Kinematically-Decoupled Impedance Control for Fast Object Visual Servoing and Grasping on Quadruped Manipulators

Riccardo Parosi[1,2,*], Mattia Risiglione[1,2,*], Darwin G. Caldwell[1], Claudio Semini[1], Victor Barasuol[1]

***Abstract*— We propose a control pipeline for SAG (Searching, Approaching, and Grasping) of objects, based on a decoupled arm kinematic chain and impedance control, which integrates image-based visual servoing (IBVS). The kinematic decoupling allows for fast end-effector motions and recovery that leads to robust visual servoing. The whole approach and pipeline can be generalized for any mobile platform (wheeled or tracked vehicles), but is most suitable for dynamically moving quadruped manipulators thanks to their reactivity against disturbances. The compliance of the impedance controller makes the robot safer for interactions with humans and the environment. We demonstrate the performance and robustness of the proposed approach with various experiments on our 140 kg HyQReal quadruped robot equipped with a 7-DoF manipulator arm. The experiments consider dynamic locomotion, tracking under external disturbances, and fast motions of the target object.**

I. INTRODUCTION

To increase the number of tasks mobile manipulation systems can execute in unstructured environments, mobility and vision are two key aspects. Concerning the former, legs allow to select footholds and control the wrench acting on the floating base, orienting and moving it to increase the manipulation workspace when necessary [1][2][3]. Until now, vision for legged platforms such as quadrupeds and bipeds has been mainly used for locomotion, e.g. to correct nominal footholds [4][5] or for navigation, e.g. visual odometry [6]. Many recent works combined the advantages of a mobile legged platform with a robotic manipulator to perform manipulation tasks: opening a door [7][8][9], pulling a rope with a basket [10], turning a valve [10][9], grasp a target object [3] and put it into a trash bin [2]. Although successful executions, most of these works provide to the robot direct knowledge of its surroundings and do not close the loop with vision for manipulation. The use of visual feedback from an onboard camera placed at the arm's end-effector, also known as Eye-In-Hand camera [11], has been shown to guarantee a more accurate positioning of the arm's end-effector for manipulation, robustness to calibration uncertainties, and reactivity to environmental changes [12]. From 2D images position-based visual servoing (PBVS) retrieves the pose of the target, while image-based visual servoing (IBVS) works with feature representation directly in the image domain. The advantages of IBVS over PBVS are the following: *(i)* it does not require any 3D model; *(ii)* it is more robust with respect to uncertainties of robot and camera model, in particular to calibration errors [13]; *(iii)* it is easier to formulate feature-based motion strategies aimed at keeping the target always in the camera field of view. Over the past, IBVS control schemes have been proposed for control of underactuated systems like drones [14], non-holonomic mobile robots [15][16] and floating-base space manipulators [17]. Despite the advantages, feature depth is unknown in IBVS and it must be estimated or measured (e.g directly from a RGB-D camera) in order to calculate the interaction matrix. To alleviate some of the problems induced by both methods, hybrid schemes [18] use 3D information, usually obtained by epipolar geometry, to control some degrees of freedom (DoF) of the camera, while the remaining ones are controlled through IBVS.

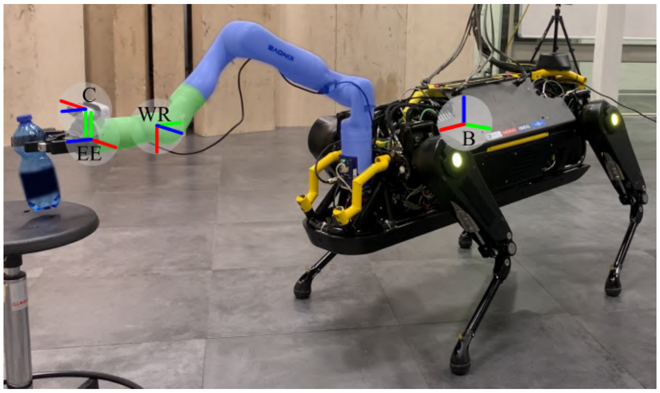

Fig. 1: IIT's 140 kg HyQReal robot equipped with a 7-DoF manipulator arm (Kinova Gen3 [19]) grasping a bottle. The translucent blue and green areas represent, respectively, the Shelbow and the Wrist kinematic groups (proposed and introduced in Sec. II-D to allow for fast visual servoing). The frames depicted on the robot associate, respectively, the red/green/blue colored axes to their $x/y/z$ coordinate axes. The frames are named as B: robot's base, WR: arm's wrist, C: camera, EE: arm's end-effector.

To have robots autonomously performing tasks that involve environment interaction, they need to have the ability to grasp objects and/or tools. Depending on the application, e.g. logistics, domestic support, construction, etc., the difference will be in what and how to manipulate. The problem of grasping a generic object is commonly split into three phases: *(a)* Search, i.e. scanning the robot's surroundings, while recognizing and distinguishing objects of interest; *(b)* Approach, i.e. plan and execute a trajectory to get close to the object; *(c)* Grasp, i.e. move the gripper to the target pose in order to

[1] Dynamic Legged Systems Lab, Istituto Italiano di Tecnologia (IIT), Genova, Italy, {name.surname}@iit.it.

[2] Dipartimento di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi (DIBRIS), Università di Genova, Genova, Italy, {name.surname}@edu.unige.it.

[*] Equal contribution.

grasp the object and close the gripper. Previous frameworks have given contributions to sub-problems related to the SAG (S: Search, A: Approach, G: Grasp) of an object. These contributions include ensuring stop-free exploration while searching [20], detecting objects in challenging environments [21], collision-free object search in cluttered scenarios [22] and locating moving targets [23]. Additionally, a few-shot object detection is presented in [24] to learn detection-based tasks for new objects. In [24], a complete SAG pipeline is executed, and the robot learns to grasp scattered objects. Although the robot is able to execute the SAG pipeline, the target is not being shown to move.

Mobile platforms are commonly used for SAG problems, due to the increased mobility and reachability. These systems have more degrees of freedom than the ones necessary from the visual task, normally defined as keeping the object in the camera field of view. Hence, the camera position can be changed in many different ways, using redundancy to optimize a configuration-dependent criteria, such as distance from obstacles, singularities and manipulability indices, or dynamic cost functions [25]. In [26] the authors formulate visual tracking tasks for a humanoid robot in a hierarchical Quadratic Programming (QP) problem, relating the motion of the visual features to the joint accelerations and solving for the latter ones. Other constraints relating feature visibility and mobility, such as dynamic consistency and center of mass control, are set as tasks/constraints in the same QP problem at different priorities. Differently, in [3], vision is not directly used in the motion control of the robot, but only in the motion generation. A 3D target position is retrieved with the camera parameters and fed to a trajectory generator for the quadruped robot Spot with a Kinova robotic arm. The whole-body robot posture is optimized through inverse kinematics to reach the desired position and the motion plan is capable to leverage on the utility of legs to increase the arm reachability and avoid collisions with a table. The platform results to have 80% accuracy in grasping a ball placed at different initial condition, but during fails the final grasping position results far from the object. The authors relate the problem either to inaccurate ball position estimation or to possible discrepancy between the real and planned initial robot condition. Additionally, the manipulator is velocity-controlled, as commonly done for visual servoing, making it stiff and not suitable for contexts where human and/or unknown obstacles are in the robot proximity.

Similarly to [3], we tackle the problem of mobile manipulation with a quadruped manipulator and vision in the loop. More specifically, we propose a control pipeline for the SAG of a target object, using an Eye-In-Hand RGB-D camera mounted at the arm's end-effector. In this work, we propose a control approach that uses the joints of the wrist, made up commonly by two or three small and compact actuators, connected through lower inertia links, for the visual task. The rest of the robotic manipulator kinematic chain and the floating base is used to move the robot and position the arm's wrist for Searching, Approaching and Grasping. Based on visual information, we generate a sequence of positions and velocities, converted later to torques for the execution of the SAG sequence. In contrast to the previous mentioned papers, all the actuators are controlled through torque control and an impedance control strategy is integrated to render impedances, through legs, on the quadruped's base and on the arm, through the arm actuators. The impedance rendered at the arm can be chosen to mitigate tracking errors, external disturbance attenuation induced by the floating base and/or by an external source (e.g. manipulation, human interaction). The proposed approach is validated with a set of experiments on the 140 kg hydraulic, torque-controlled quadruped HyQReal, with a torque-controlled 7-DoF Kinova Gen3 arm as robotic manipulator. In summary, we highlight the following contributions

- A kinematically-decoupled control approach that integrates an IBVS scheme and impedance control. The control approach maps the visual task only on the wrist, exploiting low-inertia links for fast motion and reactiveness, and the rest of the kinematic chain for less demanding arm positioning. To the best of the authors' knowledge, this is the first time visual servoing is integrated on a fully torque-controlled quadruped manipulator.
- A sequence of behavior and control signals for the Search, Approach and Grasp with a torque-controlled quadruped manipulator.
- Experimental demonstration and assessment of the approach on a quadruped manipulator, showing active compliance and visual servoing in presence of external disturbances, the ability to execute the SAG pipeline, and to track a fast-moving object.

The paper is organized as follows: Section II presents the dynamic model and the motion control of the robot. Section III describes how the whole body motion of the platform is generated. Section IV describes the experiments and discusses the results. Section V closes the paper with conclusions and future work.

II. Motion Control

A. Robot Model

The full rigid-body dynamics of a legged manipulator can be described by the set of dynamic equations in (1), where $\boldsymbol{M}$ is the inertia matrix, $\dot{\boldsymbol{u}}$ the stacked vector of generalized accelerations, $\boldsymbol{h}$ comprises the gravity, Coriolis and Centrifugal terms, $\boldsymbol{\tau}$ the actuation torques. The subscripts b, l, a, and e stand for base, legs, arm and arm's end-effector, respectively. The stacked vector of generalized accelerations $\dot{\boldsymbol{u}} = [\ddot{\boldsymbol{q}}_b^T, \ddot{\boldsymbol{q}}_l^T, \ddot{\boldsymbol{q}}_a^T]^T \in \mathbb{R}^{6+n_l+n_a}$ denotes the linear and angular accelerations of the base $\ddot{\boldsymbol{q}}_b = [\ddot{\boldsymbol{x}}_b^T, \dot{\boldsymbol{w}}_b^T]^T \in \mathbb{R}^6$ and the rest of the limb joint accelerations. $\boldsymbol{F}_g \in \mathbb{R}^{3n_c}$ are the ground reaction forces, where n_c denotes the number of contact feet; $\boldsymbol{F}_e \in \mathbb{R}^3$ denotes the external force acting on the arm's end-effector. $\boldsymbol{F}_g$ and $\boldsymbol{F}_e$ are mapped respectively to the base through the contact Jacobians $\boldsymbol{J}_{st}^T$ and $\boldsymbol{J}_e^T$. $\boldsymbol{J}_{e,a}^T \in \mathbb{R}^{6\times n_a}$ is the Jacobian matrix from base to end-effector.

$$\underbrace{\begin{bmatrix} \boldsymbol{M}_b & \boldsymbol{M}_{bl} & \boldsymbol{M}_{ba} \\ \boldsymbol{M}_{lb} & \boldsymbol{M}_l & \boldsymbol{M}_{la} \\ \boldsymbol{M}_{ab} & \boldsymbol{M}_{al} & \boldsymbol{M}_a \end{bmatrix}}_{\boldsymbol{M}} \underbrace{\begin{bmatrix} \ddot{\boldsymbol{q}}_b \\ \ddot{\boldsymbol{q}}_l \\ \ddot{\boldsymbol{q}}_a \end{bmatrix}}_{\dot{\boldsymbol{u}}} + \underbrace{\begin{bmatrix} \boldsymbol{h}_b \\ \boldsymbol{h}_l \\ \boldsymbol{h}_a \end{bmatrix}}_{\boldsymbol{h}} = \underbrace{\begin{bmatrix} \boldsymbol{J}_{st,b}^T \\ \boldsymbol{J}_{st,l}^T \\ \boldsymbol{0}_{a\times 3n_c} \end{bmatrix}}_{\boldsymbol{J}_{st}^T} \boldsymbol{F}_g + \\ + \underbrace{\begin{bmatrix} \boldsymbol{J}_{e,b}^T \\ \boldsymbol{0}_{l\times 3} \\ \boldsymbol{J}_{e,a}^T \end{bmatrix}}_{\boldsymbol{J}_e^T} \boldsymbol{F}_e + \underbrace{\begin{bmatrix} \boldsymbol{0}_{6\times 1} \\ \boldsymbol{\tau}_l \\ \boldsymbol{\tau}_a \end{bmatrix}}_{\boldsymbol{\tau}} \tag{1}$$

B. Base Controller

The base of a legged manipulator is commonly considered as the *Trunk*, in which various limbs are connected to. In this work, we use the Trunk Controller proposed in [27], that imposes a desired wrench on the base, $\boldsymbol{W}_b^d \in \mathbb{R}^6$, computed based on position and rotation errors, and it gets mapped to ground reaction forces, $\boldsymbol{F}_g$, considering friction, unilaterality constraints, and force limits of each stance leg (2) as

$$\begin{aligned} \min_{\boldsymbol{F}_g} \quad & \left\| \begin{bmatrix} \boldsymbol{I} & \cdots & \boldsymbol{I} \\ [\boldsymbol{p}_{bc_1}]_\times & \cdots & [\boldsymbol{p}_{bc_n}]_\times \end{bmatrix} \boldsymbol{F}_g - \boldsymbol{W}_b^d \right\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{F}_g\|_{\boldsymbol{R}}^2 \\ \text{s.t.} \quad & \underline{\boldsymbol{d}} \leq \boldsymbol{C}\boldsymbol{F}_g \leq \overline{\boldsymbol{d}} \end{aligned} \tag{2}$$

where $\boldsymbol{p}_{bc} \in \mathbb{R}^3$ is the relative distance of the foot in contact with respect to the base, $[\boldsymbol{p}_{bc}]_\times$ denotes the skew-symmetric matrix of vector $\boldsymbol{p}_{bc}$, $\boldsymbol{C}$ is the inequality constraint matrix, $\underline{\boldsymbol{d}}$ and $\overline{\boldsymbol{d}}$ are lower/upper bound respectively that ensure that the ground reaction forces lie inside the friction cones and the normal components of the forces are bounded by some user-defined limits. The first term in the cost represents the tracking error between the actual and the desired wrench, $\boldsymbol{W}_b^d = [\boldsymbol{F}_b^{d^T}, \boldsymbol{T}_b^{d^T}]^T$, defined as

$$\boldsymbol{F}_b^d = \boldsymbol{K}_b(\boldsymbol{x}_b^d - \boldsymbol{x}_b) + \boldsymbol{D}_b(\dot{\boldsymbol{x}}_b^d - \dot{\boldsymbol{x}}_b) \tag{3}$$

$$\boldsymbol{T}_b^d = \boldsymbol{D}_r(\boldsymbol{w}_b^d - \boldsymbol{w}_b) + \boldsymbol{K}_r\boldsymbol{e}_r \tag{4}$$

where $\boldsymbol{F}_b^d$ and $\boldsymbol{T}_b^d$ are respectively the desired force and moment for the base, i.e. $\boldsymbol{W}_b^d = [\boldsymbol{F}_b^{d^T}, \boldsymbol{T}_b^{d^T}]^T$. We define as $\boldsymbol{e}_r$ the rotational error, $\boldsymbol{D}_r \in \mathbb{R}^{3\times 3}$ and $\boldsymbol{K}_r \in \mathbb{R}^{3\times 3}$ diagonal gain matrices for the derivative and proportional term, respectively. For further implementation details on the friction constraints, we refer to [27]. To guarantee a better motion tracking for the base, we compensate the dynamic coupling effects induced by the legs and arm on the base, using the dynamic model in (1).

C. Leg Controller

We compute the torques for each leg by superimposing two control actions: a feedforward term, $\boldsymbol{\tau}_{ff}$, obtained mapping $\boldsymbol{F}_g$ from (2) to torques through the stance Jacobian, i.e. $\boldsymbol{\tau}_{ff} = -\boldsymbol{J}_{st,l}^T\boldsymbol{F}_g$; a feedback term in the form of a PD controller. The second term is needed to track swing leg trajectories. Hence, the total torque for each leg is computed as

$$\boldsymbol{\tau}_l = \boldsymbol{\tau}_{ff} + PD(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \boldsymbol{q}_l^d, \dot{\boldsymbol{q}}_l^d) \tag{5}$$

For the trajectory generation of the legs we exploit the structure of the Reactive Control Framework [28].

D. Arm Controller

The kinematic chain of common manipulators can be split into two groups: what we name as *Shelbow* group, comprising shoulder and elbow joints highlighted with blue in Fig.1, and the wrist, consisting of two or three compact joints highlighted with green in Fig.1. Commercial arms, like *Kinova Gen3* [29] and *Franka-Emika* [30], have this type of structure, where the last three joints at the wrist have smaller maximum torque peak and are connected through smaller links. The idea of this work is to use the former group to establish an impedance connection with a desired position for the wrist. Throughout the manuscript, we refer to wrist position as the origin of the wrist frame (denoted as WR in Fig. 1). For searching, approaching or grasping an object, the desired wrist position is normally defined by the camera and target position. Instead, the wrist is used for tracking an end-effector's trajectory when a target is not in the view of the camera, and to keep the target in the view of the camera once found with the visual feedback received by the Eye-In-Hand camera. The Cartesian impedance control imposed at the wrist position is applied using the Shelbow's joints and impedances are rendered in the Horizontal Frame (a reference frame whose xy plane is always horizontal and its x axis always aligned to the x axis of the robot's base) [28], to reduce cross-coupling effects with the base, as

$$\begin{aligned} \boldsymbol{\tau}_{shelbow} = \boldsymbol{J}_{sh}^T\big[&\boldsymbol{K}_p^{sh}(\boldsymbol{x}_{wr}^d - \boldsymbol{x}_{wr}) + \boldsymbol{K}_d^{sh}(\dot{\boldsymbol{x}}_{wr}^d - \dot{\boldsymbol{x}}_{wr})\big] + \\ &+ \boldsymbol{h}_{sh} \end{aligned} \tag{6}$$

where $\boldsymbol{h}_{sh}$ is the gravity, Coriolis and Centrifugal torques, $\boldsymbol{J}_{sh} \in \mathbb{R}^{6\times 4}$ is the Shelbow Jacobian matrix obtained extracting the first four columns from $\boldsymbol{J}_{e,a}$, $\boldsymbol{K}_p^{sh} \in \mathbb{R}^{3\times 3}$ and $\boldsymbol{K}_d^{sh} \in \mathbb{R}^{3\times 3}$ are virtual springs and dampers gains. Instead, $\boldsymbol{x}_{wr}^d \in \mathbb{R}^3$ and $\boldsymbol{x}_{wr} \in \mathbb{R}^3$ are the desired and current Cartesian positions of the wrist, while $\dot{\boldsymbol{x}}_{wr}^d \in \mathbb{R}^3$ and $\dot{\boldsymbol{x}}_{wr} \in \mathbb{R}^3$ are the desired and current Cartesian linear velocities of the wrist. Both current and desired positions, as well as velocities, of the wrist are expressed in the Horizontal Frame. For the wrist, the motion control law is generated according to the given reference. During search of the object, the reference is an end-effector trajectory which is tracked generating the wrist torques as

$$\boldsymbol{\tau}_{wr} = \boldsymbol{J}_{wr}^T\big[\boldsymbol{K}_{pc}^{wr}(\boldsymbol{x}_e^d - \boldsymbol{x}_e) + \boldsymbol{K}_{dc}^{wr}(\dot{\boldsymbol{x}}_e^d - \dot{\boldsymbol{x}}_e)\big] + \boldsymbol{h}_{wr} \tag{7}$$

where $\boldsymbol{J}_{wr} \in \mathbb{R}^{6\times 3}$ is the wrist Jacobian matrix obtained extracting the last three columns from $\boldsymbol{J}_{e,a}$, which dependency on the Shelbow joints is omitted. Instead $\boldsymbol{K}_{pc}^{wr} \in \mathbb{R}^{3\times 3}$ and $\boldsymbol{K}_{dc}^{wr} \in \mathbb{R}^{3\times 3}$ are virtual springs and dampers gains. The terms $\boldsymbol{x}_e^d \in \mathbb{R}^3$ and $\boldsymbol{x}_e \in \mathbb{R}^3$ are the desired and current Cartesian positions of the end-effector, while $\dot{\boldsymbol{x}}_e^d \in \mathbb{R}^3$ and $\dot{\boldsymbol{x}}_e \in \mathbb{R}^3$ are the desired and current Cartesian linear velocities of the end-effector. Both current and desired positions, as well as velocities, of the end-effector are expressed in the Horizontal Frame. When vision is activated, e.g. when an object is in the field of view of the camera, then joints' velocities for the wrist are retrieved, and used to obtain setpoints for joints' positions by integration. These joints' position and velocities are tracked to generate the torques for the wrist as

$$\boldsymbol{\tau}_{wr} = \boldsymbol{K}_{pj}^{wr}(\boldsymbol{q}_{wr}^{d} - \boldsymbol{q}_{wr}) + \boldsymbol{K}_{dj}^{wr}(\dot{\boldsymbol{q}}_{wr}^{d} - \dot{\boldsymbol{q}}_{wr})\big] + \boldsymbol{h}_{wr} \quad (8)$$

where $\boldsymbol{K}_{pj}^{wr} \in \mathbb{R}^{3\times3}$ and $\boldsymbol{K}_{dj}^{wr} \in \mathbb{R}^{3\times3}$ are virtual springs and dampers gains. Instead $\boldsymbol{q}_{wr}^{d} \in \mathbb{R}^{3}$ and $\boldsymbol{q}_{wr} \in \mathbb{R}^{3}$ are the desired and current wrist joints positions, while $\dot{\boldsymbol{q}}_{wr}^{d} \in \mathbb{R}^{3}$ and $\dot{\boldsymbol{q}}_{wr} \in \mathbb{R}^{3}$ are the desired and current wrist joints velocities.

III. Motion Generation

In this section, we describe the three main phases of our proposed method that lead to the grasping of an object: Search, Approach and Grasp.

A. Search:

To guide the camera along the search phase, we use an heuristic trajectory that avoids robot singular configurations and self-collisions. The trajectory paths for the wrist position and end-effector are illustrated in Fig. 2. First, the arm is brought to a home configuration where the links are kept away from the base. Then a circular motion, centered around the arm's base, is tracked by the wrist position using (6) (red path in Fig. 2). Along this first phase, singularities are avoided by keeping the radius of this circular trajectory lower than the wrist maximum allowable distance from the arm's base. To avoid self-collisions and image occlusions, the scanning behind the quadruped is done in two steps: first from the left and later from the right of the robot trunk. The two points that define the limits of the wrist position motion path are indicated in Fig. 2 as A and B. Once the wrist is in one of these two positions, two circular trajectories centered around it are used to search backward with the arm's end-effector, defining $\boldsymbol{x}_{e}^{d}$ and $\dot{\boldsymbol{x}}_{e}^{d}$. Hence the wrist is rotated, keeping the roll and the pitch of the end-effector fixed. If the second end-effector backward scan is completed and no object is detected, the arm returns to the home configuration and the robot is commanded to rotate around itself by 180 degrees and restart the wrist position and end-effector search trajectories. We highlight that the only reason for which the area behind the robot has not been assigned to the arm is to avoid occlusions with trunk and legs.

B. Approach:

Once the object has been detected, the robot has to align itself to approach it. The wrist joints are controlled using the joint impedance controller (8) and for the Shelbow joints the Cartesian impedance controller (7). For the alignment, first the wrist is positioned at the intersection between its circular search trajectory (solid red line in Fig. 2) and the line segment connecting the object and the origin of the robot base frame. If the intersection is located on the dashed part of the red circle, the wrist position is set at point A or B (depending on the search side). Successively, the base is aligned to the object by commanding a heading velocity until the longitudinal axis of the robot aligns with the direction of the object.

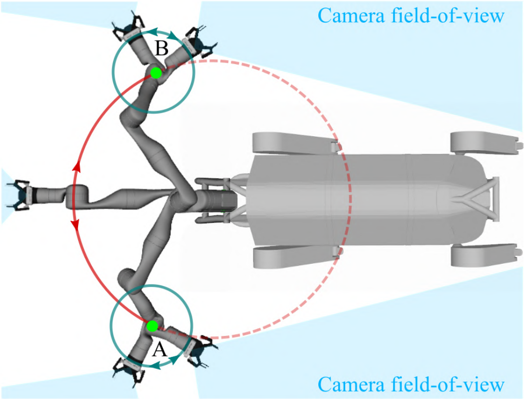

Fig. 2: Illustration of HyQReal and its arm at various postures along the trajectory paths executed during the object Search phase. The light-green dots, A and B, represent the wrist limit positions on its circular path around the arm's base. The solid red semi-circle and the green circles represent, respectively, the circular searching motion that can be executed by the wrist position and the end-effector (the dashed red path is not allowed to avoid self-collisions and camera occlusions).

In order to keep the object in the camera's field of view during the whole alignment phase, the references for the controller in (8) are given by the visual servoing. In particular, we consider as features the coordinates of a pixel in the image projection plane and we define $\boldsymbol{s}^{*}$ as desired value to be the center pixel in the projection plane, which is (0, 0). The object detection algorithm outputs the bounding box of the detected object, and we use the coordinates of its center as current features $\boldsymbol{s}$. We define the feature error as $\boldsymbol{e} = \boldsymbol{s} - \boldsymbol{s}^{*}$, which has to be minimized. The interaction matrix [31] or image Jacobian [32], here referred as $\boldsymbol{L}_{s} \in \mathbb{R}^{k\times6}$, where k is the number of features, links how the features vary if the camera moves. The interaction matrix of a 2D point in the projection plane is described as follows

$$\boldsymbol{L}_{s} = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^{2}) & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & (1+y^{2}) & -xy & -x \end{bmatrix}$$

where x and y are the coordinates of the point in the projection plane and Z is the $3D$ distance from the camera to the point in Cartesian space. It is common in the literature to refer to the estimated version of the interaction matrix as $\hat{\boldsymbol{L}}_{s}$, because Z depends on the camera calibration and on the quantities that are measured. Using only these features, the end-effector is free to change its roll orientation since all the rotations around the z axis of the camera are allowed. We impose the camera to stay always oriented parallel to the base, adding a third feature, s_{ϕ}, which constrains such rotation of the camera. When the camera is oriented parallel to the base, its x axis is always perpendicular to the z axis of the base frame. Hence, we impose

$$\boldsymbol{x}_c \cdot (\boldsymbol{R}_{cb}\boldsymbol{z}_b)^T = 0 \tag{9}$$

where we denote by $\boldsymbol{x}_c$ the camera x axis, $\boldsymbol{z}_b$ the base z axis, and $\boldsymbol{R}_{cb} \in \mathbb{R}^{3\times3}$ the rotation matrix from base to camera frame. The result of (9) is $(\boldsymbol{R}_{cb})_{zx} = 0$, with $(\boldsymbol{R}_{cb})_{zx}$ being the component at the third row and first column of $\boldsymbol{R}_{cb}$. The interaction matrix for s_ϕ, can be derived knowing that the translations of the camera cannot change its orientation, hence the first three columns of $\boldsymbol{L}_{s\phi}$ are zero, and the time derivative of a rotation matrix is the rotation matrix multiplied by the skew-symmetric of the angular velocity

$$\boldsymbol{L}_{s\phi} = \begin{bmatrix} 0 & 0 & 0 & 0 & -(\boldsymbol{R}_{cb})_{zz} & (\boldsymbol{R}_{cb})_{zy} \end{bmatrix} \tag{10}$$

where $(\boldsymbol{R}_{cb})_{zy}$ and $(\boldsymbol{R}_{cb})_{zz}$ are the components at the second column and third column in the third row of $\boldsymbol{R}_{cb}$. Stacking the features, we obtain the following error vector

$$\boldsymbol{e}_s = \begin{bmatrix} x & y & (\mathbf{R}_{cb})_{zx} \end{bmatrix}^T \tag{11}$$

To impose an exponential decay of the error, we derive the twist of the camera expressed in the camera frame as

$$\boldsymbol{\xi}_c^d = -\lambda \hat{\boldsymbol{L}}_s^+ \boldsymbol{e}_s \tag{12}$$

where $\boldsymbol{L}_s \in \mathbb{R}^{3\times6}$ denotes the interaction matrix of the three stacked features and $\hat{\boldsymbol{L}}_s^+$ denotes its estimated Moore-Penrose pseudo-inverse. The desired camera twist is mapped to the desired joint velocities and positions for the wrist as

$$\begin{cases} \dot{\boldsymbol{q}}_{wr}^d = \boldsymbol{J}_{wr}^+ \begin{bmatrix} \boldsymbol{R}_{cb}^T & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_{cb}^T \end{bmatrix} \boldsymbol{\xi}_c^d \\ q_{wr,i}^d = q_{wr,i}^d(0) + \int_0^T \dot{q}_{wr,i}^d \, dt, \quad \forall i = 1, .., N \end{cases} \tag{13}$$

where N is the number of wrist joints, $q_{wr,i}^d(0)$ the initial desidered joint configuration and T the spanned time interval.

Once the base is aligned with object, the wrist position is kept fixed, hence the references for (6) do not change. For the end-effector, the wrist is controlled as in (8) using the reference of visual servoing generated by (13). Instead regarding the base, its heading is controlled to be aligned with the one of the arm's end-effector and a walking forward velocity is commanded until the robot reaches the object proximity. In contrary to the Search, throughout most of the Approach phase, the robot's base is moving, and the effects of the legs are indirectly transmitted to the camera through the arm. We leverage on the capabilities of the impedance controller implemented at the Shelbow to mitigate tracking and external disturbance (induced by the base or any other source).

C. Grasp:

We enter into the Grasping phase when the robot is in the proximity of the target object and it can compute a grasping pose, i.e. the final position and orientation of the arm's end-effector. During the execution of this phase, the wrist position is kept at the same height of the object, by using the impedance controller (6), and its longitudinal distance is reduced. The latter choice is motivated by the fact that closer the robot moves to the grasping pose, the bigger the object gets in the camera image and the higher the chance for object detection to fail. Instead, the wrist is controlled using (8) to keep the object in sight thanks to the reference generated by visual servoing. Subsequently, the robot's base walks to reach the object within the arm workspace. Once that distance is reached, to leverage on the rotational DoFs, the base pitch is commanded to lean down or up, according to the estimated 3D object position as

$$\theta_b^d = \arctan\left(\frac{z_{so}}{x_{so}}\right) \tag{14}$$

where z_{so} and x_{so} denote the relative position of the object with respect to the shoulder, along z and x axis directions of the base frame, respectively. To complete grasping, the Shelbow group and the wrist are commanded to reach the previously computed grasping position. During this last phase, visual servoing is not active anymore being the camera too close to the object, and the grasping is performed open-loop. Reached the grasping pose, the gripper is commanded to close, the base adjusts its pitch, and the arm is repositioned to a default posture. The latter arm configuration can be in general optimized for avoiding self-collisions and increasing manipulability.

IV. Results

We performed several experiments to validate the proposed approach using our 140 kg hydraulic quadruped robot, HyQReal, and a Kinova Robotic Gen3 arm with 7-DoF. A Realsense D435i [33] is mounted at the arm's end-effector as shown in Fig. 1. We performed three types of experiments: *a)* disturbances applied on the arm on both positive and negative y-z axis of base frame while the robot is performing a trot in place (Section IV-A); *b)* the grasp of a bottle positioned at a fixed, unknown location in the room and out of the initial camera view (Section IV-B); and *c)* the grasp of a bottle thrown between two humans (Section IV-C). We used color segmentation throughout all the experiments, as object detection algorithm, with blue as color to recognize. The visual servoing gain λ is set to 3.0. For locomotion, a walking-trot gait is used, characterized by the alternated motion of diagonal leg pairs with a step frequency set to 1.3 Hz and a step duty factor equal to 0.6 (the duty factor is the ratio between the time a leg is in stance over the entire step period). To manage the sequence of actions, we developed a Behavior Tree based on the open-source project [34] (version 3.8). The Behavior Tree provides reactive behaviors to unforeseen events, such as the loss of the object from the camera view. All the experiments described in the following sections are also included in the accompanying video[1].

A. Visual servoing with external disturbance

During this first experiment, the quadruped manipulator is positioned in front of a bottle at a distance of around 1 m, and is commanded to keep the object in the field of view

[1] The accompanying video is also available at the following YouTube link: https://www.youtube.com/watch?v=ztMl52v3ncY

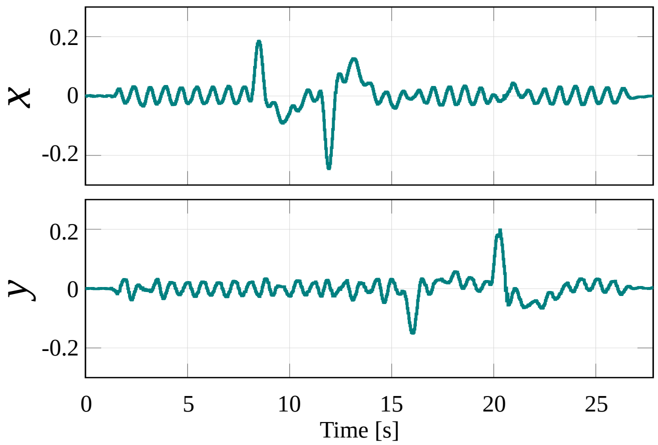

Fig. 3: Response of visual features, x-y center position of detected object (bottle) in the projection plane. Phases where visual error is higher coincide to time interval of interaction with human.

of the camera while trotting. The desired robot behavior is similar to the one expected for the Approach phase, where the quadruped manipulator has to walk and keep the object in sight. During the experiment, a disturbance is applied at the forearm by a human. First, the external forces are applied along (positive and negative) lateral and vertical directions, respectively along y and z axis of the robot's base frame. For the arm, the active controllers are: (6) for the Shelbow group, and (8) for the last three joints of the manipulator. The Cartesian impedances (6) along y, and z direction are set to: $K_p^{sh} = 50N/m$, $K_d^{sh} = 5Ns/m$. The impedance gains for the joint impedance controller, used by the last three joints, in (8) are set to: $K_{pj}^{wr} = 100N/rad$ and $K_{dj}^{wr} = 5Ns/rad$. When the human applies the force, the camera view is disturbed, hence the feature tracking is degraded, as shown in Fig. 3. Due to the impedance control strategy applied at the arm's wrist, the arm responds in a compliant way and does not try to rigidly hold its position as it would have done under velocity control. Additionally, the decoupled approach allows to have low impedances at the wrist because the Shelbow joints are not used for tracking of visual features. In terms of design, a trade-off can be established between position tracking accuracy and compliance, according to the context in which the robot has to operate. From Fig. 4, when the velocities return to zero, after a disturbance is applied, the visual features (x-y in projection plane) are zero, i.e. the object is back centered in the camera, as shown in Fig. 3 for $t \approx 8s$, $t \approx 12s$, $t \approx 17s$ and $t \approx 21s$. This means by the time the arm reaches the maximum elongation, the camera still has the object centered, as shown in the recorded sequence from Fig. 5.

B. *SAG of a bottle*

During this experiment, the quadruped manipulator has to execute the whole SAG pipeline. As it is possible to see from the picture on the left corner of Fig. 6, the Eye-in-Hand camera is not pointing towards the object at the start.

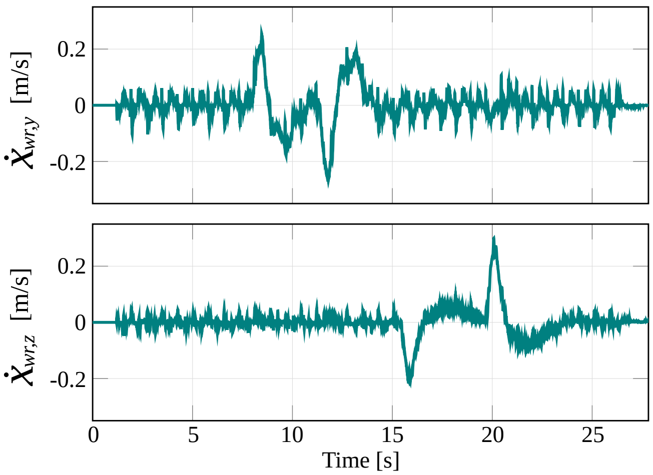

Fig. 4: Translational Cartesian velocities, relative to the Base frame, for arm's wrist along y (top) and z axis (bottom). Peaks of velocities correspond to maximum accelerations caused by human force on the arm.

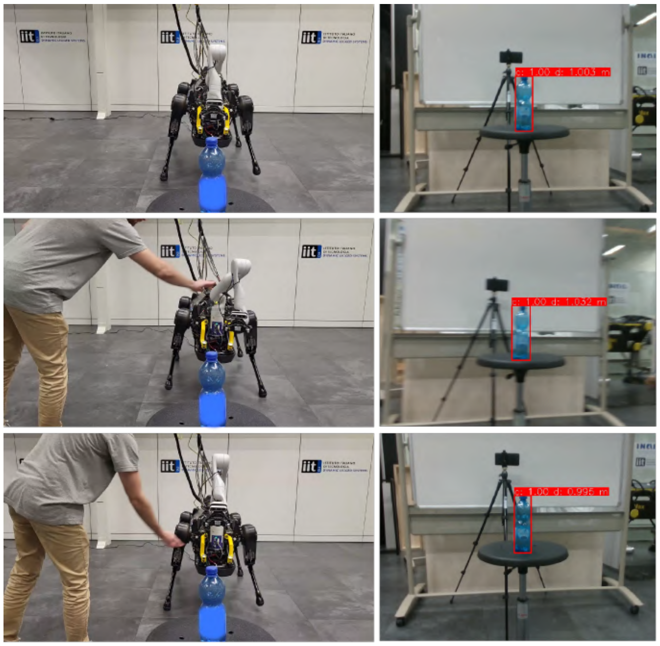

Fig. 5: Recorded sequence of HyQReal and its arm with respect to the object (left column) and the detected object from the camera view (right column). In the first row, the robot is trotting in place and, in the middle and bottom rows, the arm's elbow joint is about its maximum displacement from the initial position along lateral and longitudinal directions, respectively. The displacement is caused by the interaction with a human.

Hence, the robot uses its arm to rotate its camera around the trajectory defined in Section III-A, and it executes the Approach phase defined in Section III-B after the bottle is detected. The gains used for the active controllers during the Approach phase are defined as the previous experiment IV-

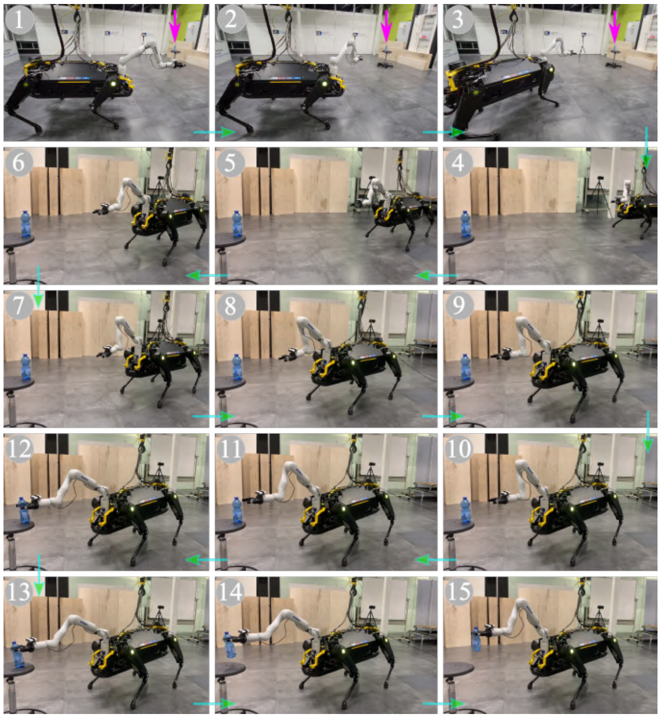

Fig. 6: Recorded sequence of HyQReal with a Kinova arm during the execution of the SAG pipeline. The snapshot sequence is read from the top to the bottom according to the numbers and light green arrows. Pink arrows indicate the localization of the bottle for the first three snapshots of the sequence (i.e., for S_1, S_2, and S_3). At S_1 the robot searches for the object. At S_2 the object is detected. At S_3 the robot adjusts its orientation w.r.t. the object. From S_4 to S_6 the robot approaches the object performing a walking-trot. From S_7 to S_{15} the robot moves its base and arm to prepare and execute the grasping.

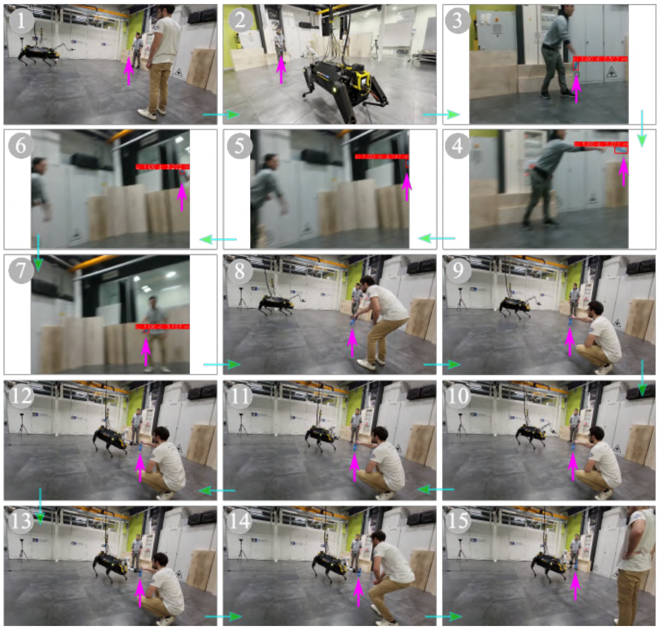

Fig. 7: Recorded sequence of HyQReal with a Kinova arm during the execution of the SAG pipeline while performing a walking-trot. The snapshot sequence is read from the top to the bottom according to the numbers and light green arrows. Pink arrows indicate the localization of the object in all the snapshots. The target object, handled by the first person, is detected at the first two snapshots (i.e. S_1 and S_2). From S_3 to S_7 the object is thrown to the second person. The reactiveness of the proposed method allows to maintain the object tracking from the launching to the catching. From S_8 to S_{10} HyQReal approaches the object. The grasping is prepared and executed from S_{11} to S_{15}.

A. At the end of the Approach phase, the robot is placed at 0.7 m far from the object. We relied on the Realsense depth estimation measurement to position the robot correctly in front of the object. The grasping phase spans along the last three rows of pictures in Fig. 6. From the bounding box of the detected object, we retrieve the pixel coordinates of its center, and we use this information, together with the depth measurement and the intrinsic parameters of the camera, to retrieve the 3D object position. As mentioned in Section III-C, in the proposed approach, the grasping position is calculated and then the base finalizes its last adjustments to bring the object inside the arm's workspace. Hence, we highlight here that the accuracy of the final grasping position depends on the accuracy of locomotion to properly position the base, and on the state estimation. From trials, we did not experience the need of re-calculating the grasping position during the Grasping phase.

C. *Visual tracking of a fast moving bottle*

During this experiment we challenged our system to keep a bottle in sight when throwing it between two people. More specifically, the quadruped manipulator starts to execute the SAG pipeline targeting the bottle handled by one of the two people present in the room. Afterwards, the bottle is thrown and passed from one to the other person. Along the entire bottle trajectory, the arm's end-effector is able to keep the bottle in the camera field of view, as shown in the image sequence of Fig. 7; the error in the visual features drives the camera to point to the flying object. Thanks to the decoupled approach, the wrist keeps the object centered and it acts as a helm, pointing towards the direction the base has to walk to. The reactivity of the wrist justifies the choice and the benefits of mapping directly the visual task to the last three joints of the manipulator. By executing the SAG pipeline, the quadruped manipulator is driven to grasp the bottle from the hands of the human as shown in the accompanying video.

V. Conclusions

In this work, we presented a control pipeline for the Search, Approach and Grasp of an object using a legged manipulator. The proposed approach defines a behavior sequence for the base and arm to solve the SAG problem, which integrates IBVS to maintain the object in the field of view of the camera and impedance control to render an active compliant behavior on the base and at the level of the wrist position. The main idea of the paper relies on assigning the visual task to the wrist, comprised normally by the last two or three joints, and the rest of the kinematic chain to place the wrist position in space. To validate the control approach, we executed experiments where the robot uses visual servoing and gets disturbed. The results show the arm's compliance and its ability to keep the object centered, thanks to the fast motions of the wrist. Additionally, we executed the complete SAG pipeline for grasping a bottle standing on a stool and to track and grasp a bottle thrown between two humans. As future work we aim at dealing with more complex surroundings, by considering obstacles and exploiting the robot positioning and posture to improve manipulation tasks. Additionally, in order to deploy this control architecture in more complicated and crowded environments, more robust neural network architectures are needed to properly detect and segment objects.

References

[1] G. Xin, F. Zeng, and K. Qin, “Loco-manipulation control for arm-mounted quadruped robots: Dynamic and kinematic strategies,” *Machines*, vol. 10, no. 8, 2022. [Online]. Available: https://www.mdpi.com/2075-1702/10/8/719

[2] Z. Fu, X. Cheng, and D. Pathak, “Deep whole-body control: Learning a unified policy for manipulation and locomotion,” 2022. [Online]. Available: https://arxiv.org/abs/2210.10044

[3] S. Zimmermann, R. Poranne, and S. Coros, “Go fetch! - dynamic grasps using boston dynamics spot with external robotic arm,” *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4488–4494, 2021.

[4] F. Jenelten, T. Miki, A. E. Vijayan, M. Bjelonic, and M. Hutter, “Perceptive locomotion in rough terrain online foothold optimization,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5370–5376, 2020.

[5] O. A. V. Magaña, V. Barasuol, M. Camurri, L. Franceschi, M. Focchi, M. Pontil, D. G. Caldwell, and C. Semini, “Fast and continuous foothold adaptation for dynamic locomotion through cnns,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2140–2147, 2019.

[6] D. Wisth, M. Camurri, and M. Fallon, “Vilens: Visual, inertial, lidar, and leg odometry for all-terrain legged robots,” *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 309–326, 2023.

[7] C. D. Bellicoso, K. Krämer, M. Stäuble, D. Sako, F. Jenelten, M. Bjelonic, and M. Hutter, “Alma-articulated locomotion and manipulation for a torque-controllable robot,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8477–8483.

[8] J. Li, H. Gao, Y. Wan, J. Humphreys, C. Peers, H. Yu, and C. Zhou, “Whole-body control for a torque-controlled legged mobile manipulator,” *Actuators*, vol. 11, no. 11, 2022. [Online]. Available: https://www.mdpi.com/2076-0825/11/11/304

[9] J.-P. Sleiman, F. Farshidian, M. V. Minniti, and M. Hutter, “A unified mpc framework for whole-body dynamic locomotion and manipulation,” *IEEE Robotics And Automation Letters*, 2021.

[10] H. Ferrolho, V. Ivan, W. X. Merkt, I. Havoutis, and S. Vijayakumar, “Roloma: Robust loco-manipulation for quadruped robots with arms,” *ArXiv*, vol. abs/2203.01446, 2022.

[11] G. Flandin, F. Chaumette, and E. Marchand, “Eye-in-hand/eye-to-hand cooperation for visual servoing,” in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 3, 2000, pp. 2741–2746 vol.3.

[12] S. Hutchinson, G. Hager, and P. Corke, “A tutorial on visual servo control,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.

[13] G. Chesi and K. Hashimoto, “Effects of camera calibration errors on static-eye and hand-eye visual servoing,” *Advanced Robotics*, vol. 17, no. 10, pp. 1023–1039, 2003. [Online]. Available: https://doi.org/10.1163/156855303322554409

[14] H. Jabbari, G. Oriolo, and H. Bolandi, “An adaptive scheme for image-based visual servoing of an underactuated uav,” *Int. J. Robotics Autom.*, vol. 29, 2014.

[15] G. Mariottini, D. Prattichizzo, and G. Oriolo, “Image-based visual servoing for nonholonomic mobile robots with central catadioptric camera,” in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2006, pp. 538–544.

[16] G. Allibert, E. Courtial, and Y. Tour, “Real-time visual predictive controller for image-based trajectory tracking of a mobile robot,” *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 11 244–11 249, 2008, 17th IFAC World Congress. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667016407731

[17] J. P. Alepuz, M. R. Emami, and J. Pomares, “Direct image-based visual servoing of free-floating space manipulators,” *Aerospace Science and Technology*, vol. 55, pp. 1–9, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1270963816301821

[18] E. Malis, F. Chaumette, and S. Boudet, “2 1/2 d visual servoing,” *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 238–250, 1999.

[19] Kinova gen3 7-dof manipulator arm. [Online]. Available: https://www.kinovarobotics.com/product/gen3-robots#Product_resources

[20] V. A. Shim, M. Yuan, and B. H. Tan, “Automatic object searching by a mobile robot with single rgb-d camera,” in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2017, pp. 056–062.

[21] B. A. Griffin, V. Florence, and J. J. Corso, “Video object segmentation-based visual servo control and object depth estimation on a mobile robot,” in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. Los Alamitos, CA, USA: IEEE Computer Society, mar 2020, pp. 1636–1646. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/WACV45572.2020.9093335

[22] T. Boroushaki, I. Perper, M. Nachin, A. Rodriguez, and F. Adib, “Rfusion: Robotic grasping via rf-visual sensing and learning,” in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 192205. [Online]. Available: https://doi.org/10.1145/3485730.3485944

[23] P. Ardn, M. Dragone, and M. S. Erden, *Reaching and Grasping of Objects by Humanoid Robots Through Visual Servoing*, 06 2018, pp. 353–365.

[24] B. Griffin, “Mobile robot manipulation using pure object detection,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2023, pp. 561–571.

[25] A. D. Luca, G. Oriolo, and P. R. Giordano, “Image-based visual servoing schemes for nonholonomic mobile manipulators,” *Robotica*, vol. 25, pp. 131 – 145, 2007.

[26] D. J. Agravante, G. Claudio, F. Spindler, and F. Chaumette, “Visual servoing in an optimization framework for the whole-body control of humanoid robots,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 608–615, 2017.

[27] M. Focchi, A. D. Prete, I. Havoutis, R. Featherstone, D. G. Caldwell, and C. Semini, “High-slope terrain locomotion for torque-controlled quadruped robots,” *Autonomous Robots*, vol. 41, pp. 259–272, 2017.

[28] V. Barasuol, J. Buchli, C. Semini, M. Frigerio, E. R. De Pieri, and D. G. Caldwell, “A reactive controller framework for quadrupedal locomotion on challenging terrain,” in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 2554–2561.

[29] Kinova. [Online]. Available: https://www.kinovarobotics.com/product/gen3-robots

[30] F. Emika. Robot instruction handbook. [Online]. Available: https://www.generationrobots.com/media/franka-emika-robot-handbook.pdf

[31] B. Espiau, F. Chaumette, and P. Rives, “A new approach to visual servoing in robotics,” *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, 1992.

[32] F. Chaumette and S. Hutchinson, “Visual servo control. i. basic approaches,” *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[33] https://www.intelrealsense.com/depth-camera d435i/. Intel realsense depth camera d435i. [Online]. Available: https://www.intelrealsense.com/depth-camera-d435i/

[34] M. C. Davide Faconti. Behavior tree. [Online]. Available: https://www.behaviortree.dev/