2023 11th RSI International Conference on Robotics and Mechatronics (ICRoM) | 979-8-3503-0810-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICRoM60803.2023.10412488

Modeling and Control of a Holonomic Mobile Robots Box Pushing System with Non-Slipping Constraints on Box Touching Points

Farid Kaviany
Department of Electrical Engineering
Amirkabir University of Technology
Tehran, Iran
farid.kaviany@aut.ac.ir

Alireza Ansari
Department of Electrical Engineering
Amirkabir University of Technology
Tehran, Iran
alireza.ansari@aut.ac.ir

Mahdi Ardestani
Department of Electrical Engineering
Amirkabir University of Technology
Tehran, Iran
mahdi.ardestani@aut.ac.ir

Mohammad A. Khosravi
Department of Electrical Engineering
Amirkabir University of Technology
Tehran, Iran
m.a.khosravi@aut.ac.ir

Hajar Atrianfar
Department of Electrical Engineering
Amirkabir University of Technology
Tehran, Iran
atrianfar@aut.ac.ir

***Abstract*—This document delves into cooperative target enclosing and tracking control for multiple holonomic mobile robots. Considering non-slipping constraints, a comprehensive model is developed for a multiple mobile robot box-pushing system. Constraint and actuation forces are determined through mathematical models, allowing effective control strategies to be implemented. The efficacy of this approach in moving the box toward its intended destination is showcased by simulation results.**

***Index Terms*—Box-pushing, Holonomic Mobile Robots, Multi-Robot Systems, Kinetic Modelling, Non-Slipping Condition**

I. Introduction

Multi-robot systems have been used in many areas, from space applications to entertainment use. The main reason for this attention is the ability of these systems to perform tasks that are very difficult or impossible for a single robot to accomplish [1]. For instance, in large-scale operations like shipping multiple parts of a massive structure, using a single robot would result in prolonged processing times and many other problems.

Deploying multi-agent robotic systems has proven highly efficient and versatile across various domains. Nevertheless, their design and implementation pose challenges, including modeling dynamic interactions, predicting behavior, and orchestrating movements. Effective control strategies must guarantee seamless cooperation, communication, and adaptation to dynamic environments. To this end, a comprehensive and interdisciplinary approach is necessary, involving robotics, AI, optimization, and communication protocols to optimize the full potential of multi-robot systems. Through innovative solutions, researchers and engineers aim to overcome these challenges and unlock the full potential of multi-agent robotic systems.

The field of distributed robotics originated in the late-1980's [2]. There has been a notable surge in research interest surrounding the modeling and control of multi-robot systems, driven by their extensive theoretical and practical utility. These systems find application in various domains, including collaborative object manipulation, such as multiple robots' coordinated pushing of objects. Kobayashi and Hosoe introduced the design of a decentralized capturing behavior by numerous mobile robots [3]. In this paper, the non-holonomic constraints have been omitted, and it is also assumed that the contact between a robot and the object is a point contact without friction. In 2014, Sayyaadi and Babaee introduced a methodology for transporting objects using a group of wheeled non-holonomic mobile manipulators. They also derived a complete dynamic model for a mobile manipulator comprising a three-wheeled mobile base and a three degree-of-freedom (DOF) manipulator [4]. Abbaspour et al. introduced an optimal formation and control approach for a team of wheeled robots aimed at manipulating a shared object [5]. This method encompasses the system's dynamics, including the kinematics and dynamics model of a constrained non-holonomic robot and the dynamics of the object.

In another attempt, Yufka and Ozkan addressed a motion-planning and control scheme for a cooperative transportation system comprising a single rigid thing and multiple autonomous non-holonomic mobile robots [6]. In 2016, Farivarnejad et al. presented a decentralized sliding mode control for autonomous collective transport by multi-robot systems [7]. They have created 3D simulations integrating accurate physical effects from the robots' wheeled actuation system and the manipulator arms' extra degrees of freedom. These simulations needed adjustments to the sliding mode controllers because Pheeno is a non-holonomic, differential-drive platform

979-8-3503-0810-5/23/$31.00 ©2023 IEEE

[8]. In 2018, Alipour and Abbaspour studied the effects of various passive and semi-active *Remote Center Compliance* (RCC) methods on formation control of a group of *Wheeled Mobile Robots* (WMRs) for object manipulation during space and terrestrial explorations [9]. They also derived the dynamics model of the constrained non-holonomic robots, objects, and RCC systems. Xinghua Lt al., in 2022, investigated cooperative target enclosing and tracking control with obstacle avoidance for multiple non-holonomic mobile robots [10].

In this paper, a detailed model for a mobile robot box-pushing system is being developed, which involves multiple robots collaborating to move a box to a specific location. Non-slipping restrictions and friction at the points of contact between the robots and the box are considered in the model. Non-slipping constraints are also incorporated to accurately represent robot-box interactions and prevent slipping or sliding. This allows effective control strategies to be developed for efficient and stable pushing toward the desired location.

The paper is organized as follows: Section II presents the modeling process of the proposed scheme and its validation. Section III discusses the control approach for this structure, followed by Section IV, which reports the simulation results. Finally, Section V presents the conclusions.

II. System Modeling

Valuable knowledge about the dynamics of a system can be gained by using mathematical models like the kinetic model. The Newton-Euler system can calculate Constraint and actuation forces, providing a complete understanding of the system. Efficient, safe, and stable strategies can be developed by predicting a system's reaction to external pressures through a kinetic model. Equations of motion are calculated using the principle of stationary action, and a kinetic model is established by considering the system's energy and limitations. If the system is designed with minimal variables, restrictions may not be necessary.

A kinetic model of the system will be presented, generated by two methods of computing the dynamic model: the Newton-Euler system and the Euler-Lagrange kinetic model. Once both models have been established, their accuracy will be validated to ensure the system's behavior is represented correctly.

A. Frames and Kinematics Analysis

In the box-pushing system depicted in Fig. 1, four mobile robots help the box move in different directions. It is easier to describe and analyze the system's behavior in Fig. 1 because all the movements and forces required for the system to function have been affected in the assigned frames. These designated frames help provide a thorough understanding of the system's dynamics, enabling a clear and succinct description of the motions and forces involved in the box-pushing procedure.

The system's circular agents have a radius of r_i, while the rectangular box has a width of w_1 and a height of w_2. The distance between agent i and the box is d_i, and the distance the agent moves when making contact with the center of the box's side is h_i. These measurements provide crucial details about the spatial dynamics between the agents and the box, enabling a comprehensive analysis of the box-pushing process. Non-slipping constraint rules must be followed when moving on the box side.

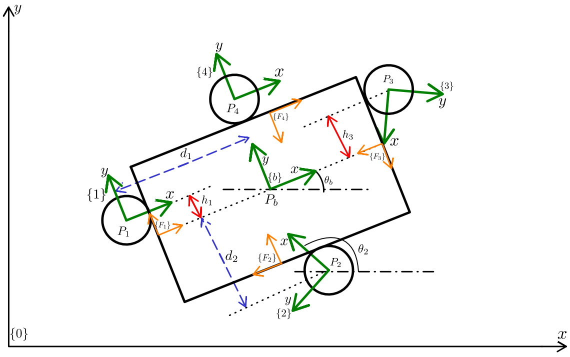


Fig. 1. System top view with assigned frames

The critical factors in the system being studied that clarify the movements of the agents are the x coordinate of the box (x_b), the y coordinate of the box (y_b), the angle of the box in the main frame (θ_b) and the angles of agents in the main frame (θ_i , $i = 1, ..., 4$). These variables include the specific positional and angular traits needed to explain the agents' motions and speeds. Equations derived from these variables can be used to accurately determine the position and velocity of the agents, as demonstrated in (1).

$$ {}^{0}\mathbf{P}_i = {}^{0}\mathbf{P}_b - {}^{0}\mathbf{R}_b {}^{b}\mathbf{R}_{F_i} \begin{bmatrix} d_i \\ h_i \end{bmatrix} \ , \ i = 1, 2, 3, 4 \tag{1} $$

$$ {}^{0}\mathbf{R}_b = \begin{bmatrix} \cos\theta_b & -\sin\theta_b \\ \sin\theta_b & \cos\theta_b \end{bmatrix} \tag{2} $$

$$ {}^{b}\mathbf{R}_{F_i} = \begin{bmatrix} \cos\theta_{F_i} & -\sin\theta_{F_i} \\ \sin\theta_{F_i} & \cos\theta_{F_i} \end{bmatrix} \ , \ \theta_{F_i} = \frac{(i-1)\pi}{2} \tag{3} $$

The relation (1) presents the kinematic model of the agent center under examination. The value of the variable d_i described in Table I remains constant but differs among agents, depicting the unique spatial correlation between each agent and box pair by indicating the unchanging distance between them. The relation (4) introduces variable h_i, denoting the distance between the agent and the center of the side of the box it touches. This measurement indicates the agent's movement relative to the reference point on the side of the box.

$$ h_i = r_i \left(\theta_i - \theta_b\right) \ , \ i = 1, 2, 3, 4 \tag{4} $$

After identifying the key variables determining the agents' location, the subsequent step involves calculating their speeds. This entails performing a time derivative on the position equation to determine the rates of change in position over

TABLE I
VALUE OF d_i FOR EACH AGENT

i	d_i
1	$\frac{w_1}{2} + r_1$
2	$\frac{w_2}{2} + r_2$
3	$\frac{w_1}{2} + r_3$
4	$\frac{w_2}{2} + r_4$

time. The resulting velocities, as expressed in (5-7), provide the necessary data to gauge the agents' speeds accurately.

$$ {}^0\dot{\mathbf{P}}_i = {}^0\dot{\mathbf{P}}_b - {}^0\dot{\mathbf{R}}_b {}^b\mathbf{R}_{F_i} \begin{bmatrix} d_i \\ h_i \end{bmatrix} - {}^0\mathbf{R}_b {}^b\mathbf{R}_{F_i} \begin{bmatrix} 0 \\ \dot{h}_i \end{bmatrix} \tag{5} $$

$$ {}^0\dot{\mathbf{R}}_b = \dot{\theta}_b \begin{bmatrix} \cos\left(\frac{\pi}{2} + \theta_b\right) & -\sin\left(\frac{\pi}{2} + \theta_b\right) \\ \sin\left(\frac{\pi}{2} + \theta_b\right) & \cos\left(\frac{\pi}{2} + \theta_b\right) \end{bmatrix} \tag{6} $$

$$ \dot{h}_i = r_i \left(\dot{\theta}_i - \dot{\theta}_b \right), i = 1, 2, 3, 4 \tag{7} $$

(5 - 7) can be rewrite as (8) which $\mathbf{J}(\mathbf{q})$ is Jacobian matrix and $\mathbf{q}$ is generalized coordinates:

$$ \begin{bmatrix} {}^0\dot{\mathbf{P}}_1 \\ \dot{\theta}_1 \\ {}^0\dot{\mathbf{P}}_2 \\ \dot{\theta}_2 \\ {}^0\dot{\mathbf{P}}_3 \\ \dot{\theta}_3 \\ {}^0\dot{\mathbf{P}}_4 \\ \dot{\theta}_4 \end{bmatrix} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \mathbf{q} = \begin{bmatrix} x_b \\ y_b \\ \theta_b \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} \tag{8} $$

B. Kinetics Analysis

There are two main methods for computing system dynamics in kinetic analysis. The Newton-Euler relations are used in the first strategy. Equations using this technique take both internal and supporting forces into account. These equations may be used to determine the effects of forces on each body in the system. This strategy, however, demands the assessment of internal and auxiliary forces. Moreover, it is possible to evaluate a minimum-set dynamic without relying on measuring internal and supporting forces by resolving these equations. In this system, the relationships are structured as (9) and (10):

$$ \text{Agent:} \begin{cases} m_i {}^0\ddot{\mathbf{P}}_i &= {}^0\mathbf{F}_{A_i} - {}^0\mathbf{R}_b {}^b\mathbf{R}_{F_i} \begin{bmatrix} F_{e_i} \\ F_{h_i} \end{bmatrix} \\ I_i \ddot{\theta}_i &= \tau_{A_i} - r_i F_{t_i} \end{cases} \tag{9} $$

$$ \text{Box:} \begin{cases} M_b {}^0\ddot{\mathbf{P}}_b &= \sum_{i=1}^{4} {}^0\mathbf{R}_b {}^b\mathbf{R}_{F_i} \begin{bmatrix} F_{e_i} \\ F_{h_i} \end{bmatrix} \\ I_b \ddot{\theta}_b &= -\sum_{i=1}^{4} (h_i F_{e_i} + (d_i - r_i) F_{t_i}) \end{cases} \tag{10} $$

The Euler-Lagrange technique, the second strategy, is based on the minimum action principle. The difference between a mechanical system's kinetic and potential energy is referred to as the action in that system. Creating a brief equation that controls the kinetic interactions inside the system is feasible using the Lagrange technique to minimize the action. In the box-pushing system, since there is no change in height or compression of a spring, there is no potential energy difference. Therefore, the computation only focuses on kinetic energy, which is expressed by (11):

TABLE II
SIMULATION AND VALIDATION PARAMETERS OF THE MODEL

Parameter Name	Parameter Symbol	Parameter Value
Box Mass	M_b	1 (kg)
Box Inertia	I_b	0.1 $(kg.m^2)$
Box Width	w_1	1 (m)
Box Length	w_2	1 (m)
Agents Mass	m_i	0.1 (kg)
Agents Inertia	I_i	0.01 $(kg.m^2)$
Agents radius	r_i	0.1 (m)

$$ \begin{aligned} T = &\frac{1}{2} M_b \left\| {}^0\dot{\mathbf{P}}_b \right\|^2 + \frac{1}{2} I_b \left| \dot{\theta}_b \right|^2 \\ &+ \sum_{i=1}^{4} \left(\frac{1}{2} m_i \left\| {}^0\dot{\mathbf{P}}_i \right\|^2 + \frac{1}{2} I_i \left| \dot{\theta}_i \right|^2 \right) \end{aligned} \tag{11} $$

System kinetics dynamic model can be evaluated from (12):

$$ \frac{d}{dt} \frac{\partial}{\partial \dot{\mathbf{q}}} T - \frac{\partial}{\partial \mathbf{q}} T = \mathbf{J}(\mathbf{q})^T \boldsymbol{\tau} \tag{12} $$

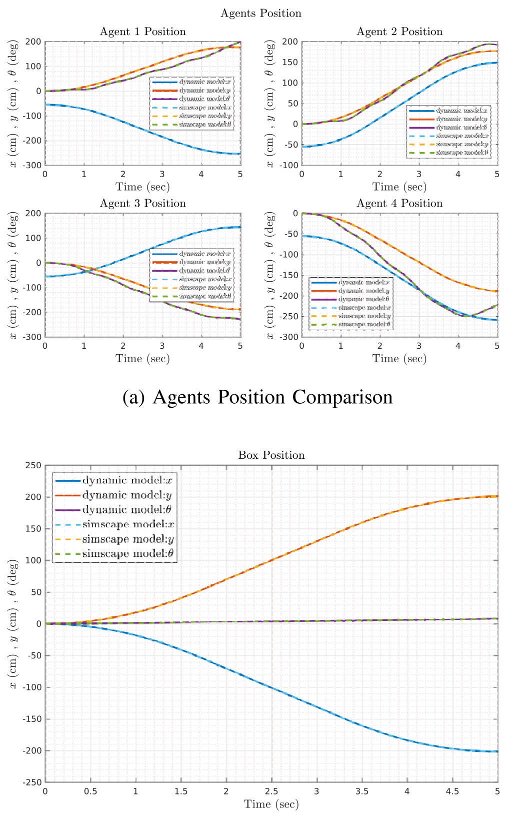
In (12), $\boldsymbol{\tau}$ refers to the force and torque of agents applied to the system. Based on [11]–[13], the dynamic model can be represented in general form in (13). The appendix contains more details of matrices used in (13).

$$ \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \mathbf{J}(\mathbf{q})^T \boldsymbol{\tau} \tag{13} $$

C. Model Validation

For precise modeling and simulation of systems, MATLAB/Simulink and Simscape Multibody toolbox are used to create a block diagram of the physical system. Revolute and prismatic joints accurately represent linear and rotational motion, respectively. These joints are crucial for illustrating the constraints on the agents and the box. Equation (4) establishes the kinematic connection between the revolute joint's orientation and the prismatic joint's location. A fundamental constraint governs the interaction between the agents' translational and rotational movements.

The Simscape model of the system and dynamic model in (13) are compared in Fig. 2 for a rectangular pulse input. The system's precision level is noteworthy, with an absolute error of order 1e-10. This feature enhances the system's reliability, making it a viable option for future use. The parameters for the model can be found in Table II.



(a) Agents Position Comparison

(b) Box Position Comparison

Fig. 2. A comparison between the results of Simscape and Dynamic Model

III. Controlling Box Pushing System

To ensure precise following of the desired route, the inverse dynamic approach is an efficient control strategy for the system. However, this strategy must be modified to comply with the limits on pushing forces exerted by agents on the box. These limitations prevent detachment or separation between agents and the box, ensuring all forces applied to the box are positive. As seen in Fig. 3, a layer that guarantees positive constraint forces to achieve this objective is added to the controller. To compute these forces, the internal wrench must be calculated. Therefore, Newton's model, represented in (14), will be employed to determine the constraint wrenches for this system.

$$\mathbb{M}\dot{\mathbf{t}} + \mathbb{W}\mathbb{M}\mathbf{t} - \mathbf{w}_C = \mathbf{w}_A \tag{14}$$

In (14), Matrix $\mathbb{M}$ is rigid bodies mass and moment of inertia, called *inertia dyad*, Matrix $\mathbb{W}$ is called *angular-velocity dyad*, $\mathbf{t}$ denotes rigid body twist, $\mathbf{w}_C$ is nonworking constraint wrench and $\mathbf{w}_A$ is working wrench of system. A property of this dynamic model is that it can be separated into the dynamics of each body, allowing for the computation of all wrenches, including working and nonworking, on each body [14]. Rigid body twist can be computed from (13) by twist shaping matrix, which contains Jacobian matrix as in (18).

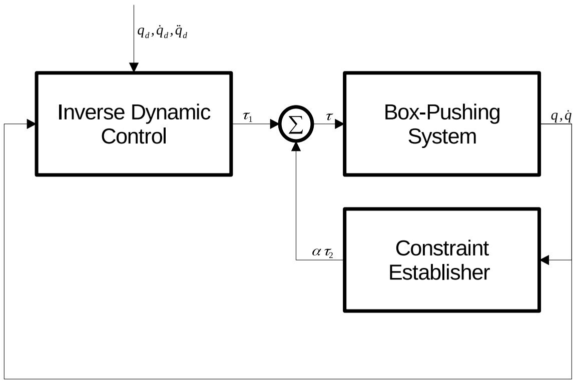


Fig. 3. Control Block Diagram

$$\mathbf{t} = \mathbf{T}(\mathbf{q})\dot{\mathbf{q}} \tag{15}$$

$$\dot{\mathbf{t}} = \dot{\mathbf{T}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{T}(\mathbf{q})\ddot{\mathbf{q}} \tag{16}$$

$$\mathbf{w}_A = \mathbf{T}^T(\mathbf{q})\boldsymbol{\tau}, \tag{17}$$

$$\mathbf{T}(\mathbf{q}) = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times4} \\ \multicolumn{2}{c}{\mathbf{J}(\mathbf{q})} \end{bmatrix} \tag{18}$$

The system model can assess the constraint wrench by calculating $\dot{\mathbf{t}}$. The constraint wrench vector consistently exists in the null space of $\mathbf{T}$. It can be managed by providing actuating forces and torques within the null space of the transpose of matrix $\mathbf{T}$. As the null space wrench does not impact the system's behavior, its sole influence must be on the constraint wrench vector to prevent it from moving in (14). Box-pushing constraints refer to internal forces acting on a box's sides that do not contribute to its movement. They are essentially nonworking wrenches and must be positive because of attachment constraints. The controller is in (21) and $\mathbf{J}^{\dagger}(\mathbf{q})$ is pseudo-inverse of $\mathbf{J}(\mathbf{q})$.

$$\boldsymbol{\tau}_1 = \mathbf{J}^{\dagger}(\mathbf{q})\left(\mathbf{M}(\mathbf{q})\mathbf{a} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}\right) \tag{19}$$

$$\boldsymbol{\tau}_2 = \text{Null}(\mathbf{J}^T(\mathbf{q}))\boldsymbol{\zeta}\begin{bmatrix} r_2\nu & 1 & r_3\nu & 1 & r_4\nu & 1 \end{bmatrix}^T \tag{20}$$

$$\boldsymbol{\tau} = \boldsymbol{\tau}_1 + \alpha\boldsymbol{\tau}_2 \tag{21}$$

$$\mathbf{a} = \ddot{\mathbf{q}}_d + \mathbf{K}_D(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_P(\mathbf{q}_d - \mathbf{q}) \tag{22}$$

$$\nu = \frac{\sum_{i=1}^{4}(q_3 - q_{(i+3)})r_i}{\sum_{i=1}^{4}(d_i - r_i)} \tag{23}$$

$$\alpha = \|[w_{c_4}, w_{c_8}, w_{c_{10}}, w_{c_{14}}]\|_{\infty} + 1 \tag{24}$$

$$\boldsymbol{\zeta} = \mathbf{I}_4 \otimes \mathbf{Q}_{q_3} : \otimes \text{ is Kronecker product} \tag{25}$$

$$\mathbf{Q}_{q_3} = \begin{bmatrix} \cos(q_3) & -\sin(q_3) & 0 \\ \sin(q_3) & \cos(q_3) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{26}$$

TABLE III
CONTROLLER PARAMETERS

Parameters Symbol	Parameter Value
$\mathbf{K}_P$	$\mathbf{I}_{7\times7}$
$\mathbf{K}_D$	$\mathbf{I}_{7\times7}$
$q_{d_1}(t)$	$\cos(0.2\pi t)$
$q_{d_2}(t)$	$\sin(0.2\pi t)$
$q_{d_3}(t)$	$\cos(0.4\pi t)$
$q_{d_4}(t)$	$\cos(0.4\pi t)+\sin(0.6\pi t+\pi/4)$
$q_{d_5}(t)$	$\cos(0.4\pi t)+\sin(0.6\pi t+3\pi/4)$
$q_{d_6}(t)$	$\cos(0.4\pi t)+\sin(0.6\pi t+5\pi/4)$
$q_{d_7}(t)$	$\cos(0.4\pi t)+\sin(0.6\pi t+7\pi/4)$

By considering relations (19)-(26) and defining error as $\mathbf{e} = \mathbf{q} - \mathbf{q}_d$, stability of the system can be shown as (29).

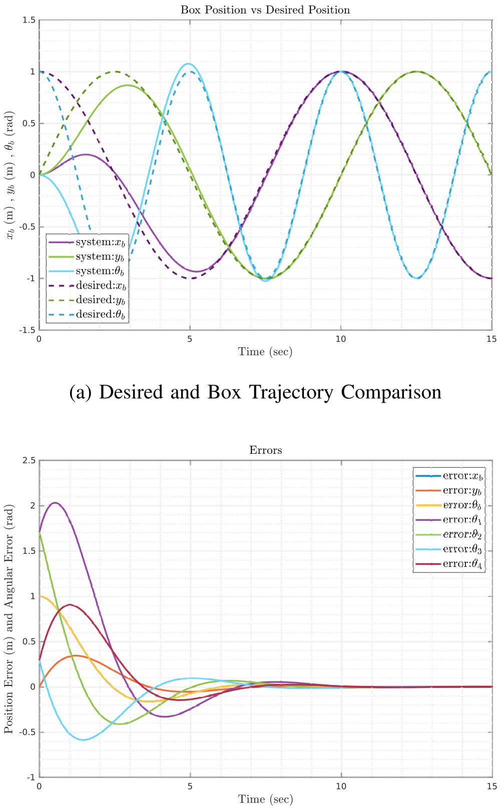
$$\begin{aligned}
\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} &= \mathbf{J}^T\boldsymbol{\tau} \\
&= \mathbf{J}^T(\mathbf{q})\mathbf{J}^\dagger(\mathbf{q})\left(\mathbf{M}(\mathbf{q})\mathbf{a} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}\right) \\
&\quad + \alpha\mathbf{J}^T(\mathbf{q})\boldsymbol{\tau}_2 \\
&= \mathbf{M}(\mathbf{q})\mathbf{a} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} && (27) \\
\Rightarrow \ddot{\mathbf{q}} &= \mathbf{a} \\
&= \ddot{\mathbf{q}}_d - \mathbf{K}_D\dot{\mathbf{e}} - \mathbf{K}_P\mathbf{e} && (28) \\
\Rightarrow \mathbf{0} &= \ddot{\mathbf{e}} + \mathbf{K}_D\dot{\mathbf{e}} + \mathbf{K}_P\mathbf{e} && (29)
\end{aligned}$$

It is observed that the error dynamics exhibit stability, indicating that precise tracking of the intended trajectory can be accomplished with a high degree of accuracy. The controller demands a positive constraint force greater than 1, which the null space part achieves.

IV. SIMULATIONS

MATLAB/Simulink 2022b simulates the system using the Simscape Multibody model. Table III represents the controller coefficients and desired path. Simulation initial condition is $\mathbf{q}(0) = \mathbf{0}_{7\times1}$ and $\dot{\mathbf{q}}(0) = \mathbf{0}_{7\times1}$. Upon analyzing and incorporating the controller output into the Simscape model, the system's outcome is visually depicted in Fig. 4. The simulation evaluates actuators and internal forces, as shown in Fig. 5.

The desired route for the box and agents involves various frequencies and phases. This is represented by a dashed line in Fig. 4a for the box states, while the box states path through time is shown by a simple line. To enhance the comprehension of the system's performance across all states, the tracking errors are displayed in Fig. 4b, where they tend to be zero. According to the stability analysis in (33), it can be concluded that the error is exponentially stable, as depicted in Fig. 4b. As can be seen, All state's errors are simple second-degree systems with poles on $0.5 \pm j\frac{\sqrt{3}}{2}$ and different initial states. The initial values for both x_b and θ_b errors are the same: $e_{x_b,\theta_b}(0) = 1$ and $\dot{e}_{x_b,\theta_b}(0) = 0$. However, the initial conditions for the y_b error dynamics are different, with $e_{y_b}(0) = 1$ and $\dot{e}_{y_b}(0) = 1$. Fig. 4b shows that both x_b and θ_b errors overlap, resulting in an undershoot of around 16.30% which is consistent with (29) due to their unique initial conditions and dynamics similarity. The errors also exhibit a settling time property, with all initial conditions taking 8 seconds to reach an absolute value of less than 2% of their initial absolute values.



(a) Desired and Box Trajectory Comparison

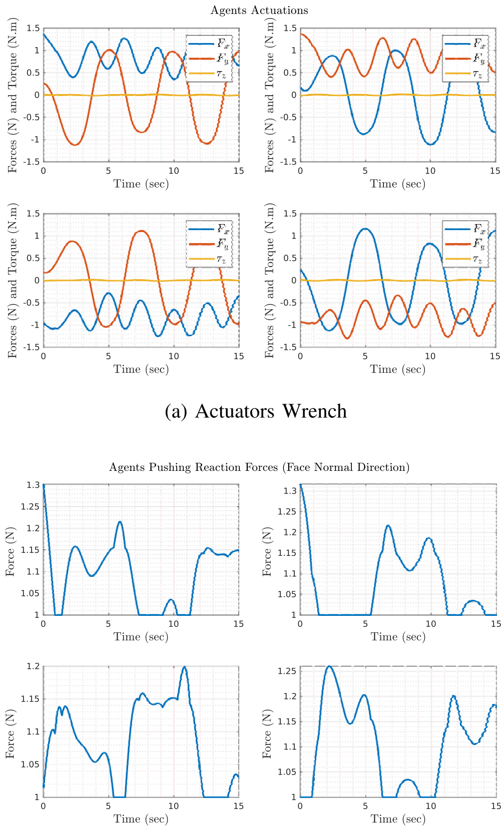
(b) System Tracking Error

Fig. 4. Tracking Performance of Closed Loop System

The controller guides the system along the desired path and sets its constraints. The fulfillment of the established limitations can be observed in the depicted Fig. 5b. As shown in Fig. 5b, there are some non-smooth forces at certain times on the actuators and internal wrench, But because of controller design, this property doesn't affect the continuity of agents' actuation. Besides, agent actuation in Fig. 5a illustrates a system with no chattering or unlimited wrench, demonstrating the controller's performance and demand feasibility.

V. CONCLUSION

This study provides a detailed and comprehensive model for controlling a holonomic mobile robot box-pushing system with stringent non-slipping constraints. The model takes into consideration the complex interactions and dynamics that occur between multiple robots and the box. Effective control strategies can be implemented using mathematical models determining constraint and actuation forces. Furthermore, the simulation results demonstrate the efficacy of the proposed



(a) Actuators Wrench

(b) Internal Forces

Fig. 5. Actuators and Internal Wrench

approach, while the stability analysis confirms the validity of the results by showcasing the behavior of the controlled system.

References

[1] H. Sayyaadi and M. Babaee, “Control of nonholonomic mobile manipulators for cooperative object transportation,” *Scientia Iranica*, vol. 21, no. 2, pp. 347–357, 2014.

[2] L. E. Parker, “Current research in multirobot systems,” *Artificial Life and Robotics*, vol. 7, pp. 1–5, Mar 2003.

[3] Y. Kobayashi and S. Hosoe, “Cooperative enclosing and grasping of an object by decentralized mobile robots using local observation,” *International Journal of Social Robotics*, vol. 4, pp. 19–32, Nov 2012.

[4] H. Sayyaadi and M. Babaee, “Control of nonholonomic mobile manipulators for cooperative object transportation,” *Scientia Iranica*, vol. 21, no. 2, pp. 347–357, 2014.

[5] A. Abbaspour, K. Alipour, H. Zare Jafari, and S. A. A. Moosavian, “Optimal formation and control of cooperative wheeled mobile robots,” *Comptes Rendus Mécanique*, vol. 343, no. 5, pp. 307–321, 2015.

[6] A. Yufka and M. Ozkan, “Formation-based control scheme for cooperative transportation by multiple mobile robots,” *International Journal of Advanced Robotic Systems*, vol. 12, no. 9, p. 120, 2015.

[7] H. Farivarnejad, S. Wilson, and S. Berman, “Decentralized sliding mode control for autonomous collective transport by multi-robot systems,” in *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 1826–1833, 2016.

[8] S. Wilson, R. Gameros, M. Sheely, M. Lin, K. Dover, R. Gevorkyan, M. Haberland, A. Bertozzi, and S. Berman, “Pheeno, a versatile swarm robotic research and education platform,” *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 884–891, 2016.

[9] K. Alipour and A. Abbaspour, “The effect of remote center compliance parameters on formation control of cooperative wheeled mobile robots for object manipulation,” *International Journal of Control, Automation and Systems*, vol. 16, pp. 306–317, Feb 2018.

[10] X. Li, X. Liu, G. Wang, S. Han, C. Shi, and H. Che, “Cooperative target enclosing and tracking control with obstacles avoidance for multiple nonholonomic mobile robots,” *Applied Sciences*, vol. 12, no. 6, p. 2876, 2022.

[11] J. J. Craig, *Introduction to Robotics: Mechanics and Control*. Addison-Wesley series in electrical and computer engineering: control engineering, Pearson/Prentice Hall, 2005.

[12] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. Wiley, 2005.

[13] H. D. Taghirad, *Parallel Robots: Mechanics and Control*. Taylor & Francis, 2013.

[14] J. Angeles, *Fundamentals of robotic mechanical systems: theory, methods, and algorithms*. Mechanical engineering series, Springer, 4th ed., 2013.

Appendix

A. Jacobain Matrix

The Jacobian matrix of a system’s model can be obtained using this equation:

$$\mathbf{J}(\mathbf{q}) = \boldsymbol{\zeta}\times$$

$$\begin{bmatrix}
\cos(q_3) & \sin(q_3) & -r_1(q_3-q_4) & 0 & 0 & 0 & 0 \\
-\sin(q_3) & \cos(q_3) & r_1-d_1 & -r_1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
\cos(q_3) & \sin(q_3) & d_2-r_2 & 0 & r_2 & 0 & 0 \\
-\sin(q_3) & \cos(q_3) & -r_2(q_3-q_5) & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 \\
\cos(q_3) & \sin(q_3) & r_3(q_3-q_6) & 0 & 0 & 0 & 0 \\
-\sin(q_3) & \cos(q_3) & d_3-r_3 & 0 & 0 & r_3 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 \\
\cos(q_3) & \sin(q_3) & r_4-d_4 & 0 & 0 & 0 & -r_4 \\
-\sin(q_3) & \cos(q_3) & r_4(q_3-q_7) & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

where $\boldsymbol{\zeta}$ mentioned in (25).

B. Inertia Matrix

The Inertia matrix for a system’s model can be obtained by using the following equation where the $M_{jk}(\mathbf{q})$ is an element of this matrix:

$$\mathbf{M}(\mathbf{q}) = \mathbf{J}(\mathbf{q})^T \text{diag}\left(\mathbf{K}_1, ..., \mathbf{K}_4\right) \mathbf{J}(\mathbf{q})$$

$$+ \,\text{diag}(M_b, M_b, I_b, 0, 0, 0, 0) = [M_{jk}(\mathbf{q})]$$

$$i = \{1, 2, 3, 4\}, \mathbf{K}_i = \text{diag}(m_i, m_i, I_i)$$

C. Coriolis-Centrifuge Matrix

The equation for obtaining the Coriolis-Centrifuge matrix of a system’s model is:

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \left[C_{ij}(\mathbf{q}, \dot{\mathbf{q}})\right], i = \{1, 2, ..., 7\}, j = \{1, 2, ..., 7\}$$

$$C_{ij}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2}\sum_{k=1}^{7} C_{ijk}(\mathbf{q})\dot{q}_k$$

$$C_{ijk}(\mathbf{q}) = \frac{\partial}{\partial q_k} M_{ij}(\mathbf{q}) + \frac{\partial}{\partial q_j} M_{ik}(\mathbf{q}) - \frac{\partial}{\partial q_i} M_{kj}(\mathbf{q})$$