Nonlinear Model Predictive Control of Rover Robotics System

Summary based on the paper by Serdar Kalaycioglu and Anton de Ruiter

Abstract

This paper summarizes the work of Kalaycioglu and de Ruiter on the real-time force and motion control of multi-rover robotics systems carrying a common load, addressing challenges such as wheel and joint torque saturation. The system, characterized by non-linearity, underdetermination, and non-holonomic constraints, is controlled using two robust algorithms: Optimal Control Allocation (OCA) and Nonlinear Model Predictive Control (NMPC). The system comprises two rovers with mecanum wheels and 7-DOF redundant arms. The paper outlines the kinematic and dynamic models, details the OCA and NMPC techniques, and presents simulation results demonstrating NMPC's superior performance in minimizing torques and contact forces.

1 Introduction

Exploration of complex environments, such as space, construction, mining, and military applications, relies heavily on mobile rovers. Space Robotics Exploration missions, particularly for assembling large space structures, require multiple rovers with mounted robotic manipulators to coordinate tasks. Challenges in load-sharing among multi-rover systems, especially under real-time force and motion control, remain significant due to wheel and joint torque saturation.

Previous research, such as Necsulescu's work on collision-free trajectories and force control, addressed vehicle motion but often ignored dynamics in non-holonomic systems. Standard Model Predictive Control (MPC) struggles with complex robotics systems, prompting advancements in Nonlinear Model Predictive Control (NMPC) facilitated by fast gradient methods and input parameterization. While mecanum wheel locomotion has been studied, controlling multiple rovers with arms carrying a common load is still nascent.

2 Description

The robotics system consists of two rovers, each equipped with four mecanum wheels and two 7-degree-of-freedom (DOF) redundant arms designed to carry a common load. The system is underdetermined with non-holonomic constraints, aiming to mitigate wheel and joint torque saturation during heavy payload manipulation.

Each rover has a center of mass C_i with pose defined by rotation angle ψ_i and position vector $\mathbf{R}_{c_i}$ in an inertial coordinate system (X, Y, Z). Coordinate axes $(x_{c_i}, y_{c_i}, z_{c_i})$ are attached to C_i, rotated around the Z-axis by ψ_i. Rover masses are m_{c_i}, and wheel masses are $m_{w_{ij}}$ ($j = 1, \ldots, 4$, $i = 1, 2$). Wheel centers are separated by distances $2a$ (along y_{c_i}) and $2b$ (along x_{c_i}). Wheels have radius s, angular rotation ϕ_{ij}, and angular rate ω_{ij}. Mecanum wheels feature rollers at angle β_{ij} between the roller's rotation axis and the x_{c_i}-axis.

3 Mathematical Framework

3.1 Model of Kinematics

The kinematics describe motion without considering forces.

- **Wheel Velocity**: The velocity of the j-th wheel center of the i-th rover, $\mathbf{V}_{m_{ij}}$, is:

$$\mathbf{V}_{m_{ij}} = \mathbf{V}_{c_i} + \mathbf{\Omega}_{c_i} \times \mathbf{r}_{w_{ij}},$$

  where $\mathbf{V}_{c_i}$ is the rover's mass center velocity, $\mathbf{\Omega}_{c_i} = \dot{\psi}_i \mathbf{e}_z$ is the angular velocity, and $\mathbf{r}_{w_{ij}}$ is the position vector from the rover's mass center to the wheel center.

- **Roller Velocity**: The velocity of a point P at the roller center, $\mathbf{V}_{p_{ij}}$, is:

$$\mathbf{V}_{p_{ij}} = \mathbf{V}_{m_{ij}} + \omega_{ij} \times \rho_{ij},$$

  where ρ_{ij} is the position vector from the wheel center to point P.

- **No-Slip Condition**: For non-slipping rollers, $\mathbf{V}_{p_{ij}} \cdot \mathbf{e}_{\beta_{ij}} = 0$, leading to:

$$\mathbf{V}_{m_{ij}} \cdot \mathbf{e}_{\beta_{ij}} = \omega_{ij} s (\mathbf{e}_{x_{c_i}} \cdot \mathbf{e}_{\beta_{ij}}).$$

  For mecanum wheels with $\beta_{ij} = 45°$, rover velocities are:

$$\mathbf{V}_{c_i} = \begin{bmatrix} s(\omega_{i1} + \omega_{i2})/2 \\ s(\omega_{i3} - \omega_{i1})/2 \\ 0 \end{bmatrix}, \quad \mathbf{\Omega}_{c_i} = \begin{bmatrix} 0 \\ 0 \\ s(\omega_{i3} - \omega_{i1})/(2(a+b)) \end{bmatrix},$$

  with $\omega_{i4} = \omega_{i1} + \omega_{i2} - \omega_{i3}$.

- **Robot Arm Kinematics**: Using the Denavit-Hartenberg convention, the transformation matrix $\mathbf{T}_f^g$ defines arm frames. Jacobian matrices $\mathbf{J}_{c_k}^i$ and $\dot{\mathbf{J}}_{c_k}^i$ compute velocities and accelerations at point k on the arm:

$$\begin{bmatrix} \mathbf{\Omega}_k \\ \mathbf{V}_k \end{bmatrix}_i = \mathbf{J}_{c_k}^i \begin{bmatrix} \mathbf{\Omega}_{c_i} \\ \dot{\theta}_i \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{V}_{c_i} \end{bmatrix},$$

$$\begin{bmatrix} \dot{\mathbf{\Omega}}_k \\ \dot{\mathbf{V}}_k \end{bmatrix}_i = \mathbf{J}_{c_k}^i \begin{bmatrix} \dot{\mathbf{\Omega}}_{c_i} \\ \ddot{\theta}_i \end{bmatrix} + \dot{\mathbf{J}}_{c_k}^i \begin{bmatrix} \mathbf{\Omega}_{c_i} \\ \dot{\theta}_i \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \dot{\mathbf{V}}_{c_i} \end{bmatrix}.$$

3.2 Model of Dynamics

The dynamics are derived using the Lagrangian formulation, with total kinetic energy T_t comprising rotational and translational components of arms and rovers. The Lagrangian equation is:

$$\frac{d}{dt}\left(\frac{\partial T_t}{\partial \dot{q}_h}\right) - \frac{\partial T_t}{\partial q_h} = Q_h,$$

where q_h are generalized coordinates (wheel and joint angles), and Q_h are generalized forces. The system dynamics are:

$$\mathbf{G}\ddot{\mathbf{q}} + \mathbf{c} = \tau,$$

where $\mathbf{G}$ is the mass/inertia matrix, $\mathbf{c}$ includes Coriolis and centrifugal terms, and τ comprises joint torques $\tau_{\theta_L}, \tau_{\theta_R}$ and wheel moments $\mathbf{M}_L, \mathbf{M}_R$.

3.3 Optimal Control Allocation (OCA) Technique

OCA is a two-stage control method for the underdetermined system.

- **Stage 1: Reference Trajectory Generation**: Impedance control generates end-effector trajectories:
$$\ddot{\mathbf{X}}_i = \mathbf{M}_i^{-1}\mathbf{B}(\dot{\mathbf{X}}_{d_i} - \dot{\mathbf{X}}_i) + \mathbf{M}_i^{-1}\mathbf{K}(\mathbf{X}_{d_i} - \mathbf{X}_i),$$
where $\mathbf{M}_i, \mathbf{K}, \mathbf{B}$ are positive definite, $\mathbf{X}_i$ are end-effector trajectories, and $\mathbf{X}_{d_i}$ are reference trajectories. Inverse kinematics solve for joint rates and accelerations via least-squares minimization.

- **Stage 2: Optimal Control Allocation**: Minimizes a quadratic cost function:
$$C = \frac{1}{2}\tilde{\mathbf{S}}^T\mathbf{H}\tilde{\mathbf{S}} + \tilde{\lambda}^T\mathbf{E},$$
where $\mathbf{H}$ is a weighting matrix, $\tilde{\lambda}$ is the Lagrangian multiplier, $\mathbf{E}$ represents constraint equations, and $\tilde{\mathbf{S}}$ includes contact forces/moments, wheel moments, and joint torques. The solution is:
$$\tilde{\mathbf{S}} = \mathbf{\Delta}^{-1}\mathbf{P},$$
with $\mathbf{\Delta}$ and $\mathbf{P}$ defined by system parameters and dynamics.

3.4 Nonlinear Model Predictive Control (NMPC) Technique

NMPC replaces OCA's second stage, using impedance control trajectories as input.

- **Core Concept**: NMPC predicts system behavior over a horizon T_p, optimizing a cost function while respecting constraints on states, outputs, and inputs.

- **Conventional MPC Formulation**:
$$C = \int_0^{T_p} \left[(\mathbf{y}(t) - \mathbf{y}_r(t))^T\mathbf{K}(\mathbf{y}(t) - \mathbf{y}_r(t)) + \mathbf{S}(t)^T\mathbf{H}\mathbf{S}(t)\right] dt,$$
subject to:
$$\dot{\mathbf{y}} = \mathbf{g}(\mathbf{y}) + \mathbf{L}\mathbf{S}, \quad \mathbf{z} = \mathbf{g}_z(\mathbf{y}) + \mathbf{H}\mathbf{S}, \quad \mathbf{y}(0) = \mathbf{y}(t_0).$$

- **Discretized NMPC Formulation**: The discrete state-space model is:
$$\mathbf{y}(k_t + 1) = \mathbf{A}_g(k_t)\mathbf{y}(k_t) + \mathbf{B}_g(k_t)\mathbf{S}(k_t),$$
$$\mathbf{z}(k_t) = \mathbf{C}_g(k_t)\mathbf{y}(k_t) + \mathbf{D}_g(k_t)\mathbf{S}(k_t).$$
The cost function is:
$$C = \frac{1}{2}\sum_{i=1}^{N_p} \left[(\mathbf{y}(k_t+i) - \mathbf{y}_r(k_t+i))^T\mathbf{K}(\mathbf{y}(k_t+i) - \mathbf{y}_r(k_t+i)) + \mathbf{S}(k_t+i-1)^T\mathbf{H}\mathbf{S}(k_t+i-1)\right].$$

4 Computer Simulation Results and Discussion

Simulations minimized joint and wheel torques and contact forces while tracking end-effector poses, using parameters based on a mini-SSRMS. Both OCA and NMPC achieved efficient performance, with NMPC outperforming OCA in reducing torques and contact forces due to its advanced optimization scheme. The conventional least-squares method was least effective.

5 Conclusions and Future Work

The paper presented OCA and NMPC as robust, efficient solutions for controlling a multi-rover system with non-holonomic constraints and torque saturation. NMPC demonstrated superior performance. Future work includes experimental validation via a testbed and incorporating normal forces to relax the no-slip assumption, accounting for dynamic friction limits.