Graphs,” *Numer. Math.*, vol. 6, no. 2, pp. 269–271, 1959.

[15] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, “Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces,” *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, 1996.

[16] S. M. LaValle and J. J. Kuffner, “Randomized Kinodynamic Planning,” *Int. J. Rob. Res.*, vol. 20, no. 5, pp. 378–400, May 2001.

[17] S. M. Lavalle and J. Kuffner, “Rapidly-Exploring Random Trees: Progress and Prospects,” in *Algorithmic and computational robotics, new directions*, 2000, pp. 293--308.

[18] K. Lynch, A. Bloch, S. Drakunov, M. Reyhanoglu, and D. Zenkov, “Control of Nonholonomic and Underactuated Systems,” no. November 2015, pp. 42-1-42–36, 2010.

[19] J. Baillieul, *Encyclopedia of Systems and Control*. 2020.

[20] C. H. King, T. L. Chen, A. Jain, and C. C. Kemp, “Towards an assistive robot that autonomously performs bed baths for patient hygiene,” *IEEE/RSJ 2010 Int. Conf. Intell. Robot. Syst. IROS 2010 - Conf. Proc.*, no. 1, pp. 319–324, 2010.

[21] Y. Jiang, M. Lim, C. Zheng, and A. Saxena, “Learning to place new objects in a scene,” *Int. J. Rob. Res.*, vol. 31, no. 9, pp. 1021–1043, May 2012.

[22] D. Katz, E. Horrell, Y. Yang, and B. Burns, “The UMass Mobile Manipulator UMan: An Experimental Platform for Autonomous Mobile Manipulation,” in *Workshop on manipulation in human environments at robotics: science and systems*, Citeseer, 2006.

[23] M. Hvilshøj, S. Bøgh, O. S. Nielsen, and O. Madsen, “Autonomous industrial mobile manipulation (AIMM): Past, present and future,” *Ind. Robot An Int. J.*, vol. 39, no. 2, pp. 120–135, 2012.

[24] E. Garcia, M. A. Jimenez, P. G. De Santos, and M. Armada, “The evolution of robotics research,” *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 90–103, 2007.

[25] B. Bayle, J. Y. Fourquet, and M. Renaud, “Kinematic modelling of wheeled mobile manipulators,” *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 1, no. January 2014, pp. 69–74, 2003.

[26] R. W. Brockett and Division, “Robotic Manipulators and the Product of Exponentials Formula,” in *International Symposium on the Mathematical Theory of Networks and Systems*, 1983.