

A Model Predictive Control Approach for Vision-based Object Grasping via Mobile Manipulator

Michalis Logothetis, George C. Karras, Shahab Heshmati-alamdar, Panagiotis Vlantis and Kostas J. Kyriakopoulos

Abstract— This paper presents the design of a vision-based object grasping and motion control architecture for a mobile manipulator system. The optimal grasping areas of the object are estimated using the partial point cloud acquired from an on-board RGB-D sensor system. The reach-to-grasp motion of the mobile manipulator is handled via a Nonlinear Model Predictive Control scheme. The controller is formulated accordingly in order to allow the system to operate in a constrained workspace with static obstacles. The goal of the proposed scheme is to guide the robot's end-effector towards the optimal grasping regions with guaranteed input and state constraints such as occlusion and obstacle avoidance, workspace boundaries and field of view constraints. The performance of the proposed strategy is experimentally verified using an 8 Degrees of Freedom KUKA Youbot in different reach-to-grasp scenarios.

I. INTRODUCTION

During the last decades, the employment of robotic systems in various fields of industry, society, medicine, agriculture and security [1] has been significantly increased. In many of these tasks, the robots interact with the environment as well as with humans, in order to facilitate, improve and expedite specific operations that were usually carried out by humans exclusively. In this context, object manipulation is a fundamental robotic operation which typically involves detection/recognition and grasping control algorithms. For example, an interaction robot should be able to pick up an object from a human, transport it through a non obstacle-free environment and deliver it in a specific location. Hence, the investigation and development of robust reach-to-grasp algorithms for robotic systems operating in partially known environments (e.g., warehouses) is of utmost importance.

Numerous studies have been reported regarding the design of control algorithms for feasible and stable robotic grasp. Issues such as force-closure, task compatibility and other desired properties have been investigated [2], [3], [4], [5], along with the problem of grasp selection and planning [6], [7].

However, robust grasping also requires precise detection of the object in 3D space. This challenging issue can be encountered by employing vision based methods which either relay on the full 3D model of the object in order to calculate the optimal grasping areas by using multiple images and/or point clouds [8], [9], [10], [11] or other methods that

The authors are with the Control Systems Lab, Department of Mechanical Engineering, National Technical University of Athens, 9 Heron Polytechniou Street, Zografou 15780, Greece. Email: {logothm,karrasg,shahab,vlantis,kkyria@mail.ntua.gr}

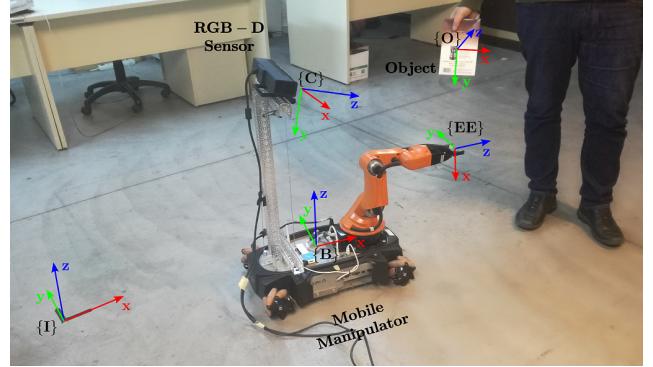


Fig. 1: Youbot Mobile Manipulator and Reference Frames

evaluate only the partial information of the object [12], [13], [14], [15].

When the desired location of the grasp has been identified, the fingers or the gripper of the robot should be guided to the computed grasping pose through a collision-free and feasible trajectory. Different techniques, such as visual servoing, path planning and haptic feedback on the fingertips can be used to pick up the object.

In [16] a combined vision-force approach was employed in order to guide a robotic hand during a grasping and manipulation task. Moreover, the authors in [17] presented a hybrid visual servoing approach for grasping and manipulation using a humanoid where force feedback was incorporated in order to deal with possible contact events. However, neither occlusion avoidance nor Field of View (FoV) constraints were taken into account in both studies. In [18], the authors proposed a vision-based Q-learning control scheme for a mobile manipulator system where FoV constraints were included. Nevertheless, possible restrictions such as joint limits, obstacle or occlusion avoidance were not addressed.

Also, probabilistic approaches are widely used for collision-free motion planning of redundant robotic systems such as mobile manipulators, due to high-dimensionality of the configuration space. In [19], [20], an integrated approach using Rapidly-Exploring Random Trees (RRTs) for effectively grasping or re-grasping of objects was proposed. In a similar vein, authors in [21] proposed a method for solving both grasp and motion planning simultaneously. However, the aforementioned studies do not deal with dynamically changing environments (possible motion of the object, noisy RGB-D measurements etc.) while the object was assumed a

priori known.

The motion control of mobile manipulators in constrained environments is a challenging problem, where multiple input and state constraints are imposed to the system. Non-linear Model Predictive Control (NMPC) [22], is an ideal approach for complex robot missions, as it is able to combine motion planning, obstacle avoidance and workspace restrictions, while handling efficiently input and state constraints.

Recent studies investigate the use of MPC in mobile robots. In [23], the authors used a dipolar vector field and NMPC with FoV constraints in order to navigate a non-holonomic vehicle in an obstacle-free environment. In [24], [25] NMPC was employed in multiple omni-directional mobile robots for target tracking and formation control tasks. Moreover, in [26] a similar to RRTs approach was combined with MPC for the navigation of mobile robots.

On the other hand, more limited is the span of literature in MPC for mobile manipulator systems. In [27], an MPC approach for a mobile manipulator without obstacles was presented. A self-triggered MPC framework for image based visual servoing of a mobile manipulator was described in [28]. Nevertheless, neither obstacle nor occlusion avoidance were considered. In [29], the authors introduce a constrained predictive control algorithm for a holonomic mobile manipulator. Restrictions in the form of acceleration, velocity, position, and obstacle avoidance were taken into account. An extended version is presented in [30], where time varying desired configurations have been taken under consideration for a linearized system of a mobile manipulator. Nonetheless, no grasping was considered in both studies.

In this work, we present a complete strategy for vision-based object grasping via a Mobile Manipulator System (MMS). An algorithm based on [15] is implemented in order to calculate *on-line* the optimal grasping areas of an unknown object handled by a human. The object is detected and tracked via an RGB-D sensor which is rigidly mounted on the system platform (Fig. 2). An NMPC motion scheme is designed to guide the mobile manipulator through a feasible and collision-free path in order to pick up the object from a human. A complex set of constraints such as obstacle and occlusion avoidance, field of view and workspace constraints, are satisfied during the reach-to-grasp operation. The performance of the proposed strategy is experimentally verified using an 8 DoF KUKA Youbot [31] in reach and grasp scenarios involving unknown objects of different geometry. The overall system architecture is depicted in Fig. 2.

The rest of the paper is organized as follows: Section II presents an analytical description of the proposed methodology which consists of: i) the MMS model ii) the object detection algorithm iii) the optimal grasping area computation method and iv) the NMPC formulation. Section III demonstrates the applicability and performance of the proposed control scheme via different experimental scenarios. Finally, Section IV concludes the paper.

II. METHODOLOGY

A. Mobile Manipulator System Model

In this work we consider a KUKA Youbot MMS which consists of a 5 DoF arm mounted on an omni-directional platform with 4 swedish mecanum wheels. The overall setup results in a redundant 8 DoF MMS (Fig. 1). The kinematic equation of the complete system is given in discrete form by:

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \mathbf{A}(\mathbf{q}_k, \mathbf{u}_k) \Delta T \quad (\text{II.1})$$

where $\mathbf{q}_k = [\mathbf{q}_{arm_k}, \mathbf{q}_{base_k}]^T$ is the state vector consisting of the arm joint states $\mathbf{q}_{arm_k} = [q_{0_k}, q_{1_k}, q_{2_k}, q_{3_k}, q_{4_k}]^T$ and the base pose states $\mathbf{q}_{base_k} = [\mathbf{I}x_{B_k}, \mathbf{I}y_{B_k}, \mathbf{I}\psi_{B_k}]^T$, respectively. The latter are defined with respect to the Inertial Frame \mathbf{I} as shown in Fig. 1. In Eq. II.1 $\mathbf{u}_k = [\mathbf{u}_{arm_k}, \mathbf{u}_{base_k}]^T$ is the complete system input vector which consists of the joint velocities of the arm $\mathbf{u}_{arm_k} = [\omega_{q_{0_k}}, \omega_{q_{1_k}}, \omega_{q_{2_k}}, \omega_{q_{3_k}}, \omega_{q_{4_k}}]^T$ and the wheel velocities (front left and right, back left and right respectively) $\mathbf{u}_{base_k} = [\omega_{FL_k}, \omega_{FR_k}, \omega_{BL_k}, \omega_{BR_k}]^T$ of the omni-directional base. A more detailed view of Eq. II.1 reveals a linear kinematic model for the arm:

$$\mathbf{q}_{arm_{k+1}} = \mathbf{q}_{arm_k} + \mathbf{u}_{arm_k} \Delta T \quad (\text{II.2})$$

while the base kinematics is obviously non-linear due to transformation of the velocities from the Body frame \mathbf{B} to the Inertial frame \mathbf{I} :

$$\mathbf{q}_{base_{k+1}} = \mathbf{q}_{base_k} + \frac{R_w}{4} \mathbf{I} \mathbf{R}_B \mathbf{T}_{CA} \mathbf{u}_{base_k} \Delta T \quad (\text{II.3})$$

where $\mathbf{I} \mathbf{R}_B = \begin{bmatrix} \cos \psi_k & -\sin \psi_k & 0 \\ \sin \psi_k & \cos \psi_k & 0 \\ 0 & 0 & 1 \end{bmatrix}$ denotes the transformation from \mathbf{B} to \mathbf{I} and

$$\mathbf{T}_{CA} = \begin{bmatrix} 1 & -1 & 1 & -1 \\ -1 & -1 & 1 & 1 \\ -\frac{1}{L+\ell} & -\frac{1}{L+\ell} & -\frac{1}{L+\ell} & -\frac{1}{L+\ell} \end{bmatrix} \quad (\text{II.4})$$

is the control allocation matrix which transforms body velocities to wheel velocity commands. R_w , L , ℓ are the geometrical properties of the base (wheel radius, vertical and horizontal lengths). The position and orientation of the MMS end-effector (ee) with respect to \mathbf{I} , is given by the forward kinematics of the complete system (arm and mobile base) as follows:

$${}^I \mathbf{p}_{ee} = \mathcal{F}_{ee}(\mathbf{q}) \quad (\text{II.5})$$

B. Object Detection

The object detection and tracking algorithm is based on the Point Cloud Library (PCL) [32] for 3D object tracking. The purpose of the library is to provide a comprehensive algorithmic base for the estimation of 3D object poses using Monte Carlo sampling techniques. The library is optimized to perform computations in real-time, by employing multi CPU

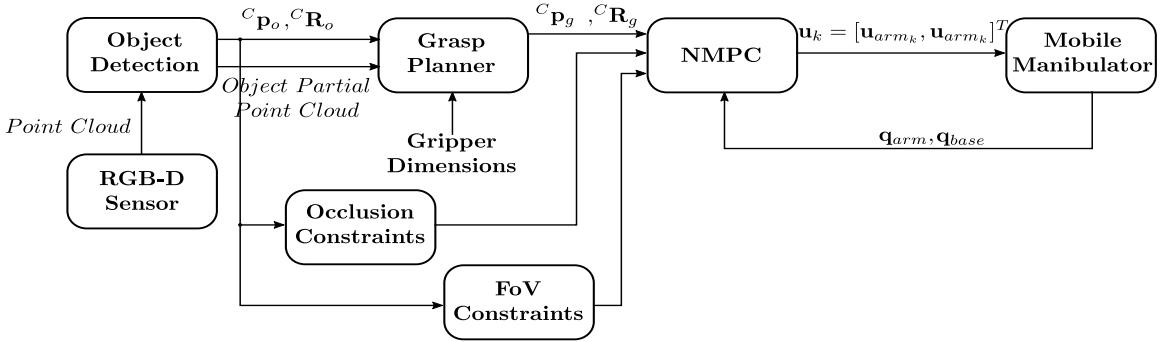


Fig. 2: Overall System Architecture

cores optimization, adaptive particle filtering (KLD sampling) and other modern techniques. The employed algorithm is based on standard 3D object tracking implementations, thus only the basic steps are briefly presented below:

- 1) The tracker is initialized by specifying the target object's point cloud. This procedure is performed by creating the segment model of the object when the tracking algorithm begins.
- 2) Using previous particles information about position and orientation, the pose of the next frame is predicted.
- 3) A likelihood formula is employed in order to calculate the weights of the particles.
- 4) An evaluation function compares the real point cloud data acquired from the RGB-D sensor with the predicted particles and performs re-sampling.

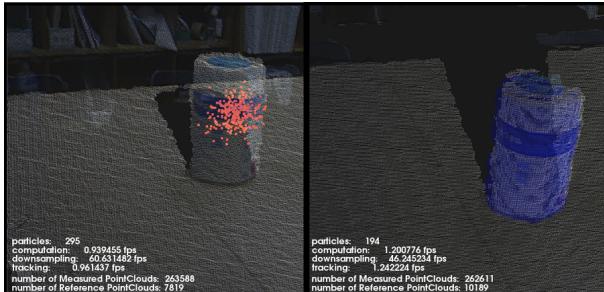


Fig. 3: Target Object Segmentation

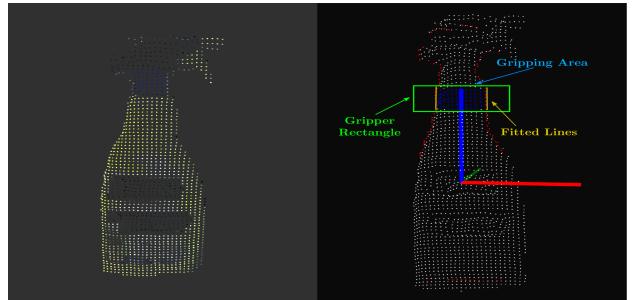
In Fig. 3 the reference model segmentation's cloud is presented in blue, while in red the particles' cloud is depicted. Moreover, the visible point clouds of the target object (Fig. 4b) are computed by applying the reference model segmentation's cloud as a 3D mask to the input cloud acquired by the RGB-D sensor.

C. Grasping Algorithm

An algorithm, inspired by [15], that computes *on-line* the optimal grasping area is integrated in the proposed reach-to-grasp architecture. The aforementioned grasping method has been selected due to its fast and robust performance. Moreover, it does not depends on off-line training data or 3D model of the object. The grasp planner relays only on the visible point cloud of the object and its characteristics of



(a) RGB-D Sensor Point Cloud



(b) Target Object Visible Points (c) Object Contour & Grasping Area

Fig. 4: Grasping Algorithm

the robotic gripper such as max opening, length and height (Fig. 5). The steps of the algorithm are presented in detail below:

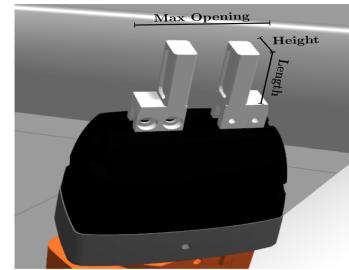


Fig. 5: Dimensions of Youbot Gripper

- 1) *Acquire the point cloud and the pose of the target object:* The visible points of the object and its pose are calculated from the object detection procedure

described in the Subsection II-B and therefore they are expressed in object local reference frame in order to process the point cloud and find the optimal grasping area.

- 2) *Transform the point cloud to its local frame:* The visible points are expressed in RGB-D sensor frame and they are transformed to its local frame in order to compute the optimal grasping area. The object frame and the transformed visible points (white colored) are depicted in Fig. 4c.
- 3) *Calculate the contour of the point cloud:* The Concave hull contour approach is used to extract the contour of the target object.
- 4) *Calculate the graspable zone:* After computing the contour of the target object the maximum gripper opening is taken into consideration in order to extract the graspable zone. The width of the contour is checked and if the points are out of the grasping range they are removed from the points of the contour. The graspable zone is defined by the red colored points in Fig. 4c.
- 5) *Compute the optimal grasping area:* Having the graspable zone that has been calculated above, the optimal pose of the gripper with respect to object is now computed using the gripper height (Fig. 5). The robotic gripper can be considered as a rectangle (depicted with green color in Fig. 4c) with height and width equal to height and max opening of the gripper, respectively. Moving this rectangle along $Z - axis$ of the object frame (the blue axis in Fig. 4c) from the z_{min} , which is the minimum z value of the all points on the graspable object zone (red points), till $z_{max} - \text{gripper height value}$ (z_{max} is the maximum value of $Z - axis$ of all the points on the graspable zone) the optimal area is computed. A force balance optimization method is used to evaluate every grasping area. More specifically, a straight line is fitted for the points on the two grasped sides (Fig. 4c) and the angle between the two resulting lines is computed. The bigger the angle is, the grasp quality is reduced for a parallel gripper. Consequently, the best grasp should be on two parallel straight lines. The smallest angle means the best and more reliable grasp. When the angle between the lines of two or more rectangles has the same value equal to the minimum one, the pose of the rectangle that is closer to the centroid of the target object's points is selected as the optimal one. The position of the rectangle w.r.t the target object frame ${}^O\mathbf{p}_g$ is calculated. The rotation of the optimal grasping pose is computed using the Rodrigues' rotation formula: ${}^O\mathbf{R}_g = \mathbb{I}_{3 \times 3} + [\mathbf{v}]_x + [\mathbf{v}]_x^2 \frac{1-c}{s^2}$, where $\mathbf{v} = \hat{\mathbf{v}}_l \times \hat{\mathbf{z}}$, $s = \|\mathbf{v}\|$ and $c = \hat{\mathbf{v}}_l \cdot \hat{\mathbf{z}}$. The $\hat{\mathbf{z}}$ is the unit vector along $Z - axis$ and $\hat{\mathbf{v}}_l$ is the unit vector along the one line of the optimal gripper rectangular. Finally, the position and orientation of the grasping area are expressed in RGB-D sensor frame: ${}^C\mathbf{p}_g = \begin{bmatrix} {}^C\mathbf{R}_O & {}^C\mathbf{p}_O \\ \mathbb{O}_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} {}^O\mathbf{p}_g \\ 1 \end{bmatrix}$ and ${}^C\mathbf{R}_g = {}^C\mathbf{R}_O {}^O\mathbf{R}_g$.

The grasping steps are given in Algorithm 1.

Algorithm 1 Grasping Algorithm

```

1: procedure FASTGRASP(maxOpening, height)
2:   top:
3:   ObjectPoints  $\leftarrow$  GetObjectVisiblePoints()
4:   ObjectPose  $\leftarrow$  GetObjectPose()
5:   if ObjectPoints.Size()  $> 0$  then
6:     ObjectPoints  $\leftarrow$  TransformPoints(ObjectPoints, ObjectPose)
7:     contour  $\leftarrow$  ConcaveHullContour(ObjectPoints)
8:     graspContour  $\leftarrow$  GraspableZone(contour, maxOpening)
9:      ${}^O\mathbf{R}_g, {}^O\mathbf{p}_g \leftarrow$  OptimalGrasp(graspContour, height)
10:     ${}^C\mathbf{R}_g, {}^C\mathbf{p}_g \leftarrow$  TransformPoint( ${}^O\mathbf{p}_O, {}^O\mathbf{ObjectPose}$ )
11:    return  ${}^C\mathbf{R}_g, {}^C\mathbf{p}_g$ 
12:   goto top

```

D. Model Predictive Control Scheme

As discussed previously, the proposed motion control scheme for the reaching and grasping task is based on the predictive control notion. More specific, at a given time instant k , the NMPC utilizes the current state measurement of the MMS q_k in order to solve an Optimal Control Problem (OCP) with respect to a control sequence $\mathbf{U}_k = \left[\mathbf{u}_{k|k}^T, \mathbf{u}_{k+1|k}^T, \dots, \mathbf{u}_{k+N-1|k}^T \right]^T$, for a prediction horizon N , with $\mathbf{u} = [\mathbf{u}_{\text{arm}}, \mathbf{u}_{\text{base}}]^T$.

1) *Optimization Problem:* First, for the sake of simplicity, we denote $\mathbf{x}_k = \mathcal{F}_{ee}(\mathbf{q}_k)$, that is the non-linear mapping of q at time step k to the end-effector pose ${}^I\mathbf{p}_{ee}$ with respect to frame I by employing Eq. II.5. Now, using the above declarations, the OCP of the NMPC is given as:

$$\begin{aligned} \min_{\mathbf{U}_k} J(\mathbf{x}_k, \mathbf{U}_k) = \\ \min_{\mathbf{U}(\cdot)} \sum_{i=0}^{N-1} F(\hat{\mathbf{x}}_{k+i|k}, \mathbf{u}_{k+i|k}) + V(\hat{\mathbf{x}}_{k+N|k}), \end{aligned} \quad (\text{II.6a})$$

subject to:

$$\hat{\mathbf{x}}_{k+i|k} = \mathcal{F}_{ee}(\hat{\mathbf{q}}_{k+i|k}), \quad i = 0, 1, \dots, N \quad (\text{II.6b})$$

$$\begin{aligned} \hat{\mathbf{q}}_{k+i|k} = \hat{\mathbf{q}}_{k+i-1|k} + \mathbf{A}(\hat{\mathbf{q}}_{k+i-1|k}, \mathbf{u}_{k+i-1|k}) \Delta T, \\ i = 1, \dots, N \end{aligned} \quad (\text{II.6c})$$

$$g_q(\hat{\mathbf{q}}_{k+i|k}) < 0, \quad i = 1, 2, \dots, N \quad (\text{II.6d})$$

$$g_u(\mathbf{u}_{k+i|k}) < 0, \quad i = 0, 1, \dots, N-1 \quad (\text{II.6e})$$

where $J(\cdot)$ is the optimal cost consisting of the running F and terminal V costs. At time instant k , the solution of the OCP ((II.6a)-(II.6e)) is providing an optimal control sequence, denoted as:

$$\mathbf{U}_{k,opt} = \left[\mathbf{u}_{k|k}^T, \mathbf{u}_{k+1|k}^T, \dots, \mathbf{u}_{k+N-1|k}^T \right] \quad (\text{II.7})$$

where the first control vector (i.e., $\mathbf{u}_{k|k}^T$) is applied to the system. Notice we use the double subscript notation for the predicted state of system (II.1) inside the OCP of the NMPC:

$$\hat{\mathbf{q}}_{k+i|k} = \hat{\mathbf{q}}_{k+i-1|k} + \mathbf{A}(\hat{\mathbf{q}}_{k+i-1|k}, \mathbf{u}_{k+i-1|k}) \Delta T$$

where the vector $\hat{\mathbf{q}}_{k+i|k}$ denotes the predicted state of the system (II.1) at sampling time $k + i$ with $i \in \mathbb{Z}_{\geq 0}$. The predicted state is based on the measurement of the system at sampling time k (i.e., provided by the on-board encoders

and odometry), while applying a sequence of control inputs $\left[\mathbf{u}_{k|k}^T, \mathbf{u}_{k+1|k}^T, \dots, \mathbf{u}_{k+i-1|k}^T\right]$. It holds that $\hat{\mathbf{q}}_{k|k} = \mathbf{q}_k$. The running cost function $F(\cdot)$, as well as the terminal cost $V(\cdot)$, are both of quadratic form, i.e., $F(\hat{\mathbf{x}}, \mathbf{u}) = \hat{\mathbf{x}}^T \mathbf{Q}_F \hat{\mathbf{x}} + \mathbf{u}^T \mathbf{R}_F \mathbf{u}$ and $V(\hat{\mathbf{x}}) = \hat{\mathbf{x}}^T \mathbf{P}_V \hat{\mathbf{x}}$, respectively, with \mathbf{Q}_F , \mathbf{R}_F and \mathbf{P}_V being positive definite matrices. Particularly, we select \mathbf{Q}_F as a positive definite matrix that penalizes the state error and \mathbf{R}_F as a positive semi-definite matrix that penalizes the control effort. In similar way, we select \mathbf{P}_V to be a positive definite matrix that penalizes the state error at the final prediction step. Moreover g_q and g_u are continuously differentiable functions denoting the state and input inequality constraints, respectively, and will be analyzed in detail in the following subsections.

Remark 1: The optimization problem is solved using gradient-based methods (i.e., Augmented Lagrange Multipliers, L-BFGS and Quadratic Interpolation algorithms). Thus, the cost and the inequality constraint functions must be continuously differentiable functions of the state and input trajectories.

2) *Obstacle Avoidance & Workspace Constraints:* Consider a robot which operates inside the workspace $\mathcal{W} \subset \mathbb{R}^3$ with boundaries $\partial\mathcal{W}$ and scattered obstacles located within. We may consider without loss of generality, that the robot and the obstacles can be modeled as spheres. Hence, let $\mathcal{B}({}^I\mathbf{p}_{MMS}, r)$ be a sphere with center ${}^I\mathbf{p}_{MMS} = [{}^Ix_{B_k}, {}^Iy_{B_k}]$ and radius r that surrounds the mobile platform and arm. The radius r depends directly on the configuration of the manipulator and it is the position of the end-effector w.r.t base frame \mathbf{B} which can be calculated using the arm forward kinematics. Moreover, the \mathcal{M} static obstacles within the workspace are also defined as spheres $\pi_m = \mathcal{B}({}^I\mathbf{p}_{\pi_m}, r_{\pi_m})$, $m \in \{1, \dots, \mathcal{M}\}$, where ${}^I\mathbf{p}_{\pi_m} \in \mathbb{R}^3$ is the center and the $r_{\pi_m} > 0$ the radius of the obstacle π_m . According to the property of spherical world [33], for each pair of obstacles $m, m' \in \{1, \dots, \mathcal{M}\}$ the following inequality holds:

$$\|\pi_m - \pi_{m'}\| > 2r + r_{\pi_m} + r_{\pi_{m'}} \quad (\text{II.8})$$

which intuitively means that the obstacles m and m' are disjoint in such a way that the entire volume of the MMS can pass through the free space between them. Therefore, there exists a feasible trajectory ${}^I\mathbf{p}_{MMS}(t)$ for the MMS that connects the initial configuration ${}^I\mathbf{p}_{MMS}(t_0)$ with the desired one ${}^I\mathbf{p}_{MMS_d}$ such as:

$$\mathcal{B}({}^I\mathbf{p}_{MMS}(t), r) \cap \{\mathcal{B}(\mathbf{p}_{\pi_m}, r_{\pi_m}) \cup \partial\mathcal{W}\} = \emptyset, m \in \{1, \dots, \mathcal{M}\}$$

A graphical representation of the feasible trajectory is depicted in Fig. 6.

3) *Occlusion Avoidance:* Let $\mathbf{q}_{arm} \in \mathbb{R}^m$ be the vector of manipulator joint displacements where m is the number of manipulator joints and ${}^B\mathbf{p}_i \in \mathbb{R}^3$ be the position of joint i , $i = 0 \dots m$ with respect to the mobile platform base frame \mathbf{B} as Fig. 7 depicts. The Cartesian coordinates of each i joint can be calculated using the arm forward kinematics:

$${}^B\mathbf{p}_i = \mathcal{F}_i(\mathbf{q}_{arm}) \quad (\text{II.9})$$

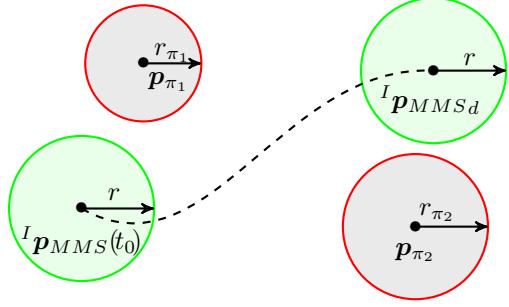


Fig. 6: Graphical representation of a feasible robot trajectory from an initial position ${}^I\mathbf{p}_{MMS}(t_0)$ to a desired one ${}^I\mathbf{p}_{MMS_d}$.

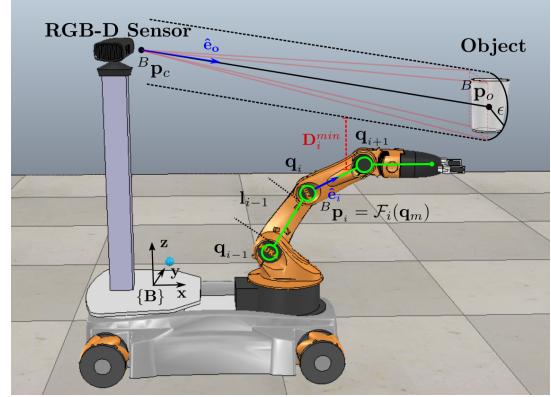


Fig. 7: Geometry of Occlusion Avoidance

where $\mathcal{F}_i(\cdot)$ is the forward kinematics from base frame \mathbf{B} to i -th joint. If link i is modeled by a line segment from joint i with Cartesian coordinates ${}^B\mathbf{p}_i$ to joint $i+1$ with Cartesian coordinates ${}^B\mathbf{p}_{i+1}$, the unit vector of the link with length l_i is computed by:

$$\hat{\mathbf{e}}_i(\mathbf{q}_{arm}) = \frac{{}^B\mathbf{p}_{i+1} - {}^B\mathbf{p}_i}{l_i} \quad (\text{II.10})$$

The line segment equation \bar{S}_i of the i -th manipulator link is defined by

$$\begin{aligned} \bar{S}_i : S_i(\mu_i, \mathbf{q}_{arm}) &= {}^B\mathbf{p}_i + \mu_i \cdot \hat{\mathbf{e}}_i \\ \mu_i \in \mathbb{R} \wedge \mu_i &\in [0, l_i] \end{aligned} \quad (\text{II.11})$$

Considering a line segment \bar{S}_o from robot camera Cartesian coordinates ${}^B\mathbf{p}_c$ to target object position ${}^B\mathbf{p}_o$, the corresponding equations are:

$$\begin{aligned} \bar{S}_o : S_o(\lambda) &= {}^B\mathbf{p}_c + \lambda \cdot \hat{\mathbf{e}}_o \\ \lambda \in \mathbb{R} \wedge \lambda &\in [0, l_o] \end{aligned} \quad (\text{II.12})$$

where $\hat{\mathbf{e}}_o = \frac{{}^B\mathbf{p}_o - {}^B\mathbf{p}_c}{l_o}$ is the unit vector and $l_o = \|{}^B\mathbf{p}_o - {}^B\mathbf{p}_c\|$ the length of the line segment \bar{S}_o . Then, the Minimum Distance D_i^{\min} between each link of the robot and the occlusion line can be calculated. The vector between any two points on the two lines (occlusion and robot link) is:

$$D_i(\mu_i, \mathbf{q}_{arm}, \lambda) = S_i(\mu_i) - S_o(\lambda) = ({}^B\mathbf{p}_i - {}^B\mathbf{p}_c) + \mu_i \cdot \hat{\mathbf{e}}_i - \lambda \cdot \hat{\mathbf{e}}_o \quad (\text{II.13})$$

The shortest among these vectors is perpendicular to both lines and thus:

$$\begin{aligned} (^B\mathbf{p}_i - ^B\mathbf{p}_c) \cdot \hat{\mathbf{e}}_i + \mu_i^{min} - \lambda^{min}(\hat{\mathbf{e}}_i \cdot \hat{\mathbf{e}}_o) &= 0 \\ (^B\mathbf{p}_i - ^B\mathbf{p}_c) \cdot \hat{\mathbf{e}}_o + \mu_i^{min}(\hat{\mathbf{e}}_i \cdot \hat{\mathbf{e}}_o) - \lambda^{min} &= 0 \end{aligned} \quad (\text{II.14})$$

Solving for the line parameters and considering that the line segments are not parallel ($\hat{\mathbf{e}}_i \cdot \hat{\mathbf{e}}_o \neq 1$), we obtain

$$\begin{aligned} \mu_i^{min} &= \frac{(^B\mathbf{p}_i - ^B\mathbf{p}_c) \cdot \hat{\mathbf{e}}_o - [(^B\mathbf{p}_i - ^B\mathbf{p}_c) \cdot \hat{\mathbf{e}}_i](\hat{\mathbf{e}}_i \cdot \hat{\mathbf{e}}_o)}{1 - (\hat{\mathbf{e}}_i \cdot \hat{\mathbf{e}}_o)^2} \\ \lambda^{min} &= \frac{(^B\mathbf{p}_i - ^B\mathbf{p}_c) \cdot \hat{\mathbf{e}}_i - [(^B\mathbf{p}_i - ^B\mathbf{p}_c) \cdot \hat{\mathbf{e}}_o](\hat{\mathbf{e}}_i \cdot \hat{\mathbf{e}}_o)}{1 - (\hat{\mathbf{e}}_i \cdot \hat{\mathbf{e}}_o)^2} \end{aligned} \quad (\text{II.15})$$

However, we suppose the minimum distance between two finite line segments \bar{S}_i and \bar{S}_o which means that $0 < \mu_i^{min} < l_i$ and $0 < \lambda^{min} < l_o$. Therefore, if the value of μ_i^{min} computed by eq. II.15 is greater than l_i , then $\mu_i^{min} = l_i$ and if $\mu_i^{min} < 0$, then $\mu_i^{min} = 0$. Similarly for λ^{min} . In the situation that two line segments are parallel ($\hat{\mathbf{e}}_i \cdot \hat{\mathbf{e}}_o = 1$), the distance between them is constant. The parallel distance can be computed by setting one parameter to zero, e.g. $\lambda^{min} = 0$, and solving for the other (μ_i^{min}) using the equations in II.14.

Having computed the parameters of the line segments as described above, it is trivial to calculate the vector between the proximate points by substituting them in eq. II.13:

$$\mathbf{D}_i(\mu_i^{min}, \mathbf{q}_{arm}, \lambda^{min}) = (^B\mathbf{p}_i - ^B\mathbf{p}_c) + \mu_i^{min} \cdot \hat{\mathbf{e}}_i - \lambda^{min} \cdot \hat{\mathbf{e}}_o \quad (\text{II.16})$$

and consequently, the minimum interline distance is

$$D_i^{min} = \|\mathbf{D}_i(\mu_i^{min}, \mathbf{q}_{arm}, \lambda^{min})\| \quad (\text{II.17})$$

Hence, in order to avoid a possible occlusion between the camera (RGB-D sensor) and the target object, the above inequalities must be satisfied:

$$D_i^{min} - \epsilon > 0, \forall i \in [1, m] \wedge i \in \mathbb{N} \quad (\text{II.18})$$

where $\epsilon \in \mathbb{R}$ is the radius of the augmented cylinder in which the 4-sided pyramid calculated by the object dimensions is inscribed as shown in Fig. 7.

4) Field of View Constraints: Let us define vector ${}^o\mathbf{p}_c = [{}^o\mathbf{x}_c, {}^o\mathbf{y}_c, {}^o\mathbf{z}_c, {}^o\psi_c]$ describing the pose of the camera (RGB-D sensor) with respect to the object frame and a, b the field of view angles in horizontal and vertical axes. If y_T, z_T are the dimensions of the object bounding box along the horizontal and vertical axes, the following inequalities must hold in order to always maintain the object inside the camera field of view:

$$- {}^o\mathbf{y}_c + {}^o\mathbf{x}_c \tan\left({}^o\psi_c - \frac{a}{2}\right) - \frac{y_T}{2} \geq 0 \quad (\text{II.19a})$$

$${}^o\mathbf{y}_c - {}^o\mathbf{x}_c \tan\left({}^o\psi_c + \frac{a}{2}\right) - \frac{y_T}{2} \geq 0 \quad (\text{II.19b})$$

$$- {}^o\mathbf{z}_c - {}^o\mathbf{x}_c \tan\left(\frac{b}{2}\right) - \frac{z_T}{2} \geq 0 \quad (\text{II.19c})$$

$${}^o\mathbf{z}_c - {}^o\mathbf{x}_c \tan\left(\frac{b}{2}\right) - \frac{z_T}{2} \geq 0 \quad (\text{II.19d})$$

The above inequalities are actually a 3D extension of the 2D visibility constraints appear in [23], thus the analytical derivation is omitted due to space limitations.

5) State Constraints by Design: The arm joints have hardware limits which reflect to the feasible positions of the joints. These limits are captured by the following inequalities:

$$\mathbf{q}_{arm_i} \leq |\bar{q}_{arm_i}|, i = 0 \dots 4 \quad (\text{II.20})$$

where by (\cdot) we denote the upper bound of the corresponding variable. The omni-directional wheels of the mobile platform can rotate freely, hence no position constraint is imposed.

6) Input Constraints by Design: As it is shown in Eq. II.1, the inputs of the system are the joint and wheel velocities at time step k as described in vector \mathbf{u}_k . As it is expected, the inputs have upper and lower bounds by design, and if these bounds are exceeded the inputs are saturated. Thus, we define the input constraint set U_m as follows:

$$\mathbf{u}_k = [\mathbf{u}_{arm_k}, \mathbf{u}_{base_k}]^T \in U_m \subseteq \mathbb{R}^9 \quad (\text{II.21})$$

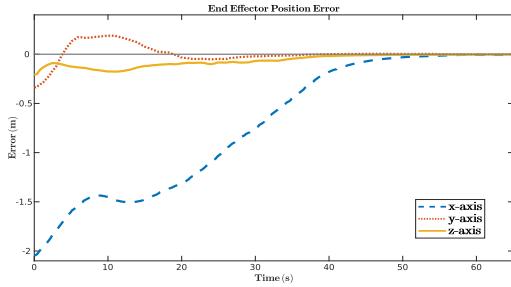
These constraints are of the form $|\omega_{q_{im}}| \leq \bar{\omega}_{q_{im}}, i = 0 \dots 4$ for the arm and $|\omega_{FL_m}| \leq \bar{\omega}_{FL_m}, |\omega_{FR_m}| \leq \bar{\omega}_{FR_m}, |\omega_{BL_m}| \leq \bar{\omega}_{BL_m}, |\omega_{BR_m}| \leq \bar{\omega}_{BR_m}$ for the mobile platform wheels.

III. EXPERIMENTAL RESULTS

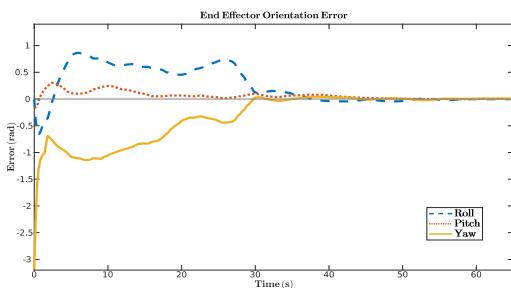
In order to demonstrate the efficacy of the proposed strategy, two experimental reaching and grasping procedures with different objects were carried out using a KUKA Youbot MMS, equipped with a Kinect RGB-D sensor as shown in Fig.1. Both experiments can be seen in the accompanying video, while, due to space limitations, the experimental results only for the first case study are discussed in this section. In the experimental scenarios, a human located inside the workspace, is holding an object. The Youbot starts from an initial pose and it is guided by the proposed scheme in order to reach the object and pick it up from the human.

In the experiments, the radius of the obstacles was $r_{\pi_m} = 0.15m$, while the dimension of the workspace was $Length \times Width = 4m \times 2.8m$. The input bounds for the arm joint velocity commands are $|\bar{\omega}_{q_{im}}| = 0.4rad/s$, while for the wheel velocity commands of the omni-directional platform are $|\bar{\omega}_{FL_m}| = |\bar{\omega}_{FR_m}| = |\bar{\omega}_{BL_m}| = |\bar{\omega}_{BR_m}| = 1.57rad/s$.

The position and orientation errors of the MMS end-effector are depicted in Fig. 8. As it shown by the system responses, the end-effector successfully reaches the pose of the target object. Moreover, Fig. 9 shows the control inputs calculated by the NMPC scheme, which remain within the specified bounds (depicted by red dashed lines) at all times. The responses of the arm joint states are omitted since it can be observed from the accompanying video that the system never reached its joint limits. The trajectory executed by the robot can be seen in Fig. 10. As it can be observed, the bounding circle (green disc) of the robot intersects neither the two static obstacles (red discs) nor the workspace's boundaries (black bold lines). The radius of the green circle changes according to the configuration of the arm which is depicted in 5 sample trajectory poses. In Fig. 11 the occlusion constraints $D_i^{min} - \epsilon$ are depicted, as Distances 1, 2, 3, where it is shown that the NMPC scheme successfully



(a) End Effector Position Error



(b) End Effector Orientation Error

Fig. 8: End Effector Position and Orientation Errors during Reaching and Grasping

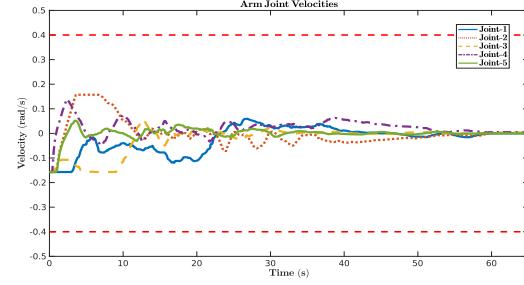
manages to keep them within the specified limits (always positive). Finally, the Field of View constraints are shown in Fig. 12 where each line represents the inequalities presented in Eq. II.19. As it can be observed, the NMPC successfully respected the specified FoV restrictions.

IV. CONCLUSION

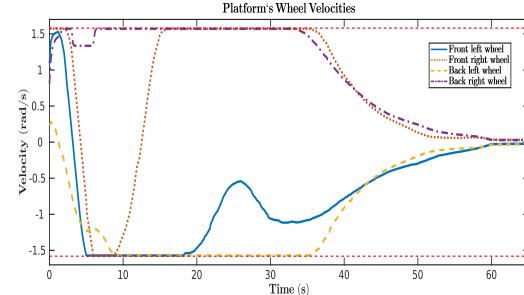
In this paper we presented a vision-based object grasping and motion control architecture for a mobile manipulator system. The optimal grasping areas of the object were estimated via an appropriate algorithm and the on-line measurements of an on-board RGB-D system. The overall (reaching and grasping) motion of the MMS was handled via a Nonlinear Model Predictive Control scheme, capable to handle complex and multiple input and state constraints, such as occlusion and obstacle avoidance, workspace boundaries and field of view restrictions. The performance of the proposed strategy was experimentally verified using a KUKA Youbot, operating inside a constrained workspace with obstacles, in order to acquire an object from the hands of a human.

REFERENCES

- [1] T. Stock and G. Seliger, "Opportunities of sustainable manufacturing in industry 4.0," *Procedia Cirp*, vol. 40, pp. 536–541, 2016.
- [2] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 1, pp. 348–353, IEEE, 2000.
- [3] A. Sahbani, S. El-Khoury, and P. Bidaud, "An overview of 3d object grasp synthesis algorithms," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 326–336, 2012.
- [4] A. Bicchi, "On the closure properties of robotic grasping," *The International Journal of Robotics Research*, vol. 14, no. 4, pp. 319–334, 1995.



(a) Arm Joints Velocity Commands



(b) Platform Wheels Velocity Commands

Fig. 9: Control Inputs

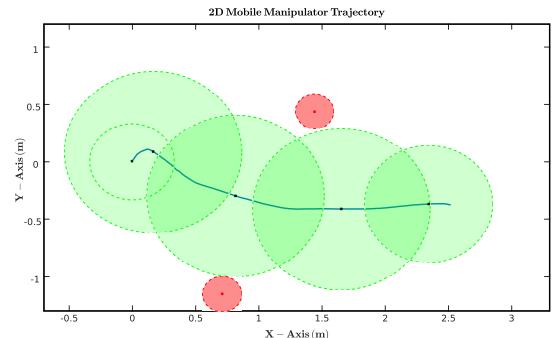


Fig. 10: 2D Mobile Manipulator Trajectory with Obstacle Avoidance and Workspace Boundaries

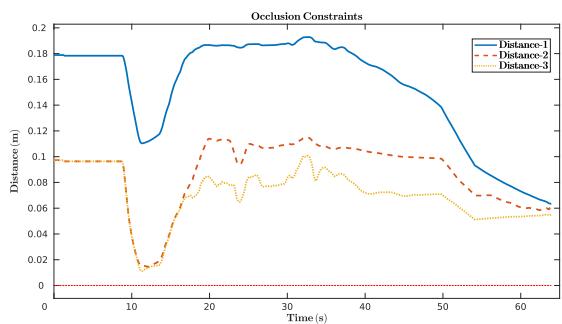


Fig. 11: Occlusion Constraints

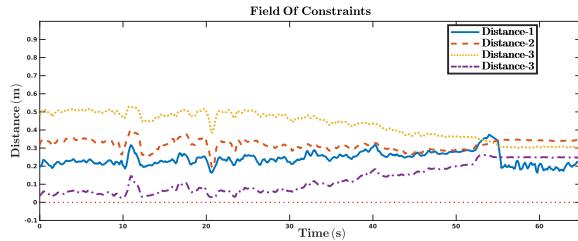


Fig. 12: Field of View Constraints

- [5] L. Han, J. C. Trinkle, and Z. X. Li, "Grasp analysis as linear matrix inequality problems," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, pp. 663–674, 2000.
- [6] X. Zhu and J. Wang, "Synthesis of force-closure grasps on 3-d objects based on the q distance," *IEEE Transactions on robotics and Automation*, vol. 19, no. 4, pp. 669–679, 2003.
- [7] B. Mirtich and J. Canny, "Easily computable optimum grasps in 2-d and 3-d," in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pp. 739–747, IEEE, 1994.
- [8] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 2, pp. 1824–1829, IEEE, 2003.
- [9] K. Huebner and D. Kragic, "Selection of robot pre-grasps using box-based shape approximation," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 1765–1770, IEEE, 2008.
- [10] G. M. Bone, A. Lambert, and M. Edwards, "Automated modeling and robotic grasping of unknown three-dimensional objects," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 292–298, IEEE, 2008.
- [11] B. Wang, L. Jiang, J. Li, and H. Cai, "Grasping unknown objects based on 3d model reconstruction," in *Advanced Intelligent Mechatronics. Proceedings, 2005 IEEE/ASME International Conference on*, pp. 461–466, IEEE, 2005.
- [12] C. Goldfeder, M. Ciocarlie, J. Peretzman, H. Dang, and P. K. Allen, "Data-driven grasping with partial sensor data," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 1278–1283, IEEE, 2009.
- [13] K. Hsiao, S. Chitta, M. Ciocarlie, and E. G. Jones, "Contact-reactive grasping of objects with partial shape information," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 1228–1235, IEEE, 2010.
- [14] B. Calli, M. Wisse, and P. Jonker, "Grasping of unknown objects via curvature maximization using active vision," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 995–1001, IEEE, 2011.
- [15] Q. Lei and M. Wisse, "Fast grasping of unknown objects using force balance optimization," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pp. 2454–2460, IEEE, 2014.
- [16] M. Prats, P. Martinet, A. P. Del Pobil, and S. Lee, "Vision force control in task-oriented grasping and manipulation," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 1320–1325, IEEE, 2007.
- [17] N. Vahrenkamp, S. Wieland, P. Azad, D. Gonzalez, T. Asfour, and R. Dillmann, "Visual servoing for humanoid grasping and manipulation tasks," in *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on*, pp. 406–412, IEEE, 2008.
- [18] Y. Wang, H. Lang, and C. W. De Silva, "A hybrid visual servo controller for robust grasping by wheeled mobile robots," *IEEE/ASME transactions on Mechatronics*, vol. 15, no. 5, pp. 757–769, 2010.
- [19] N. Vahrenkamp, M. Do, T. Asfour, and R. Dillmann, "Integrated grasp and motion planning," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 2883–2888, IEEE, 2010.
- [20] N. Vahrenkamp, D. Berenson, T. Asfour, J. Kuffner, and R. Dillmann, "Humanoid motion planning for dual-arm manipulation and re-grasping tasks," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 2464–2470, IEEE, 2009.
- [21] J. Fontanals, B.-A. Dang-Vu, O. Porges, J. Rosell, and M. A. Roa, "Integrated grasp and motion planning using independent contact

- regions," in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pp. 887–893, IEEE, 2014.
- [22] F. Allgower, R. Findeisen, Z. K. Nagy, et al., "Nonlinear model predictive control: From theory to application," *Journal-Chinese Institute Of Chemical Engineers*, vol. 35, no. 3, pp. 299–316, 2004.
- [23] S. Maniatopoulos, D. Panagou, and K. J. Kyriakopoulos, "Model predictive control for the navigation of a nonholonomic vehicle with field-of-view constraints," in *American Control Conference (ACC), 2013*, pp. 3967–3972, IEEE, 2013.
- [24] T. P. Nascimento, A. P. Moreira, and A. G. S. Conceição, "Multi-robot nonlinear model predictive formation control: Moving target and target absence," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1502–1515, 2013.
- [25] T. P. Nascimento, A. G. Conceição, and A. P. Moreira, "Multi-robot nonlinear model predictive formation control: the obstacle avoidance problem," *Robotica*, vol. 34, no. 3, pp. 549–567, 2016.
- [26] A. Brooks, T. Kaupp, and A. Makarenko, "Randomised mpc-based motion-planning for mobile robot obstacle avoidance," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 3962–3967, IEEE, 2009.
- [27] S. Ide, T. Takubo, K. Ohara, Y. Mae, and T. Arai, "Real-time trajectory planning for mobile manipulator using model predictive control with constraints," in *Ubiquitous Robots and Ambient Intelligence (URAI), 2011 8th International Conference on*, pp. 244–249, IEEE, 2011.
- [28] S. Heshmati-Alamdari, G. C. Karras, A. Eqtami, and K. J. Kyriakopoulos, "A robust self triggered image based visual servoing model predictive control scheme for small autonomous robots," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 5492–5497, IEEE, 2015.
- [29] G. B. Avanzini, A. M. Zanchettin, and P. Rocco, "Constraint-based model predictive control for holonomic mobile manipulators," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 1473–1479, IEEE, 2015.
- [30] G. B. Avanzini, A. M. Zanchettin, and P. Rocco, "Constrained model predictive control for mobile robotic manipulators," *Robotica*, vol. 36, no. 1, pp. 19–38, 2018.
- [31] R. Bischoff, U. Huggenberger, and E. Prassler, "Kuka youbot-a mobile manipulator for research and education," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 1–4, IEEE, 2011.
- [32] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, (Shanghai, China), May 9–13 2011.
- [33] D. Koditschek and E. Rimon, "Robot navigation functions on manifolds with boundary," *Advances in Applied Mathematics*, vol. 11, no. 4, pp. 412–442, 1990.