HYBRID-ADAPTIVE SWITCHED CONTROL FOR ROBOTIC MANIPULATOR INTERACTING WITH ARBITRARY SURFACE SHAPES UNDER MULTI-SENSORY GUIDANCE

by

Danial Nakhaeinia

Thesis submitted in partial fulfillment of the requirements for

Ph.D. in Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Electrical Engineering and Computer Science
University of Ottawa

Abstract

Industrial robots rapidly gained popularity as they can perform tasks quickly, repeatedly and accurately in static environments. However, in modern manufacturing, robots should also be able to safely interact with arbitrary objects and dynamically adapt their behavior to various situations. The large masses and rigid constructions of industrial robots prevent them from easily being re-tasked. In this context, this work proposes an immediate solution to make rigid manipulators compliant and able to efficiently handle object interactions, with only an add-on module (a custom designed instrumented compliant wrist) and an original control framework which can easily be ported to different manipulators. The proposed system utilizes both offline and online trajectory planning to achieve fully automated object interaction and surface following with or without contact where no prior knowledge of the objects is available.

To minimize the complexity of the task, the problem is formulated into four interaction motion modes: free, proximity, contact and a blend of those. The *free motion* mode guides the robot towards the object of interest using information provided by a RGB-D sensor. The RGB-D sensor is used to collect raw 3D information on the environment and construct an approximate 3D model of an object of interest in the scene. In order to completely explore the object, a novel coverage path planning technique is proposed to generate a primary (offline) trajectory. However, RGB-D sensors provide only limited accuracy on the depth measurements and create blind spot when it reaches close to surfaces. Therefore, the offline trajectory is then further refined by applying the *proximity motion* mode and *contact motion* mode or a blend of them (*blend motion* mode) that allow the robot to dynamically interact with arbitrary objects and adapt to the surfaces it approaches or touches using live proximity and contact feedback from the compliant wrist.

To achieve seamless and efficient integration of the sensory information and smoothly switch between different interaction modes, an original hybrid switching scheme is proposed that applies a supervisory (decision making) module and a mixture of hard and blend switches to support data fusion from multiple sensing sources by combining pairs of the main motion modes. Experimental results using a CRS-F3 manipulator demonstrate the feasibility and performance of the proposed method.

ACKNOWLEDGEMENTS

This thesis represents not only my work at the keyboard but it is the lessons I have learned, many experiences I have gained and the result of works I have done with the invaluable help of remarkable individuals who I wish to thank and acknowledge.

First and foremost I would like to thank my supervisor Professor Pierre Payeur for all his contributions, guidance, patience and support through the rough road to finish this thesis. I would also like to thank my co-supervisor Professor Robert Laganière, for all the support and encouragement over the course of this research and several publications.

I wish to acknowledge the contribution of Mr. Pascal Laferrière to this research via the development of the instrumented compliant wrist, as well as the financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) provided under the Strategic Project and Discovery Research Grant programs. Appreciation also goes to the Canada Foundation for Innovation (CFI) that made possible the high-end robotic research infrastructure used in this work.

A heartfelt thanks to my family and specially my beloved parents. Words cannot express how grateful I am for the endless support and love you gave me to achieve my goal.

Dedication

To

The Most Holy Person, Mother

The angel who is always my supportive

Dearest Person, Father

A strong and gentle soul who is my guardian angel

Best friends, Siblings

Gifts from God who are my inspiration

TABLE OF CONTENTS

List of Figures

List of Tables

Chapter 1. INTRODUCTION

1.1 Motivation and Problem Statement

Industrial robots rapidly gained popularity in manufacturing because of performing tasks quickly, repeatedly and accurately in relatively static environments. The first approaches to control a robot were developed based on the definition of a highly structured model of the environment and design of the robots for a specific task. These robots cannot see and perceive their workspace. They are mainly applicable only for specific purposes when there is no change in the environment. As robotic technology develops, robots start to move out of factories and are being repurposed for alternate tasks and applications. However, beside the kinematic capabilities of industrial robots, their heavy and rigid construction prevent them from easily being re-tasked. Therefore, repurposing them for alternate tasks and applications requires precise control of forces and motion, and cannot be fully accomplished without a comprehensive set of sensing modalities that monitor the position, orientation, shape, surface characteristics and even objects transformation under external constraints. For that matter, it is essential to develop innovative integrated sensing and control methods to provide next generation robots with increased versatility that will make them perform closer to what human beings can achieve.

The latest forecast from the International Federation of Robotics (IFR) shows that more than 1.4 million new industrial robots are expected to be installed and the number of industrial robots deployed worldwide will increase to around 2.6 million units by 2019. In the race for automation in manufacturing, the demand in Canada increased by 49 percent in 2015 and annual growth of 5 to 10 percent in sales of robots in North America is expected from 2016 to 2019. A significant number of the industrial robots (around 70 percent) are currently applied in the automotive, electrical, electronics and metal and machinery industries [1]. In order to extend the usability of industrial robots currently at work or the ones that will be installed in the near future, these robots should be able to react and self-adapt their behavior to the configuration of or changes in the environment. The flexibility and adaptation to various unmodeled objects in arbitrary positions raise new challenges in reactive object interaction and exploration tasks which involve a full range of sensing, planning and fine control steps that must be performed by proper planning and control

algorithms, and be seamlessly integrated. Therefore, smooth interaction with an object and complete following of its surface under consideration of the kinematic constraints and also environment uncertainties such as possible deformation and movement of the objects at the contact point, are the problems addressed in this thesis. It consists of achieving complete exploration and alignment of a robot's end-effector with an unknown, and not precisely located, surface while accommodating its curvature.

This work is motivated by applications in inspection, cleaning, painting, welding and particularly security screening. The research was conducted in the specific context of detecting dangerous or prohibited substances hidden in a vehicle, boxes and parcel post without direct human intervention, an asset that is critical to ensure the security of populations and properties worldwide. While very efficient technologies exist to detect the presence of minute amounts of dangerous material particles, the process of collecting such particles safely and efficiently in an automated manner remains a challenge. In order to accomplish such complex operations over surfaces of arbitrary shape, the robotic system must be able to sense the environment and analyse the information accurately, but still rapidly. In the literature, a large range of sensors are frequently employed to support such complex tasks. Unfortunately, many types of sensors are not capable to capture color and depth images simultaneously, or are slow and require a considerable amount of time to perform range acquisition. To overcome these limitations, several attempts have been made to capitalize on novel consumer-grade RGB-D cameras such as the Microsoft Kinect sensor. This sensor is able to generate high density depth maps and corresponding color images in a fraction of a second. However, important limitations remain with these sensing solutions. Beyond a limited depth resolution, their bounded depth of field results in the lack of adequate sensing in close proximity to surfaces. Also, they do not provide any touch/tactile sensing capabilities to ensure robust interaction control when the robot is in contact with an object. Therefore, this work aims at bringing object-robot interaction and robotic surface following functionalities at a higher level of performance, dexterity and operational capabilities, by focusing on data collection, processing, and decision stages that are not yet well mastered by state-of-the-art technologies. In this context, this work addresses two key issues of flexibility and adaptability in reactive interaction with objects by introducing new adaptive controllers and an original hybrid switched motion control

system that integrates multimodal sensory information (vision, proximity, touch) for offline and online motion planning supported by a Kinect sensor and a specially designed instrumented compliant wrist with embedded proximity and contact sensors.

1.2 Objectives

This work aims to provide a solution to retrofit on classical rigid industrial robots and provide them with a level of compliance to afford the necessary flexibility and adaptability with only the addition of an instrumented wrist device and a control layer that does not affect the internal controller of the robot (from the manufacturer) to make them more flexible and adequate to tackle complex tasks that they were not initially designed for. To achieve the goal, the main objectives of this work are as follows:

- To develop innovative means for robotic systems to automatically scan, explore, and interact with miscellaneous objects, and react in real-time to dynamic changes in the environment while performing integrated multi-sensorial (vision, proximity, touch) exploration and interaction.

- To develop a global trajectory planning method for customized robot-object interaction and surface following approach using fast but low accuracy RGB-D sensors when no contact with an object is required.

- To design and develop unique adaptive controllers for the collection, mapping and processing of proximity and contact sensing stages data for a robotic manipulator to achieve smooth and reliable online close proximity and contact interaction, and surface following.

- To achieve a seamless and efficient integration of multimodal sensory information by smoothly switching between different interaction modes using an original hybrid switched control system.

Besides contributing to scientific knowledge in control systems, sensing and robotics, the results of this research are of immediate benefit to advanced manufacturing, security robotics, human-robot interaction, transformable/deformable objects manipulation, or assistive robotics in uncontrolled environments.

1.3 Thesis Organization

The following chapters contained in this thesis are organized as follows.

Chapter 2 provides a review of a wide range of research domains which are related to this work. The major focus of this chapter is to address the research problems in literature and describe the existing challenges.

Chapter 3 presents the research methodology and the proposed system design. The first section of the chapter introduces the general perspective of the system design. In the next section, the proposed offline (free motion) path planning method to generate a global trajectory to scan and explore the object's surface completely under guidance from only a peripheral RGB-D vision stage is presented. The third section provides details of the design and implementation of the online (local) motion planning strategy using sensory information provided by an instrument compliant wrist with embedded proximity and contact sensing stages. It also details the hybrid switching control scheme that is developed to integrate and switch smoothly between all interaction modes.

Chapter 4 describes experimental results obtained with a CRS-F3 manipulator equipped with a custom-built compliant wrist to validate performance and efficiency of the proposed system in offline and online object-robot interaction and surface following tasks. The results are also compared with other methods.

Chapter 5 concludes this work, followed by a summary of contributions and suggestions for future research.

Chapter 2. BACKGROUND AND RELATED WORK

As robots system become more flexible, the costly and time consuming pre-programming and teaching process can be progressively eliminated to enable them to operate in different places and tasks. A significant number of approaches have been proposed to control industrial robots in dealing with uncertainty by incorporating additional sensors, redesigning the robot mechanically or developing new control methodologies. In general, the previously proposed solutions for an industrial manipulator to perform interaction with objects and tackle the surface following problem can be divided into 4 categories: 1) equip robot with or incorporate additional sensors, 2) modify or mechanically redesign of the manipulator, 3) modify trajectory by designing and developing adaptive controllers, or 4) develop hybrid control strategies and switching control systems.

This chapter presents an overview of the previously proposed approaches and developments in areas related to this research in the four above mentioned categories. The performance of the most frequently used approaches related to object-robot interaction and surface following and their pros and cons are reviewed and discussed.

2.1 Sensor-Based Control

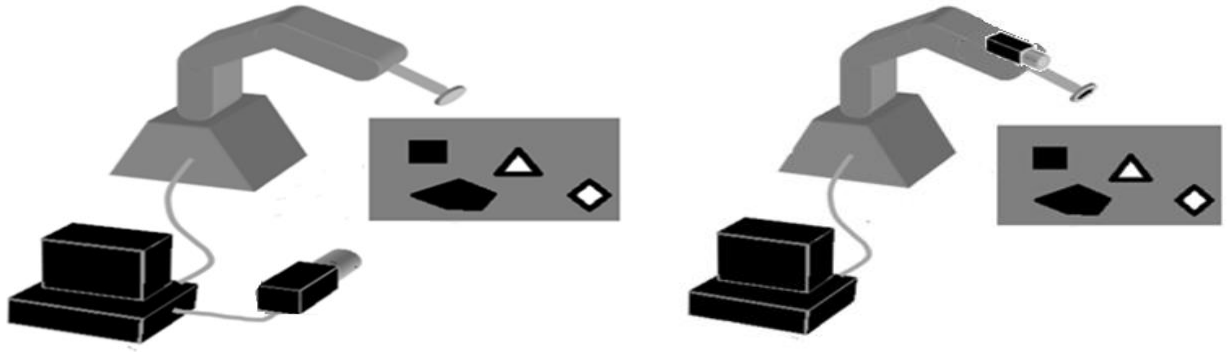
Three types of sensors are mostly used for robotic manipulators control: vision (2D/3D cameras), force/torque, and tactile. Vision sensors provide global or local information to localize the robot and the objects relative to their environment. This information is used by the robot controller to reach the object or target while avoiding undesired obstacles. On the other hand, force/torque and tactile sensors information are available when the robot enters into contact with the object. The local surface properties can be extracted from the contact location which may provide supplementary information for the robot to identify and manipulate the object. According to the information available from sensors, four frequently used robotic control strategies have been developed to control the robot's movements: visual servoing, force/torque control, tactile sensing control, and hybrid control (vision/force, vision/tactile, and position/force). This section reviews such sensing and control strategies used in surface following and object-robot interaction tasks.

2.1.1 Visual Servoing

Using visual feedback in order to control a robot is commonly called visual servoing [2]. Usually, visual features such as points, lines and regions are used in various object manipulation tasks such as alignment of a robot manipulator with an object. A visual servoing system acquires a scene in the robot environment by one or more cameras in order to control the robot end-effector's pose with respect to an object in the scene or to feedback local information about a specific object for further manipulation. The camera can be placed in different positions. There are two typical configurations according to the camera position in a visual servoing system: eye-to-hand (Figure 2.1a), or eye-in-hand (Figure 2.1b).

Independently from the camera position, calibration is usually required to determine the camera's intrinsic parameters such as the image center, focal length and aspect ratio, and extrinsic parameters related to the relative position and orientation of the camera with respect to the robot. The camera can be mounted on the robot end-effector (eye-in-hand) to provide more precise vision of the local environment, or it can be mounted at a fixed position (eye-to-hand) to perform other tasks where the simultaneous observation of the robot and its environment is required.

Various visual servoing approaches have been proposed for more than three decades starting from simple pick–and–place tasks to today's real-time, advanced manipulation of objects. Lane tracking for cars, navigation for mobile platforms and manipulation of objects are some applications of visual servoing. It is generally used for automated manipulation of objects which requires object detection, feature extraction, segmentation, recognition, servoing, alignment and grasping. Visual control techniques can be classified as open-loop or closed-loop control (that is visual servoing) based on the input reference of the control loop.



(a) (b)

Figure 2.1: a) Eye-to-hand configuration, and b) eye-in-hand configuration.

Open-loop robot control initially was used in robotic systems incorporating computer vision known as "look-and-move" technique [3]. In this approach, the vision system works as a pose estimator in order to obtain the required motion command for the robot (Figure 2.2). First, the robot sees and recognizes the environment helped by a computer vision system, and then, it performs the motion based on the data acquired in the previous step [4]. In an open-loop control system using the "look-and-move" approach, it is assumed that the object position does not change over the execution time, from the moment the vision system obtains the object position until the robot reaches this position. For example in a task where the robot must reach the position of an object in the workspace, the open-loop system does not check if the object is reached during the task or once the robot reaches the final position. Therefore, the accuracy in positioning the end-effector with respect to the target object depends on the accuracy of the visual sensor, the exactness of feature extraction, the robot kinematic calibration and the camera calibration.

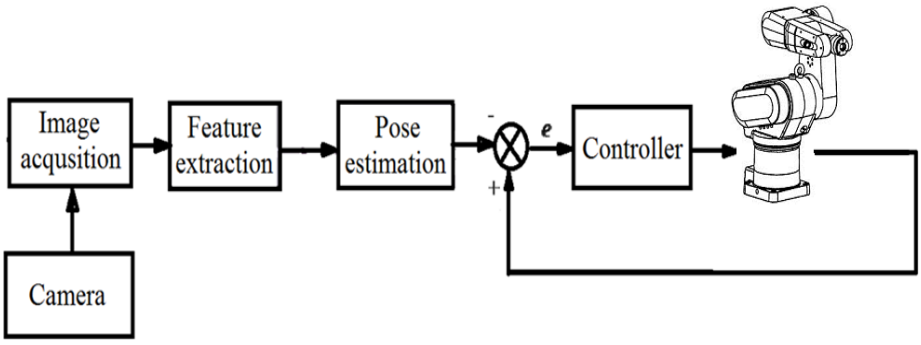


Figure 2.2: Open-loop visual control.

Simple and repetitive assembly tasks can be performed quickly through look-and-move open-loop controls. However, they are not proper for complex tasks due to inaccuracy in calibration or modelling. An open-loop visual servoing scheme for positioning and tracking task is presented in [5] that computes the reference input of the robot joints controller with the target's projection location in the image plane. The operation is performed only once. Fujimoto [6] applied a multi-rate controller to the visual servo system of a 6 DOF manipulator which takes advantage of both feed-forward and feedback schemes. In [7] the open-loop controller allows the robot to perform faster movements and does not require visual feedback from the hand. When sight of the hand is available within the camera field of view, the closed-loop controller allows for precise positioning of the hand in the image plane. More recently, Chang and Wu [8] presented a vision-based control approach for hand-eye robotic assembly tasks. In this work, the look-then-move open-loop control or closed-loop control laws are synthesized to accomplish the required assembly task.

Closed-loop control is an alternative to the previous approach. It is based on the explicit use of visual information in the control feedback loop [2-9], which leads to the concept of visual servoing. This approach permits to correct possible errors in the object position estimation obtained from the computer vision system. Moreover, it permits to adapt the robot trajectory in view of possible movements of the objects in the workspace. Hence, vision is part of a control system where it provides feedback about the state of the environment. A closed-loop control of a robot system consists of two intertwined processes: tracking and control. Tracking provides a continuous estimation and update of features during the respective robot and object motions. Based on this sensory input, a control sequence is generated. In addition, the system may also require an automatic initialization which commonly includes feature segmentation and object recognition. Visual servoing approaches can be divided in two different groups: direct and indirect visual servoing respectively.

The robot internal controller is not present in direct visual servoing (Figure 2.3). Thus, the internal joints controller of the robot is replaced by the visual controller, which uses the vision system data directly to control and stabilize the robot. In other words, visual feedback is used to directly

compute joints inputs and stabilize the mechanism. The full nonlinear dynamic model of the robot is considered in the control analysis [10].

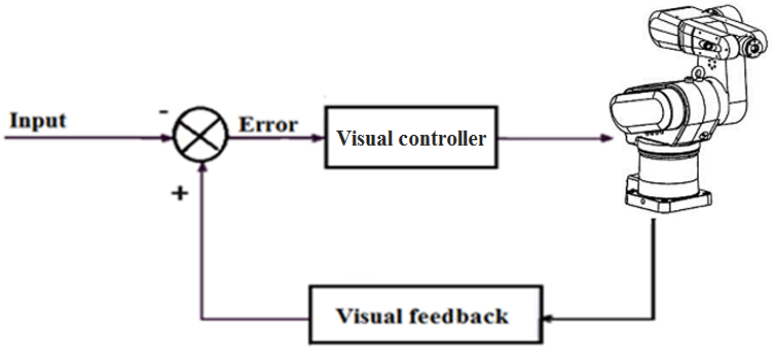


Figure 2.3: Direct visual servoing.

The first solution to the direct visual servoing problem was proposed by Miyazaki and Masutani [11]. Similar works were proposed by Espiau *et al*. [12] and Kelly *et al*. [13], for which the control of interaction between a robot and its environment is based on data provided by a visual sensor inside the closed-loop control scheme in a camera-in-hand configuration. In [14-16], the dynamic model of the manipulator in the control analysis and the robustness against parametric uncertainties of the vision system are considered. Later on, Cheah *et al*. [17] proposed a vision based tracking controller with adaptation to uncertainty in depth information. In [18], a direct dynamic visual servoing at high sampling rates was presented for machines used for production of devices that inherently consist of equal features placed in a repetitive pattern. The motor inputs are driven directly by the vision controller without the intervention of low level joint controllers. Silveira and Malis [19-20] proposed a direct visual servoing technique where the control error as well as the control law is fully based on image data and the visual features used in the control law are the pixel intensity (i.e. metric measures are neither required nor estimated). In other works, a direct visual servoing system is described in [21] that employs a network of cameras providing high-speed vision feedback that is robust to occlusions. More recently, Tamadazte *et al*. [22] demonstrated an accurate micro-positioning scheme based on a direct visual servoing process. This technique uses only the pure image signal (photometric information) to design the control law.

In contrast to direct control, indirect visual servoing has a hierarchical or cascaded control architecture. The indirect visual servoing category can be split into two sub-categories: position-

based and image-based visual servoing respectively [23]. Both techniques are usually implemented in an external loop that provides an input to the robot internal controller.

2.1.1.1 Image-based Visual Servoing

In image-based visual servoing (IBVS), control values are computed on the basis of image features directly. In this approach, image features of the objects in the visual data are used to determine the control law without the necessity of calculating their 3D location. It results in a more robust approach regarding calibration errors than position-based visual servoing. This approach is also known as 2D visual servoing because the control action is computed in the two dimensional space of the image. 2D visual features such as points, corners, lines and regions are typically used in an image-based visual servoing system. The features help to recognize the projection of a specific object in the scene without ambiguity. In contrast to position-based visual servoing, image-based visual servoing is more suitable when a geometric model of the task is not available. IBVS has three main advantages in comparison with other visual servoing methods: *i*) it is "model-free", which means that it does not require a model of the target object, *ii*) it is robust to camera model errors, and *iii*) it is also insensitive to camera calibration errors [24].

Figure 2.4 shows the general architecture of an image-based visual servoing system. The loop control of IBVS is fed back with the visual features obtained from the images, without the necessity of determining the 3D pose of the object to reach. The control actions of the image-based visual servoing system to move the robot at each iteration are performed by reducing the image distance error between a set of current and desired image features in the image plane. Once the error is zero (i.e., the current and the desired features match), the task concludes.

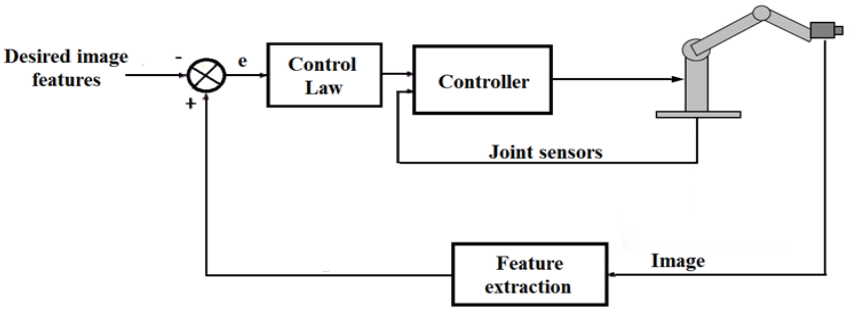


Figure 2.4: Image-based visual servoing architecture.

The velocity of the camera (C_v), is the input for controlling the robot movements in a basic IBVS approach. The control law is obtained using the interaction matrix or image Jacobian (J_s). The interaction matrix relates the variations of the visual features ($\dot{f}$) in the image with the variations of the poses of the camera in the 3D space, i.e., its velocity [25] :

$$\dot{f} = J_s . C_v \tag{2.1}$$

The interaction matrix is a function of the current value of the visual features in pixels, the intrinsic camera parameters, and the 3D information relating to the 3D points corresponding to the visual features (e.g., the distance from the camera to the corresponding object feature). The way to compute the interaction matrix is a major topic of research in image-based visual servoing. Offline update is the simplest way to compute the interaction matrix which is computed only once before the visual servoing task is started. The interaction matrix depends on the value of the visual features. Therefore, the distance in the final position defines the distance between the camera and the object. The number of computing operations is reduced using the approach during the visual servoing development and maintains the convergence [26]. Nevertheless, convergence problems in large movements have been stated in where the desired visual features are used to compute the interaction matrix [27]. Moreover, the visual features may get out of the camera view and cause the failure of the task. Therefore, the local minima of the image features and the singularities of the interaction matrix are drawbacks of the IBVS which may lead to the IBVS failure. A key point to solve the problem of singularity of the interaction matrix and the local minima of the image features is the choice of image features. Several efforts have been made to determine a better solution than offline computation of the interaction matrix such as decoupling image features to deliver a triangular or diagonal interaction matrix [28-30]. Using proper image moments as image features in visual servoing renders the interaction matrix with a maximal decoupled structure [29]. Therefore, the singularity problem of the interaction matrix is avoided and the performance of the IBVS system is significantly improved.

In order to obtain a permanently updated interaction matrix, the intrinsic camera parameters and/or the camera-object depth values must be estimated during the visual servoing task. The intrinsic camera parameters are usually computed offline in a calibration phase, and as such the camera intrinsic parameters are constant. However, in some visual servoing tasks the intrinsic parameters

are not constant or they require zooming to provide high precision images. Camera online calibration was developed in [31, 32] using the images acquired during a visual servoing task. Pomares *et al*. [31] proposed a method based on virtual visual servoing where the real intrinsic camera parameters are computed from the images acquired up to the moment. An adaptive IBVS controller was developed in [32] where the value of the intrinsic parameters are determined during the task that regulated the feature points in an image to the desired locations.

The offline intrinsic camera parameters and depth estimation and its implementation are easier and less time consuming than online estimation. Nevertheless, the visual features may still get out of the image plane and the visual servoing task may fail. Therefore, the visibility problem has received particular attention in the literature. To solve the problem, a minimum number of image features must remain in the camera's field of view during visual servoing. The most widespread solutions to assure the visibility of all features during the control task are based on using potential fields [33]. A potential function was proposed in [34] to guarantee that all features remain in the image during the control task and repel feature points from the boundary of the image plane. Intrinsic-free visual servoing [35, 36] is another solution to this issue that zooms in order to keep all features in the field of view throughout the trajectory.

In addition to the previous problems of online calibration, the camera-object depth estimation is not an easy task and it can introduce some undesirable behaviors. Cervera *et al*. [37, 38] introduced 3D information in the feature vector of the control law. In this approach, the 3D pose estimation is not performed but the distance or depth from the camera to each characteristic point is computed. The linear displacement of the 3D object with respect to the camera is obtained using the feature vector which includes 3D information. In similar work, De Luca *et al*. [39] proposed a method to estimate on-line the value of the depth for feature points while the camera is moving through the scene, by using tools from nonlinear observer theory. Using the method, an object most likely remains in the field of view of the camera during the task. However, the results can be improved using a 3D pose estimation from the intrinsic camera parameters and the estimated depth. Therefore, a proper camera calibration and 3D information obtained by a pose estimation algorithm or a stereo visual system is required.

The complexity of the objects in the scene and finding adequate visual features is another important issue for the temporal efficiency of visual servoing systems. To perform the IBVS the object should convert to its simplest form. Various techniques are used for visual features extraction. Therefore, the image processing tasks are mainly segmentation of the specific features and computation of their center of gravity. Pages *et al*. [40, 41] solved the complexity problem by adding external features to any object in the scene with a structured light emitter that is linked to the camera to produce a suitable set of visual features. The system is applicable for plane-to-plane tasks where only three degrees of freedom are considered. According to the visual features vector, the interaction matrix is only decoupled into local conditions for the positions near the desired location [42], or the global convergence in [43]. The global convergence proves robust against calibration errors. Visual features are provided by a structured light emitter based on laser pointers, and by defining an image transformation, the misalignment between the camera and the lasers has been improved. To achieve the analysis of stability for the previous system a new set of visual features is described in [42]. However, extracting visual features is too complex or too time consuming for positioning with respect to non-textured objects. Pages *et al*. solved the problem of visual servoing tasks with non-featured objects [43] by projecting coded structured light to the scene. Visual features are available independently from the object appearance with the use of coded patterns. Experiments show that good results are obtained when the classical IBVS control law is applied using the points drawn on the object surface and the robot is positioned with respect to planar objects. On the other hand, when using non-planar objects, the results show that the camera motion is noisier, slower and less monotonic. Furthermore, the online interaction matrix estimation is not possible because the camera and the projector are not calibrated.

The image-based approach may reduce computational delay, eliminate the necessity for image interpretation and eliminate errors due to sensor modeling and camera calibration. However, it does present a significant challenge to controller design since the system is non-linear and highly coupled. Furthermore, one of the problems with IBVS schemes is that it is difficult to estimate depth.

2.1.1.2 Position-based Visual Servoing

In a position-based visual servoing (PBVS) system [also called 3D visual servoing], image features are extracted from the image and a model of the scene and the target is used to determine the pose of the target with respect to the frame attached to the camera. A camera is the sensor used to estimate the position and the orientation (pose) of the target with respect to the camera coordinate system (Figure 2.5). The position (P_C) and orientation (φ_C) of the object with respect to the reference coordinate frame of the camera defines the current features. The reference input is the pose of the object with respect to the camera frame at the robot desired position (P_d) and orientation (φ_d). The current and the desired features are expressed in three dimensional (3D) coordinates. Finally, the control error (e) is defined by the difference between current and desired 3D poses. The 3D pose of the robot and the kinematic error is generated in the Cartesian space and mapped to actuators commands. This classical computer vision problem is called the 3D localization problem.

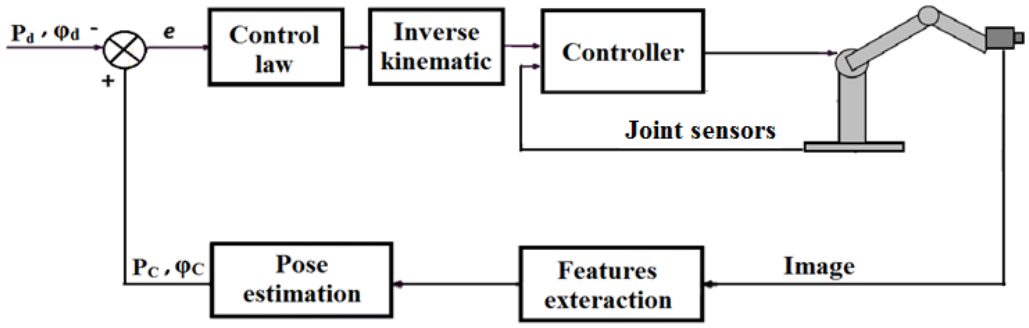


Figure 2.5: Position-based control architecture.

The main advantage of using position-based control is in defining tasks in a standard Cartesian frame, and if the pose parameters are accurate, the robot trajectory is straight and without oscillation. On the other hand, the control law depends on the vision system calibration parameters, and can become widely sensitive to calibration errors. Moreover, it is impossible to ensure that the object will always remain in the camera field of view during the entire duration of the servoing task.

Various methods were proposed to solve the problems. Wilson *et al*. [44] proposed a Cartesian position-based visual servo control for robots with a single camera mounted at the end-effector.

They applied an extended Kalman filter (EKF) to obtain a recursive solution of the photogrammetric equations, and to properly combine redundant measurements. In a similar work, an adaptive Kalman filter was proposed in [45] to address the problem of nonlinear systems that cannot be linearized or where the model is unavailable. In [46] a position-based visual servoing approach was developed for a multiarm robot based on the real-time estimation of the pose of a target object using the extended Kalman filter. Janabi-Sharifi and Marey [47] presented a new algorithm, namely iterative adaptive EKF (IAEKF) by integrating mechanisms for noise adaptation and iterative-measurement linearization to estimate position and orientation of an object in real time. In [48], it is shown that a visual sensor elaborating 3D features at video rate can be used to estimate the pose of the target object and reach a specific pose between the sensor frame and a target object frame. In [49], the authors used a mixed 2D-3D approach in order to take into account the size of the shape in the image space. They defined an ellipsis which includes all the features used in the reconstruction algorithm, and design a control law in order to keep it in the image plane. Thuilot *et al*. [50] presented a position-based modeling approach in order to guarantee that the object remains in the field of view of the camera during the whole robot motion by using a representation of the pose which separates the rotational and translational dynamics. Chesi *et al*. [51] proposed a switching controller to address a similar problem. As long as all feature points are inside the image frame, a position based controller is used. Once a feature enters the boundary of the image frame, a controller consisting of either pure rotational motion or pure translational motion is activated. The drawback of this switching controller is the presence of chattering. In [52], Murao *et al*. presented a receding horizon approach for the stabilization of a robot arm using a position-based visual controller. In each iteration a nonlinear optimization problem is solved.

A position-based visual servoing system (PBVS) impedance control was proposed by Lippiello *et al*. [53], and a method for geometric reconstruction was presented in [54] for use in PBVS. It offers the ability to control the pose of the camera or a body with respect to a planar object in the scene. Herrejon *et al*. [55] proposed a PBVS for catching a 3D flying object using Recursive Least Squares (RLS) trajectory estimation from a monocular image sequence. More recently, the problem of moving a camera from an initial pose to a final pose is addressed in [56] that formulates a convex optimization problem for the translational motion. In [57] the proposed scheme is

designed based on a combination of the relative pose (position and orientation) of a vision system with information from an inertial sensor (acceleration and angular velocity) to get a final relative pose. In addition, a Kalman filter is used to handle asynchronous information from those sensors.

The position-based visual servoing has some shortcomings to accomplish servoing tasks: *i*) it requires the estimation of the pose of the target, which requires some form of a model, and *ii*) precise system calibration (camera, camera/robot) is required to estimate the desired velocity of the robot and in order to achieve accurate positioning.

Stereo configurations and the Kinect sensor have been widely applied in the literature to obtain 3D information from the scene and reduce the complexity of the PBVS. Stereo vision was first applied in visual servoing by Kase *et al*. [3]. Stereo visual servoing presents some particularities in comparison to monocular visual servoing. There are two cameras and two images at each control loop iteration where each image has a different center of gravity. In [58], Cervera and Martinet described a stereo visual servoing control scheme using 2D information. In this work the interaction matrix is composed by stacking the interaction matrices obtained from each image as if they come from the same image. The proposed method is robust with respect to calibration errors but some problems can appear if the relationships between frames are not properly taken into account. They also present a new feature vector, composed of the visual features measured in pixels and the disparity of the two images of the visual stereo rig [59] where the use of 3D features allows the linearization of the interaction matrix and therefore a better joint decoupling. In [60] visual features are obtained from segmented image of objects and the stereo configuration is used for estimating 3D positions and orientations. This method suffers from coupling between orientation and translation and a better study of the influence of noise and the camera calibration is necessary to improve the robustness of the stereo visual servoing schemes proposed. In another work, Recatala *et al*. in [61] presented a stereo visual servoing system using a stereo pair of cameras mounted in an eye-in-hand configuration on a robot arm for positioning of a gripper with respect to an object. Grasp points are used as control features in the design of the control law. The epipolar geometry of two fixed cameras overlooking the scene was considered in [62, 63] for

online estimation of the image Jacobian. The stereo configuration is used to obtain the epipolar constraints or the fundamental matrix computation.

Alternatively, several solutions for vision-guided robotic inspection make use of high-cost 3D profiling cameras, scanners, sonars or combinations of them, which often results in lengthy acquisition and slow processing of massive amounts of information. However, the ever growing popularity and adoption of RGB-D sensors, especially the Microsoft Kinect sensor, motivated its introduction in the development of vision-guided robotic systems. Compared with stereo cameras and other 3D sensors, the Kinect decreases the computation time of depth, avoids complex calculations and simplifies the design effort of the controller in visual servoing systems. In the last decade, these new sensors have changed the way in which the environment can be measured and perceived by intelligent automatic systems.

Teuliere and Marchand [64] presented a 3D visual servoing technique using depth maps extracted from RGB-D sensor data to achieve robotic positioning tasks. This method does not require the estimation of the 3D pose, nor the extraction and matching of 3D features and only requires dense depth maps provided by 3D sensors. Rakprayoon *et al*. [65] proposed a Kinect-based obstacle detection for manipulator that relies on depth images. A vision-guided robotic system for tracking of a moving object using a Kinect camera was proposed in [66] where the Kinect provides 3D information of a target object and a Kalman filter is used to predict the target state (position and velocity) estimation. In another work, García *et al*. [67] presented a comparative study of the performance of D-IBVS (depth image-based visual servoing) in estimating the depth using a low cost RGB-D sensor like Kinect in three different ways. The visual servoing system has been developed over ROS (Robot Operating System) to perform visual servoing tests using RGB-D sensors. Review of literature shows that the main advantage of 3D visual servoing is that it allows the decoupling of the translational and rotational control loops. It does not require predefined geometric models and ensures the control law convergence. The authors conclude that the extra information (depth) provided by RGB-D sensors improves the positioning accuracy of the system in comparison with classic IBVS and does not require the estimation of the 3D pose as for PBVS techniques. Furthermore, the RGB-D sensor's short response time is advantageous in a variety of

applications requiring robot guidance such as control of grasping and manipulating tasks where the robot tracks trajectories for moving away from and/or moving near to the objects.

These findings inspired the selection of the Kinect sensor for the peripheral vision stage that will support offline path planning and the approach phase in the solution that is developed in this thesis. However, vision sensors are very sensitive to lighting, provide limited accuracy on depth measurements and are not able to determine physical properties of objects and interaction forces. These limitations also motivated the development of additive sensing stages and related control mechanisms that will be detailed in this thesis.

2.1.2 Force/torque Control

Force/torque sensors are critical components to extend the capability of robot manipulators, especially when there is physical interaction between the robot and the objects located in its workspace. With help from these sensors the robot is made able to deal flexibly with uncertainties in its environment and to execute complicated tasks. During interaction, force control is required to manage the forces properly [68]. In force/torque control, the force/torque sensors information is used as input for the controller. The input which consists of force/torque values applied between the object and the robot at contact points is then processed by control laws to regulate the contact force/torque for execution of the desired tasks by the robot.

Two strategies are commonly used to control interaction forces: passive and active force control. Passive force control is an open loop control system. The robot is typically equipped with a special-purpose compliant tool, designed for a particular task, to add flexibility to the end-effector. The robot does not require any sensor to measure the forces involved and the compliant tool adapts to the changes in the environment independent of the robot. However, compliant tools are not very precise or accurate and not proper to performing tasks that involve tolerance to large errors between the planned path and the actual path under compliance, as it is limited by the allowed range of compliance of the robot [69]. On the other hand, active force control works as a closed-loop control system. In active force control, the robot is equipped with a force/torque sensor that measures forces and moments and feeds these back to modify the defined path and adjust the

resulting forces. This strategy implies that the robot be compliant and responds with sufficient flexibility to the position and orientation of the contact surface to permit the adaptation without breakages. Active force control can be divided into indirect force control and direct force control.

2.1.2.1 Indirect Force Control

The indirect force control achieves force control using motion control without explicit closure of a force feedback loop [70]. Impedance control (admittance control) [52, 71-73] is an example of indirect force controller. The impedance control's objective is to regulate the mechanical impedance, defined as a dynamic relation between the exerted force and the movement error, of the robot along each direction on the task space (Figure 2.6).

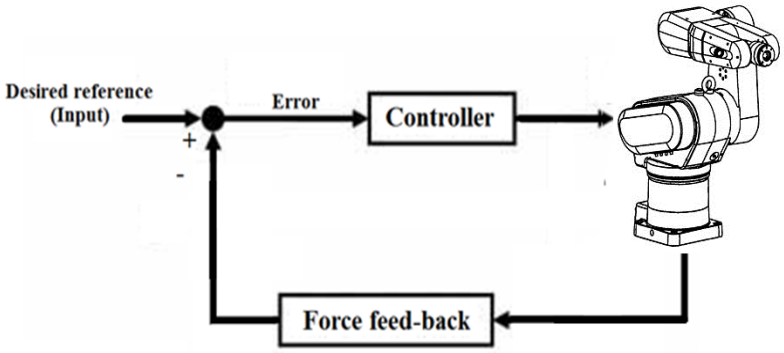


Figure 2.6: Indirect force control.

Admittance control, which corresponds to the robot reaction to motion deviation, generates a deviation from the desired motion impedance control. Usually, the desired impedance is chosen linear or of second order, and the robot and the environment are modeled as in a mass-spring-damper system with adjustable parameters (Figure 2.7).

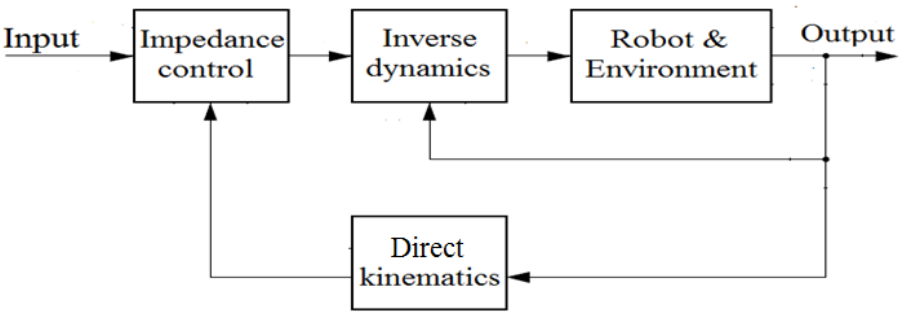


Figure 2.7: Impedance Control Scheme.

2.1.2.2 Direct Force Control

The direct force control scheme requires an explicit model of the interaction task. It continually measures the output of the system and compares the feedback to a desired motion and contact forces defined by the user under a closed force feedback loop. The error signal is fed into the controller to adjust the output until it matches the reference input and the error signal becomes zero. Hybrid force/motion control [74-78] is an example of direct force control. Hybrid force/motion control's objective is controlling the motion along the unconstrained task directions and force (moments) along the constrained task directions. According to the decomposition, the contact force and the end-effector motion can be controlled simultaneously in two mutually independent subspaces, each made of a number of directions in the workspace (Figure 2.8).

The force/torque control systems enable the robot to deal flexibly with uncertainties in the environment and adjust interaction forces between the robot and environment. However, they are only applicable in the context of surface following when there is physical interaction between the robot and the objects located in its workspace. An initial contact point between the tool and the object must be prescribed and the force controlled contact must be maintained.

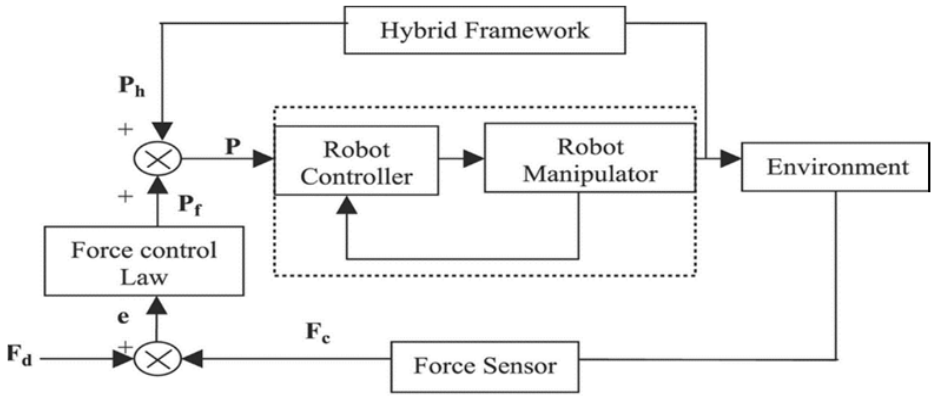


Figure 2.8: Hybrid force/motion control.

2.1.3 Tactile Sensing Based Control

Tactile (cutaneous) sensing is a more recent trend and has generally been inspired from human tactile sensing. Tactile sensors emulate the human skin by arrays of sensors which obtain distribution of the measured property over the contacting surface [79]. Tactile sensors can determine different physical properties of objects (pressure, deformation, stress, vibrations…) precisely through their contact with the world. There are various families of tactile sensors because they are implemented based on different technologies. These sensors can be classified into a number of different groups depending upon their construction: piezoresistive (optical), piezoelectric, capacitive, and elastoresistive sensors. According to the measured property, tactile sensors may also be categorized into two main groups: static and dynamic. Sensors from the former group analyze the physical properties of two static contacting surfaces based on the measurement of the normal pressure over the contacting surfaces or their deformation. On the other hand, sensors from the latter group measure the vibrations or changes in stress of two moving contacting surfaces to determine when slippage between the contacting surfaces takes place [80]. Robotic tactile sensing can be grouped into two categories as well, based on the function or the task to be accomplished: "perception for action" which is applied in grasp control and dexterous manipulation, and "action for perception" which is used in object recognition, modeling and exploration [81]. The analogous terms of extrinsic/external and intrinsic/internal touch sensing in robotics correspond to cutaneous and kinesthetic sensing in humans respectively. Intrinsic tactile sensors are placed inside the mechanical structure of the robot and derive the contact data using

force/torque sensors, while extrinsic tactile sensors are mounted at or near the contact surface of the robot to obtain the local data. Therefore, tactile control is able to adjust the way the surfaces of the robot and the object come into contact such that the relative position of these surfaces are optimal for a specific manipulation task.

Despite variations between tactile sensor types, they provide the required information at contact points between the robot and objects to adapt the robot's behavior to the objects' physical properties and to changes, such as displacement, in the environment. The main objectives of applying tactile control in robotics systems are object identification and manipulation control.

In manipulative tasks, tactile information is used as a control parameter to obtain contact point estimation, surface normal, curvature measurement, and slip detection through measurement of normal static forces [82, 83]. In [84], the direction of force has been considered in addition to magnitude to regulate the balance between normal and tangential forces to ensure grasp stability. Kane *et al*. [85] developed a novel tactile array manufactured in a silicon process, while Okamura and Cutkosky [86] used tactile sensing for detection of small surface features (cracks, bumps, and ridges) during haptic exploration and dexterous manipulation. Charlebois *et al*. [87] investigated an exploratory procedure (EP) based on rolling a probe on the surface of an unknown object for determining the local shape or curvature of objects, using contact sensing by a dexterous robotic agent. Galvez *et al*. [88] proposed an intrinsic tactile sensing method to compute the normal vectors of the surface and the shape of an object from a geometric model of the robot. In other works, different types of tactile sensors have been used for hardness or consistency measurement [89], surface texture detection [90] and temperature measurement [91]. In [92], a haptic sensor system has been developed which combines a 16x16 array of force sensing resistor (FSR) elements to recognize small-size three dimensional (3D) objects and enhance the haptic control of robotic manipulation. Dynamic events have been considered to detect stress changes, slip and other temporal events at a contact point [93]. Dahiya *et al*. [94] presented the design of a tactile sensing system while physical and operational constraints of a robotic system are taken into account. Mittendorfer and Cheng [95] developed a new generation of active tactile modules in order to approach multi-modal whole body touch sensation for humanoid robots and recently Oliveira *et*

al. [96] developed a multimodal bio-inspired tactile sensing module to provide the required data for non-normal forces estimation and surface classification by vibrations and accelerations.

Vision sensors can provide non-contact measurement of the environment but due to their limited accuracy on depth measurements and the blind spots in depth measurement in close proximity of the object (the minimum range for the Kinect is about 0.6 m) are not suitable for tasks that involve contact with object. On the other hand, touch (Force/tactile) sensing control system enables the robot to measure the local shape of objects and regulate the interaction of robots with the object that comes into contact [97]. However, they have a complex and distributed nature and require large data processing. Due to the limited bandwidth of the touch sensors, the execution speed of the task is often restrained to prevent loss of contact and information. Furthermore, they are applicable only where planned trajectories are known beforehand [81]. In order to enable the robot to interact with arbitrary objects with or without contact, in this work the sensor-based control concepts are exploited and further developed to make rigid manipulators compliant and adaptable to the changes in the environment.

2.2 Manipulator Modification and Redesign

The structure of a robot and choosing the right tools are as important as motion control and path planning. In addition to sensing improvements, the need for flexibility and reducing the risk of damages suggest complementing control systems by increasing the robot mechanical compliance, hardware modifications, retrofitting, and new components. Indeed, compliance is an increasingly desired property to provide adaptable interface between robots and the environment where there are environment (including human)-robot interactions [98]. Inspired from the human-like adaptable compliance, robots should have similar capabilities to that embedded in biological systems in order to safely interact with their environment. An intrinsic compliance can be incorporated to the robot in form of flexible links or passive compliance at the robot joints [97, 98]. Attempts have been made at constructing safe robots with variable-impedance actuation [99], parallel and distributed macro-mini actuation [102], and incorporating compliant structures into the actuators [103]. The most frequent type of compliance incorporated in robot manipulators are flexible joints [104], which allows compliance and decoupling between the actuator and the link. Flexible joints reduce the

impedance of stiff actuators but deteriorate performance in high speed tasks such as trajectory following. Variable impedance actuators were proposed [105, 106] to tackle the problem with the way that actuator's compliance varies according to the task requirements. Flexibility also can be achieved using flexible links [107, 108]. Control of flexible link manipulators arises new engineering challenges due to vibration and deformation of links that are highly nonlinear, and also reduce the positioning accuracy.

An alternative way to enable robots to safely interact with the environment and increase their compliancy is to use soft robotics [109]. The new paradigm aims to design robots that are more adaptable and flexible by replacing their rigid counterparts by soft structures [110]. Soft robots are inspired from biological systems. The design of these robots is still in early stages and not fully commercialized. The main challenge in developing soft robots is that there is no general theory and formal guidelines on how to control the unconstrained structures and design lightweight structures that bring the desired degree of flexibility and safety.

Although the proposed methods increase the compliancy and flexibility of traditional robots, their accuracy in positioning and stiffness tuning has been discarded. In order to improve the precision and stability, higher stiffness and additional sensory information is required to control the robot properly in a highly changing environment. To get a desired level of stiffness and exert suitable forces, elastic actuators were developed [111, 112]. Elastic actuators that use springs are more predictable and simpler to model and control in comparison to flexible and soft robots with complicated dynamic structures. However, using only springs for elasticity and force control can lead to oscillations in the robot's motion.

Precise force measurements is hardly a necessity for some object manipulation tasks like humans who can feel and estimate forces being applied to the body and react to them without a need for precise force measurement. Therefore, to simplify the adaptive control of robot manipulators in dynamic environments, the force control problem can be transformed into position control. For this purpose, compliant wrists were proposed [113-115]. Compliant wrists are mounted at the extremity of a manipulator robot and provide a level of compliance to the robot. A compliant wrist combines

passive compliance using elastic actuators and embedded sensors that permits active force control and compensation of positioning error. The force applied to the environment is estimated based on the amount of compression and extension of embedded springs, assuming the spring constants are known. In contrast to flexible joints and links, a compliant wrists can be easily mounted on a manipulator's end-effector and requires no modification to the existing platform, no complete re-evaluation of the control system, and reduces the complexity and costs significantly. This may lead to quick adoption by roboticists and industry. Therefore, in this work the flexibility problem is addressed by using the custom designed instrumented compliant wrist, as a one-fits-all solution to retrofit on mass market manipulators in order to provide them with a level of compliance and the necessary flexibility to deal with complex, unstructured scenarios, eventually including human beings.

2.3 Adaptive Object-Robot Interaction

Surface following, in the context of this work, is defined as a task where a robot end-effector is in contact with an object and moves along its surface while maintaining a given angle and contact force between the end-effector and the surface, and while ensuring proper coverage of a given portion of that surface. In most of the proposed methods in the literature [116-119] a predefined model of the object is used to plan a trajectory, and adaptive laws are devised based on the dynamic models of the robot systems to align the robot end-effector with the surface and adjust the contact force. However, the overall kinematics and dynamics of robot manipulators and their environment are nonlinear, therefore it is very difficult to derive them exactly, especially the latter one. To allow standard industrial robots to interact smoothly with arbitrarily shaped or dynamic objects, reliable interaction control strategy and reactive planning algorithms are required to direct the robot motion under consideration of the kinematic constraints and environment uncertainties. Although conventional (PD/PID) controllers are the most applicable controller for robot manipulator control, they are not suitable for nonlinear systems and lack robustness to disturbances and uncertainties. A general solution to this problem is to design and develop adaptive controllers. In this section, adaptive motion planning and control strategies specifically related to the robot-object interaction and surface following (or tracking) applications will be reviewed and discussed. Adaptive controllers can be divided into three categories: gain scheduling, model reference and self-tuning.

2.3.1 Gain Scheduling Adaptive Control

Gain scheduling can be considered as an open-loop or direct adaptive control system because the efficiency of the adaptation gains at modifying the system performance is not measured and fed back. The gain scheduling controller parameters are updated using an adjustment rule based on the external or internal signal value to the plant [120]. Advantage of this type of controller is its fast adaptation, reliable operation and easy implementation [121]. Most of the proposed gain scheduling controllers are developed using conventional PD/PID controllers, fuzzy logic and neural network. An adaptive PD - Gain Scheduling controller was proposed in [122] to adapt the PD parameters on-line depending on the operation conditions and to control each joint movement independently of the others for a six-degree-of-freedom robot manipulator. Helal *et al*. [123] proposed a multi-loop PID gain scheduling controller for a 2-DOF arm robot trajectory control. The fuzzy gain scheduling has been very popular for adaptive control of robot manipulators. A PID Adaptive Fuzzy Gain scheduling (PID-AFGS) was presented in [124] for intelligent control of a robot arm. In a similar work [125], a fuzzy gain scheduling is used for position control of a 2 DOF robot manipulator. In [126], an online-tuning gain scheduling dynamic neural PID (DNN-PID) controller using a neural network was developed for real-time control of a manipulator to compensate for environmental variations. In another work [127], a neural network (ANN)-based proportional-integral (PI) gain scheduling is used to tune the force controller and interact adaptively with an unknown environment.

While gain scheduling methods have been used in a variety of applications with satisfactory results, they involve a lack of relation between the dynamic characteristics of the environment and the original nonlinear plant model and can only adapt to variations which are priori known or measurable in the system. For that reason the system can fail if the relationship changes.

2.3.2 Model Reference Adaptive Control

Another popular adaptive control structure is the Model Reference Adaptive Controller (MRAC). The MRAC is used when the plant parameters are unknown or to track the output using a trajectory or command signal [128]. The structure works based on the principle of adjusting the controller parameters where the difference between the output of the actual plant and the output of the

reference model (desired performance) is measured. This information is used by an adaptation mechanism to adjust the controller parameters and force the plant to track the reference model and make the difference converge to zero. The advantage of the MRAC is that it does not require to know the plant parameters completely, here referring to the robot and its environment models. The parameters approximations can be improved during the adaptive control process [129].

The main application of the adaptive controller is in tracking a reference trajectory but it has received a considerable attention for the motion control of robotic manipulators [130]. Traditional adaptation laws for the MRAC include the MIT rule [131], Lyapunov second direct method [132], and the Popov hyperstability theory [133]. The drawback of the traditional MRAC method is that it involves heavy computational loads related to the Euler-Lagrange dynamics formulation [134]. To improve the computational efficiency, the conventional control systems are modified through employing neural networks, or fuzzy logic. Wilson and Rock [135] used two neural networks to modify the MRAC control system to a reconfigurable control system and accomplish fast adaptation. Fuzzy MRAC control systems were proposed to improve the manipulator performance interaction with the environment [136], damp the end-effector vibration and precise tip-positioning [137], and achieve rapid and accurate tracking control [138]. The MRAC drawbacks are related to the stability of the controller because it is difficult to develop stable adaptation rules and the system relies on non-linear segment cancellation [139].

2.3.3 Self -Tuning Adaptive Control

Self-tuning controllers is a class of adaptive control which is used to control unknown systems where the plant has constant (or slowly varying) parameters. The controller automatically tunes its parameters online to deliver a dynamic model of the plant at every time step [140, 124]. Self-tuning controllers offer a better performance for real-time and multi-sensory robotic applications in comparison to classic control approaches that rely heavily on analytical techniques.

To overcome the conventional controller's problem, various self-tuning adaptive controllers have been proposed during the past decades. Fuzzy self-tuning PID controllers were widely replaced and used by conventional PID controllers where the PID gains are tuned based on a fuzzy inference

system. Mohammed *et al*. [141] proposed a fuzzy system called Order-Fuzzy Proportional Integral Derivative (FO-Fuzzy-PID) for trajectory tracking control for a robot manipulator. A gain scheduling fuzzy self-tuning structure for motion control of robot manipulators was proposed in [142] and an online real-time self-tuning multi-input–multi-output (MIMO) fuzzy bang-bang controller (FBBC) was used in [143] for two-link rigid-flexible robot manipulators. Neural networks have also been applied frequently in designing adaptive self-tuning controllers for robotic manipulators motion control. Yuh [144] proposed a self-tuning type neural network controller for nonlinear time-varying robot systems. Neural network self-tuning PID control was used by Al-Khayyt [145] which combines a conventional PID controller and neural network learning capabilities for robot manipulator trajectory tracking. In a similar work [146], a neural network controller was applied to identify the system parameters and a PID controller was used to define the control law. A hybrid force and position controller based on adaptive neuro fuzzy inference system proportional derivative + integral (ANFIS-PD+I) was proposed by Chaudhary *et al*. [147] for trajectory tracking control of a robot manipulator under constrained environment.

Compared with other adaptive controllers, self-tuning controllers are more reliable in presence of variation in the system. The controllers do not require precise model of the environment (system) and can automatically tune the controller gains using the past input and output values and online measurements when the system changes. The controller's fast and reliable performance in dealing with uncertainties in the environment motivates us to develop new self-tuning controllers for adaptive position and orientation control of the robot.

2.4 Hybrid Control Strategies

To develop a comprehensive navigation system in unstructured environments it is required to combine the sensory information from various sources, integrate different control modes stably, and switch smoothly between them. For this purpose, hybrid dynamical systems are proposed which generate control laws by composition or combination of continuous and discrete signals or control modes for smooth and efficient switching between control modes and interaction with the environment.

Touch sensors are mainly used in the context of the classical constrained hybrid force/position control which requires exact knowledge of the environment in the form of constraints imposed to the end-effector motion. In the other words, the geometry of the environment is assumed to be known, but its position and orientation with respect to the robot end-effector are unknown. In [148], the problem of force and position control of a robot manipulator performing compliant tasks is addressed which describes a 3D surface tracking controller based on the Smith predictor design. An adaptive motion and force tracking control for high performance robust control of a manipulator in contact with unknown stiff surfaces was proposed in [149]. Automation of surface finishing processes such as deburring, grinding, chamfering, and polishing has been an interesting application of compliant force/motion control in the manufacturing industry. A complete surface finishing task presented in [150] is divided into three phases (free motion phase, transition phase, and constrained motion phase). Motion/force control algorithms are developed for each phase depending on the location of the robot end-effector. In [151], a hybrid form of natural admittance control is used to address the surface finishing challenges. The hybrid motion control consists of position control using automated registration of the physical part with a CAD representation, and force control to exert controlled and gentle pressure on the part's surface. Yin *et al*. [152] proposed a methodology for active tracking of unknown surface using force sensing and control technique. They present a strategy of force sensing and control on the basis of information fusion of force and position, fuzzy hierarchical coordination and neural control. In a similar work, Li and Li [153] applied a neural network to classify an unknown environment. On-line force feedback data are employed to estimate the normal and tangential directions of the unknown environment to generate the reference trajectory for the target impedance model. The curvature is calculated in real time to adjust the speed of the tangential direction by fuzzy reasoning according to current and forecast contact force. Papageorgiou *et al*. [154] proposed a methodology to drive the end-effector of a non-redundant manipulator in close proximity of a surface while avoiding obstacles. Once the end-effector is close to the surface, a second controller takes over to stabilize the end-effector at a predefined distance to the surface. Later, in [155] they added force control for tasks that involve dealing with contact on surfaces. The problem of kinematic input constraints and a non-smooth kinematic controller was discussed in [156]. Wang and Li [157] presented an approach to control a manipulator while tracking a surface where the tactile sensing is integrated with force-torque

information as the feedback. Hybrid impedance control is applied to follow both the position and force trajectories. Later, they extended the work to control the end-effector of a redundant manipulator in [158].

A fundamental advantage of using vision-based control over force/torque control is that no contact with the object is required as it allows non-contact measurement of the environment. Vision sensors provide global or local information to determine the robot and the objects relative locations in the environment. This information can be used by the robot controller to reach to the object or target while avoiding undesired obstacles. However, most of the proposed approaches on vision control for robot manipulators have involved free motion control [159]. These approaches demonstrate a good performance in dealing with planar surface following but are not very successful over 3D structures because the depth of the object is also required for precise path planning. Therefore, hybrid vision/force controllers have been proposed to deal with cases where free motion and contact tasks are involved. Vision and force are two complementary sensing capabilities that can greatly improve the autonomy of a robot manipulator performing interaction with its environment.

In recent years, several approaches have been proposed where force and vision measurements are combined as hybrid visual/force control. In fact, the robot achieves global/local information about the environment using vision and can later adjust its motion with respect to the local constraints by perception of the force applied to the end-effector. Pichler and Jagersand [160] presented an uncalibrated hybrid force-vision control system where force sensing is incorporated into an uncalibrated visual servoing system to simultaneously update a surface model while controlling the manipulator. In a similar work [161], an eye-in-hand vision and a force control is integrated based on the task frame formalism (TFF) in an uncalibrated workspace, and in [162] a force-torque sensor mounted at the wrist of the manipulator and a visual sensor with an uncalibrated single camera fixed to the ceiling of the work space are used to control the manipulator, such that the end-effector follows a path on an unknown surface. Olsson *et al*. [163] proposed another uncalibrated visual servoing strategy for surface following using multiple cameras. An edge-based rigid-body tracker is used in an observer-based controller, and combined with a six degree-of-freedom force/impedance controller for interaction with a stiff uncalibrated environment. Ramey

et al. [164] presented a real-time 3D surface tracking using a stream of rectified stereo pairs based on the iterative updating of surface representations directly from the images and a re-weighted least squares minimization wherein a mask is incorporated to increase robustness to occlusion information. The surface tracking method is formulated for a general family of linear parametric surface models and the cases of planar and tensor product surfaces. An adaptive hybrid visual servoing/force control was proposed by Hosoda *et al*. [165] who presented an online estimator for an image Jacobian matrix which describes the relation between image features and the tip position/orientation of the manipulator, and a method to estimate the normal vector of the unknown constraint surface. Similarly, in [166], a visually servoed adaptive controller is proposed for motion and force tracking with uncertainties in the constraint surface, kinematics, dynamics, and camera model. Adaptive Jacobian matrices are used to compensate the uncertainties due to internal and external parameters and a Lyapunov-like function is presented to prove the stability of the proposed vision–force controller. Lippiello *et al*. [167] proposed a robot force/position control approach with force and visual feedback where the environment is a rigid object of known geometry but with unknown and time-varying pose. The relative pose is estimated online based on all the available sensor data (visual, force and joint position measurements) using the Extended Kalman Filter (EKF).

Although the proposed hybrid vision/force approaches enhanced the robot's motion control, they are still slow and require a considerable amount of time due to the limited contact points of force and tactile sensors which impose delays. The approaches are not proper to perform surface following tasks when there is a large error between the planned path and final path and also where close proximity (non-contact or prior to contact) interaction or surface following is desired. Furthermore, the majority of the cases estimate the 3D parameter of the target object from the extracted image data obtained from a 2D camera or 3D scanner (laser sensor) system which is usually time-consuming and require extensive data processing.

2.5 Chapter Summary

This chapter presented a detailed review on the sensors, approaches and control strategies that have been proposed to control the movements of robot manipulators using the information collected by

various sensors. As vision, force and tactile sensors are the most commonly used devices to analyze the robot's interaction with the environment, first their corresponding control strategies have been exposed in section 2.1, for visual servoing, force, and tactile sensing-based control.

A great number of publications demonstrate that visual servoing is and will be the most widespread and efficient method for manipulator control. Particularly, position-based, image-based and stereo/RGB-D (3D) visual control techniques are also detailed in order to further describe various forms of visual servoing. However, the review of literature showed that vision information only is not sufficient and accurate enough to perform tasks that require close and physical interaction with objects due to occlusion issues, lightening sensitivity and limitation in available density or resolution on depth measurements. To overcome the problem, touch sensors (force/torque and tactile) are often incorporated to the system to provide additional sensory information. Two different techniques have been presented for implementing force control: passive force control and active force control. Similarly, tactile control techniques have been categorized according to their application in robotic tasks: object identification and manipulation control.

Using various sensors and applying proper control strategies enables the robot to perceive its environment and interact with objects in static environments. However, beyond selecting proper sensors, and in order to perform complex tasks in the presence of uncertainties in unstructured environments, it is also required to design robots and develop control strategies that are more adaptable and flexible. Approaches from the literature that are most related to this thesis have been overviewed, with their respective sensory configuration and the final application for each control technique compared. This review reveals that adaptive controllers and hybrid control systems offer a better performance in dealing with uncertainties in unstructured environments. As such, vision measurements are usually combined with force/tactile measurements to form hybrid visual/force control for robotic manipulation.

Although these approaches demonstrate good performance, they are only applicable where there is a physical interaction between the robot and the objects, and require precise models of the robot and the environment, with the latter being extremely hard to obtain. To allow these robotic

platforms to interact safely with transforming or moving objects, a reliable interaction control strategy and trajectory planning algorithm is required to direct the robot motion under consideration of the kinematic constraints and environment uncertainties. A hybrid multi-sensory system is expected to be able to control the robot kinematics and adapt its behavior to dynamic situations which have an important influence on the reliable and smooth interaction between the robot and the object. Therefore, this work proposes a hybrid-adaptive switched control system that consists of two model-free adaptive controllers for flexible and reactive interaction with transformable objects of different shapes with only the addition of an instrumented wrist device to prevent any damage and ensure finely controlled contact using multimodal sensory information.

Chapter 3. PROPOSED APPROACH

In order to enable industrial robots to interact with objects of different shapes and accommodate their curvature in a dynamic and unstructured environment, they are expected to behave autonomously, adapt to changes in the environment and react to the objects movement and deformation safely and flexibly. A traditional way to interact with and explore objects with a robotic manipulator is to use geometric models of the objects. However, this requires that exact models of the objects are acquired and computed in order to closely follow the surface of these objects using conventional techniques (deliberative control). Deliberative (model-based) planning searches for the optimal path and generates a plan to reach the goal, assuming comprehensive and accurate knowledge of the environment. Then the system performs an action within the context and limitations of the static model of the environment and that supports the planning stage. The main drawback of deliberative control is that the system cannot adapt to new scenarios. To deal with the shortcomings of the deliberative approach, adaptive (reactive) planning was developed. Adaptive motion planning methods generate control commands based on currently perceived information. As such it is not required to build a complete model of objects. However, when no prior knowledge about the object is available, this may result in following a wrong direction, especially at corners and on curved surfaces. To achieve close surface following with contact, an initial contact point between the robot tool and the object must also be prescribed by the user, which introduces a supplementary constraint. Addressing these limitations, this work proposes an original hybrid-adaptive motion planning approach that integrates vision, proximity and contact sensing in a hybrid control structure which combines the advantages of planning in deliberative architectures along with the quick response of reactive architectures.

3.1 Multi- Stage Control System Overview

Surface following and interaction with objects can be accomplished either with contact (for applications such as polishing, welding, cleaning, or particles collection which is of particular interest in this work) or without contact (for applications like painting and inspection). In this work, to guide the robot's end-effector towards an object, and then control its motion to execute the desired interaction with the surface, the motion control task is divided into four interaction modes (Figure 3.1): three main motion modes (free, proximity, contact) and one transition mode

in between the main modes (blend), where each motion mode uses specific sensory information. Figure 3.1 shows how the proposed multi-modal sensing strategy is taken advantage of to develop a multi-stage control approach. In order to efficiently navigate a robotic manipulator and interact with objects of various but unmodeled shapes, the robot movement over the objects surface is controlled using three motion planning techniques: deliberative, reactive and a combination of those (hybrid). To design the deliberative system and generate an offline trajectory, a global but approximate model of objects is acquired by a RGB-D sensor, namely a Kinect sensor (Appendix A), to ensure complete coverage of the object's surface which is explored.

Adaptive (reactive) control system is developed by integrating in different stages the proximity and contact sensory information that is provided online by a specially designed compliant wrist with embedded sensors, which was developed by other researchers prior to this work (Appendix B). The adaptive control stage allows for refinement and compensation of the low accuracy of the RGB-D data collected by the Kinect sensor when close and physical interaction between the robot and the object is required. For this purpose, two unique model-free and self-tuning adaptive controllers are designed to control the robot position and orientation using the available online sensory information. The proposed adaptive controllers do not require precise mathematical model of the robot and the environment, or any force/torque calculation and learning procedure, which enables the robot to perform reactive and fast interaction with or without contact with a surface.

To achieve a seamless and efficient combination and integration of the deliberative (offline) and reactive (online) control systems, a stable and reliable switching scheme is also presented. It is well known that classic hard switching (event-based) schemes can cause chattering, oscillation and instability if a subsystem fails due to noise or other possible sensor failures. To address the problem, a hybrid switching scheme is proposed that allows data fusion from multiple sensing modalities, and provides the manipulator with the capability to operate independently on any given mode, or in transition in between control modes to perform more complicated tasks as required by different applications. Unlike previous works, the proposed hybrid switched control system employs a supervisory (decision making) module that uses hard and blend switches for smooth

transitions and weighted combinations of the different motion modes to achieve smooth but accurate robot-object interactions in proximity as well as in contact with surfaces.

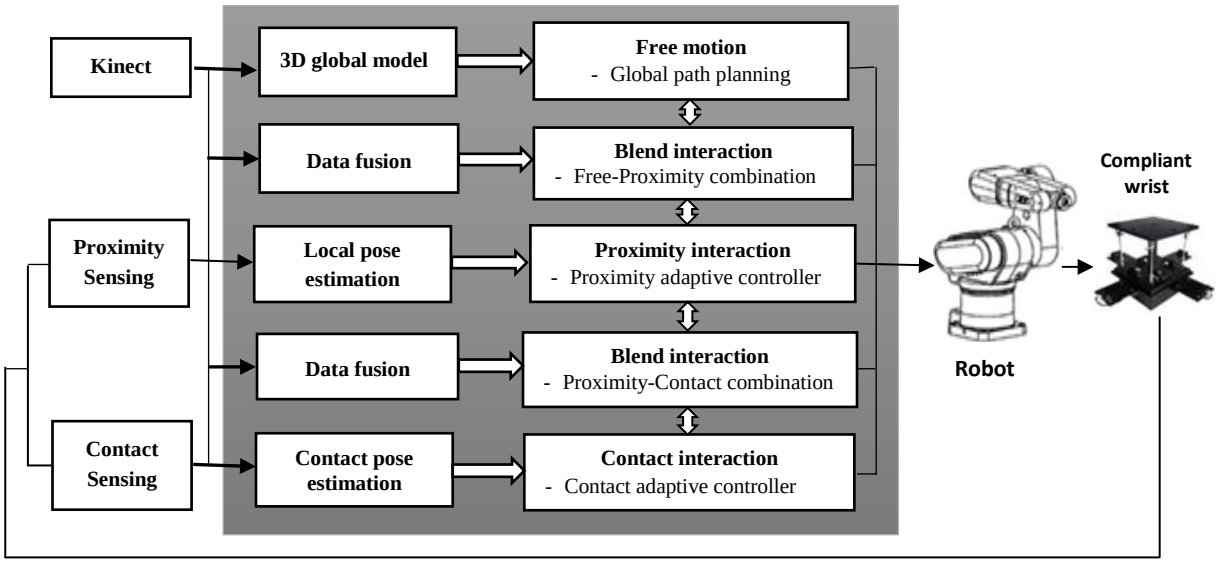


Figure 3.1: Proposed hybrid-adaptive control system.

The main components and interconnections, as well as how the control modes interact with each other is shown in Figure 3.2. The first step to achieve interaction with an object is the detection and localization of that object in the robot workspace. For that matter, the Kinect sensor is used for rapidly acquiring color and 3D data on the environment. The data is then processed to segment and construct a 3D model of the object of interest in the scene. In order to completely explore the object, a novel coverage path planning technique is proposed to generate a primary (offline) trajectory as a general guidance for navigation and interaction with the object (*free motion*). The acquisition speed of the Kinect and its low cost have been major selection criteria for this sensor to be used for offline trajectory planning. However, the information provided by the Kinect is not accurate enough for developing reactive object-robot interaction control when close and physical interaction is required. Furthermore, the object may move or deform under the influence of the physical interaction. Therefore, proximity control and contact control modes are also designed and integrated with the system to dynamically update the 3D model through using a custom designed instrumented compliant wrist (Appendix B). This stage supports the modification of the position and orientation of the robot based on the online sensory information when it reaches in close proximity where the compliant wrist embedded sensors can detect the object (*proximity mode*) or

in contact with the object where the robot makes contact with the object (*contact mode*). In order to switch and transit smoothly between the interaction modes, blend motion modes are proposed that allow data fusion from multiple sensing modalities by combining pairs of the main motion modes. The design, implementation and integration of the system components are detailed in the following sections.

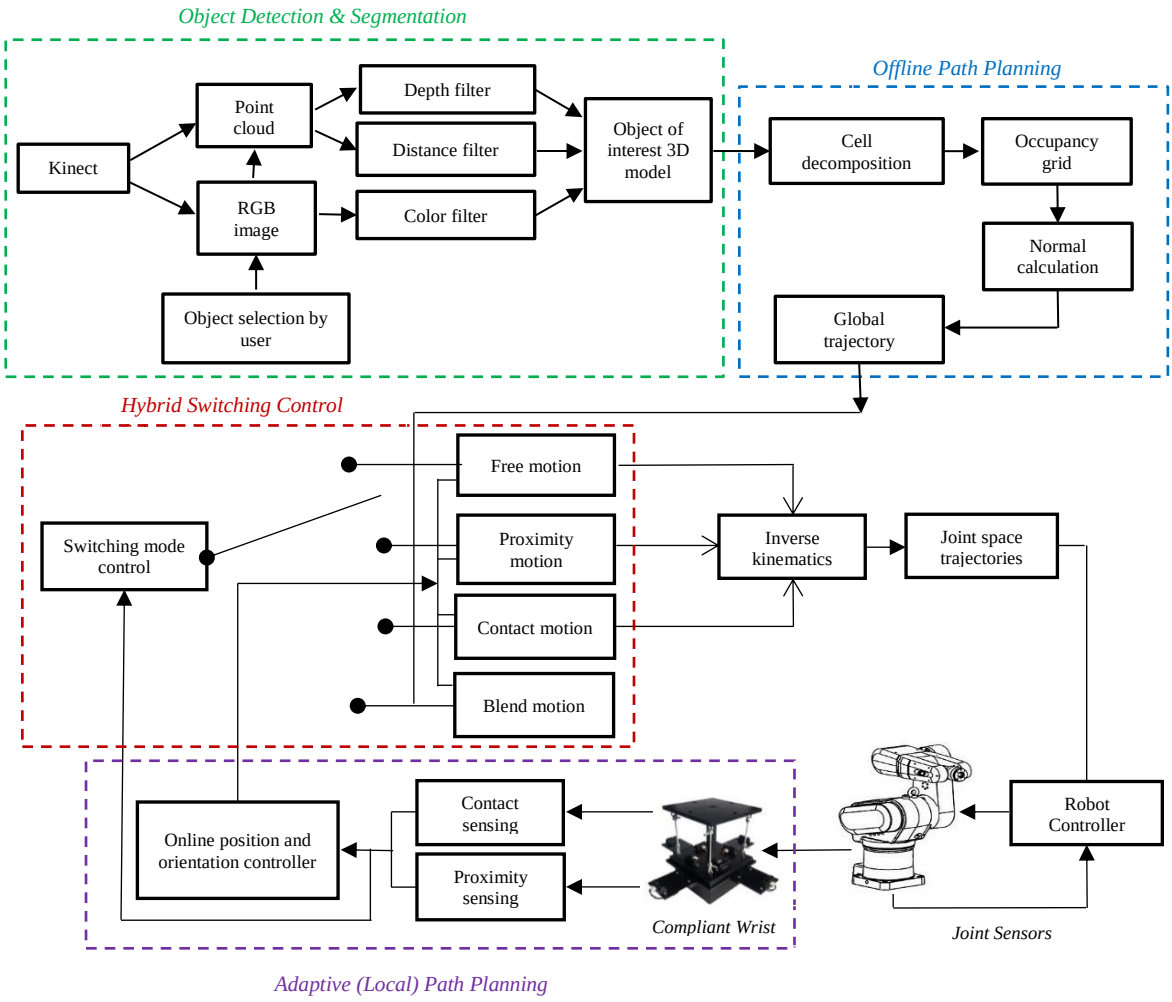


Figure 3.2: Block diagram of the proposed hybrid switching control system.

3.2 Offline (free motion) Trajectory Planning

In the free motion phase which refers to the robot movement in an unconstrained work space, an approximate 3D model of the object of interest is constructed using 3D data collected by a Kinect

sensor. The 3D model is used to localize the object in the robot workspace and generate a global trajectory to guide the robot towards the selected surface and to scan and explore the entire object's surface that is visible by the Kinect and reachable by the robot. Based on the surface shape acquired by the peripheral vision stage, a unique coverage path planning (CPP) strategy is developed (detailed in Section 3.2.2) using dynamic 3D occupancy grid to plan a global trajectory, ensure complete coverage of the selected surface, and support the early robotic exploration stage of the object's surface.

3.2.1 Object Detection and Segmentation

In order to efficiently navigate a robotic manipulator and interact with objects of various shapes, identification and localization of the object of interest in the robot workspace must first be achieved. For that matter, the Kinect sensor is positioned behind the robot with the sensor's viewing axis (Z-axis) perpendicular to the bodywork for collecting data over the object, as shown in Figure 3.3.

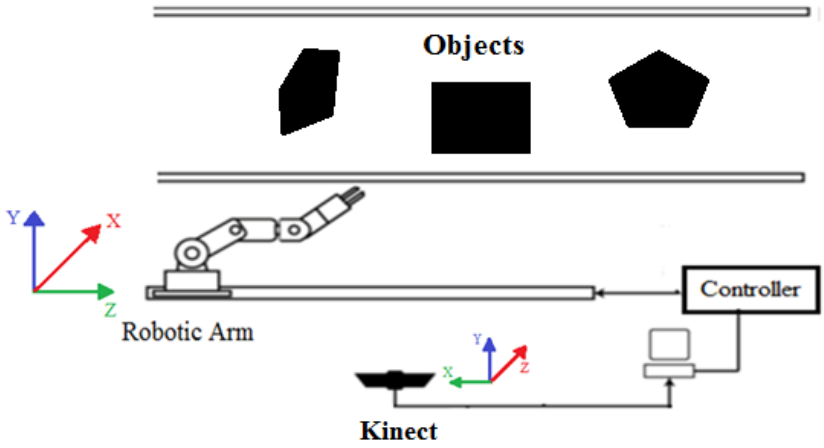


Figure 3.3: The RGB-D sensor positioning.

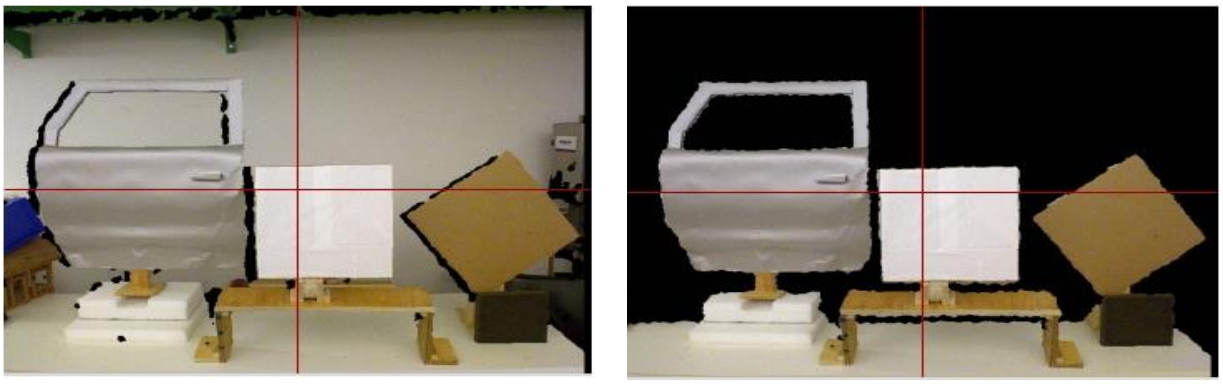
The data acquired by the Kinect sensor contains an RGB image and a collection of points ($PointCloud$). Each 3D point, p_i, with (x, y, z) coordinates is defined with respect to the Kinect reference frame and supports corresponding (r, g, b) values representing color (Eq. 3.1).

$$PointCloud = \bigcup_{i=0}^{\nu-1} p_i \,, \forall i, j \in [0, \nu - 1], p_i \cap p_j = \emptyset$$

$$p_i = [p_{ix}, p_{iy}, p_{iz}, p_{ir}, p_{ig}, p_{ib}] \tag{3.1}$$

where ν is the number of points in the point cloud.

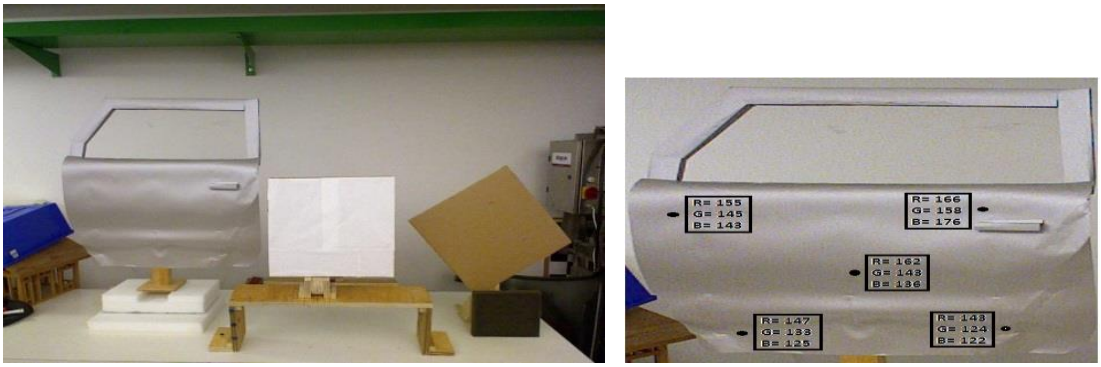
When multiple objects are present in the working environment, objects of interest must be identified and discriminated from the scene. Three filters are used to extract and segment an object of interest from the workspace: a depth filter, a color filter, and a distance filter. The depth data range covered by the Kinect sensor is typically between 0.5m to 6m and the error in depth considerably rises with the distance [168, 169]. To filter out the points that are not in the desired depth range (Dz), a thresholding filter is applied (see algorithm 1) which eliminates all the points with a depth value (Z-coordinate) over a pre-set desired range. As shown in Figure 3.4, the points in the original depth image acquired by the Kinect which are located outside of the desired range are filtered out, here represented as black pixels. Therefore, the less accurate background area is discarded and the depth range is limited to the actual robot workspace. Color segmentation allows to group objects of similar color. However, if there are multiple objects of similar color in the scene, a single object of interest cannot be identified only using color segmentation. Therefore, a distance filter is also applied on the point cloud to extract only the object of interest, as will be detailed below.



(a) (b)

Figure 3.4: Depth filter: a) original data from Kinect, and b) after applying the depth filter.

In order to optimize the robot movement, specific information about the object of interest should be extracted from the RGB-D data using color and distance filters. After acquiring the data, the RGB image (Figure 3.5a) is presented to the operator. A mouse event callback function waits for the operator to click on any points of the desired object in the image from which the RGB values of the pixels (Figure 3.5b) are extracted. To increase the accuracy of the color segmentation, the operator can choose a number of points (μ) on the object of interest according to the object's size and shape. The RGB values of the operator selected points of interest, *ObjectRGB*, as defined in Eq. 3.2 are first used to define a range of color for the object. The points in the RGB image that are within the same color range and for which the Euclidean color distance (ΔE) from the selected points (*Object RGB*) is less than a specific pre-set threshold (λ) are extracted from the point cloud, while the other points are eliminated.



(a) (b)

Figure 3.5: a) RGB image of the working environment, and b) RGB values corresponding to the selected points over the user-selected object of interest.

Since each pixel in the RGB image corresponds to a particular point in the RGB-D point cloud, a primary estimate of the object of interest's position in the workspace is extracted from the points corresponding to the points selected by the operator. After RGB color-based clustering has been applied, the (x,y,z) coordinates of the selected points, *ObjectCoordinate*, as defined in Eq. 3.3, are used as query points to filter out even more points that are not within a desired 3D Euclidean distance range with respect to the query point.

$$ObjectRGB = [\alpha_k, \beta_k, \gamma_k], k \in [0, \mu - 1] \quad (3.2)$$

$$ObjectCoordinate = [\mathcal{X}_k, \mathcal{Y}_k, \mathcal{Z}_k], k \in [0, \mu - 1] \quad (3.3)$$

where α, β, γ correspond to RGB values, respectively, for the color space used in this work.

The distance of each point in the point cloud from each query point is computed and the points for which the distance deviation ($\Delta£$) with respect to the query points values are above the pre-set threshold [£] are dropped. The value of £ (Eq. 3.4) is the minimum distance between each query point from the other query points in the *ObjectCoordinate* set. Eventually, the remaining set of points (O_i) form the object of interest RGB-D point cloud, $ObjCloud^{Kinect}$, as defined in Eq. 3.5.

$$\boldsymbol{for}\ (k, q = 0 \ldots \mu - 1)$$
$$£_k = \min \sqrt{(\mathcal{X}_q - \mathcal{X}_k)^2 + (\mathcal{Y}_q - \mathcal{Y}_k)^2 + (\mathcal{Z}_q - \mathcal{Z}_k)^2} \quad (3.4)$$

where $[\mathcal{X}_q, \mathcal{Y}_q, \mathcal{Z}_q]$ are the coordinates of the q[th] query point that belongs to $[0, \mu - 1]$, q is not equal to k, and μ is the number of selected points by the operator (query points).

$$ObjCloud^{\boldsymbol{Kinect}} = O^K = \bigcup_{i=0}^{m-1} O_i, O_i \in [0, m - 1] \quad (3.5)$$

where m is number of extracted points and $O_i = [o_{ix}, o_{iy}, o_{iz}]$.

However, the points are defined with respect to the Kinect reference frame and must be transformed to the robot's base frame (Eq. 3.6).

$$ObjCloud^{\boldsymbol{Robot}} = O^R = T_K^R . O^K \quad (3.6)$$

where T_K^R is the homogeneous transformation matrix of the Kinect sensor's frame with respect to the robot's base frame. The transformation matrix is estimated using the method proposed in [170, 171]. Algorithm 1 details the entire object of interest's segmentation process.

Algorithm 1: Object of interest segmentation

1: **Inputs:** $PointCloud = [p_i] = [p_0, p_1 \dots p_{v-1}], ObjRGB = [\alpha_k, \beta_k, \gamma_k],$
$ObjCoordinate = [\mathcal{X}_k, \mathcal{Y}_k, \mathcal{Z}_k], k \in [0, \mu - 1], T_K^R$

2: **Output:** $ObjCloud^{\boldsymbol{Robot}} = O_i^R$

3: **Parameters:** Dz, ν, μ, £, λ.

4: $\boldsymbol{for}\ (k = 0 \dots \mu - 1)$

5: $\boldsymbol{for}\ (i = 0 \dots \nu - 1)$ **do**

6: $\boldsymbol{\Delta Z_k} = [\mathrm{Dz} - p_{iz}]$

7: $\boldsymbol{\Delta E_k} = \sqrt{(\alpha_k - p_{ir})^2 + (\beta_k - p_{ig})^2 + (\gamma_k - p_{ib})^2}$

8: $\boldsymbol{\Delta £_k} = \sqrt{(\mathcal{X}_k - p_{ix})^2 + (\mathcal{Y}_k - p_{iy})^2 + (\mathcal{Z}_k - p_{iz})^2}$

9: $\boldsymbol{if}\ (\Delta Z_k \geq 0\ \&\ \Delta E_k \leq \lambda\ \&\ \Delta £_k \leq £_k)$ **then**

10: $O^K[i] = p_i$

11: $\boldsymbol{end\ if}$

12: $\boldsymbol{end\ for}$

13: $\boldsymbol{end\ for}$

14: $O^R = T_K^R . O^K$

15: $\boldsymbol{return}\ O^R$

3.2.2 Surface Coverage Path Planning

Based on the object of interest's 3D data extracted from the original depth image, a coverage path planning (CPP) strategy is developed using dynamic 3D occupancy grid to plan a global trajectory and guide the robot end-effector towards the surface, and then control its motion to execute the desired exploration and interaction with the surface.

The key task of coverage path planning (CPP) is to guide the robot over the object surface and guarantee a complete coverage using the preprocessed RGB-D data from which only the selected object of interest remains. CPP for industrial robotic application has recently undergone significant research especially in automated inspection, security screening [172], painting and surface

following tasks [173]. The CPP algorithms can be performed either off-line when the environment (object) is static, which requires a priori knowledge of the environment, or on-line when the environment might change from when it was acquired, in which case real-time sensory information is required to update the information and produce online trajectory [174]. Occupancy grids are one of the most popular modeling techniques used to support CPP. In most of the previously proposed methods [175-176], the occupancy grid is built in an offline phase and it is not updated during operation. In contrast to previous works, here the CPP is developed using 3D occupancy grid method which utilizes offline and online trajectory planning. To support the offline part of the planning, an occupancy grid is initially formed using the information provided by the Kinect sensor. Once the robot reaches in proximity to the surface, online planning is performed with the occupancy grid being dynamically refined, using the data acquired by sensors built in the compliant wrist to modify the position and orientation of the robot when it is in close proximity (proximity interaction) and in contact with the object (contact interaction). The process continues during the entire robot-surface interaction.

3.2.2.1 3D Occupancy Grid

The construction of an occupancy grid to represent an object starts with a cubic bounding box surrounding the entire object which gets recursively subdivided into smaller cells until the maximum level of resolution is reached. Here, the volume of the bounding box is dynamic according to the volume of the object of interest (Eq. 3.7).

$$(Z_{max} - Z_{min}) \times (X_{max} - X_{min}) \times (Y_{max} - Y_{min}) \tag{3.7}$$

where Y_{min}, Y_{max}, X_{min}, X_{max}, Z_{min} and Z_{max} correspond respectively to the minimum and maximum values of the X, Y and Z coordinates, in millimeters, in the point cloud of the object of interest, $ObjectCloud^{Robot}$, obtained in Eq. 3.6. The size of the cells can be selected based on the desired resolution of the grid for a particular application and then the object surface is discretized to a set of uniform cubic cells (Eq. 3.8). Given the application of surface following for particles collection that is considered here, the object of interest is mapped into an occupancy grid with cubic cells of the same size as the robot's tool plate size to ensure complete coverage with minimum number of

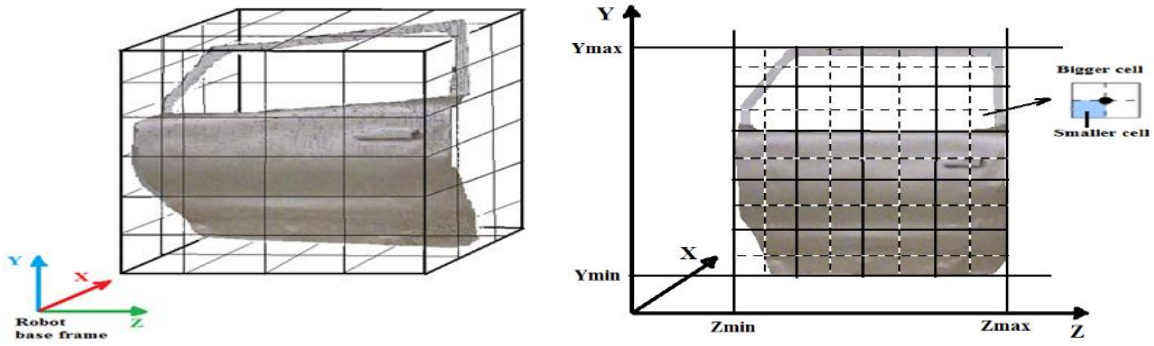
moving steps. In the present case the tool plate measures $130^{mm} \times 130^{mm} \times 130^{mm}$ (Figure 3.6a).

$$Obj = \bigcup_{I=0}^{H} \bigcup_{J=0}^{V} \bigcup_{K=0}^{P} C_{I,J,K}\,, I \in [0, H], J \in [0, V], K \in [0, P] \tag{3.8}$$

where $C_{I,J,K}$ represents a cell in the object model and

$$\begin{aligned} P &= ||(X_{max} - X_{min})/130|| \\ H &= ||(Y_{max} - Y_{min})/130|| \\ V &= ||(Z_{max} - Z_{min})/130|| \end{aligned} \tag{3.9}$$

The level of resolution has an important impact on the precision of the object borders in the model. The major issue arises when cells are partially occupied. That usually happens where the object has acute edges and also near the contours of the object of interest. In order to mitigate the problem, the grid resolution is increased and each cube is subdivided to 8 smaller cells (Figure 3.6b). The smaller cells (C^S) are cubes with size of $65 \times 65 \times 65$, in mm, with each larger cell (C^B) containing 8 smaller cells in two layers of four.



(a) (b)

Figure 3.6: a) Object of interest 3D model, and b) object discretized to a set of uniform cubic cells.

The resolution and therefore the precision of the occupancy grid can be increased by changing the maximum number of recursive subdivisions, but it requires a larger memory space to store the model. Considering that the RGB-D sensor is perceiving the scene from a point of view that is

relatively perpendicular to the surface of the object, the regions in the back are essentially occluded, actually creating only a 2.5D model of the object of interest. Advantage is taken of the fixed configuration of the Kinect sensors to make the model more compact and achieve higher computational efficiency by decreasing the number of cells in the model. For that purpose, the matching front and the back layers of the larger cells (Figure 3.7a) are merged into one layer, and the resulting pairs of smaller cells are replaced with a cuboid cell of size $130 \times 65 \times 65$, in mm, the largest side being on depth, here corresponding to the X-axis in the robot's reference frame as depicted in Figure 3.7b. Each larger cell now contains only four smaller cells, instead of eight (Figure 3.7b), and the model being considered is a 2.5D surface representation from the point of view of the Kinect sensor.

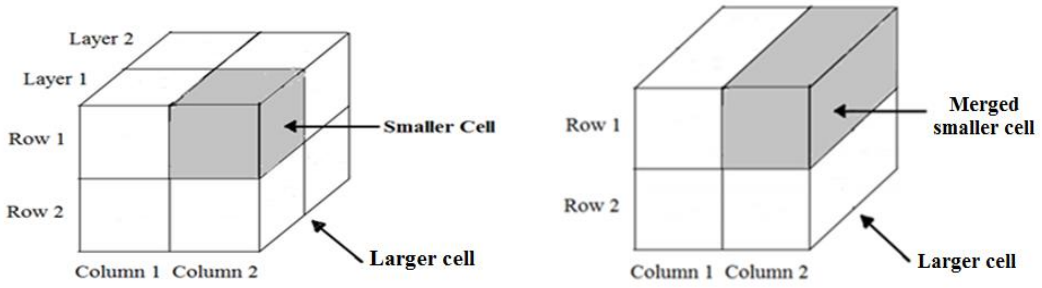


Figure 3.7: a) Smaller cell and larger cell representation, and b) merging smaller cells to reduce data volume.

The occupancy grid is formed by determining the state of each cell. The smaller cuboid cells ($C^S_{i,j,k}$) are classified with two states as occupied or free. A cell is tagged as occupied if it contains at least one vertex from the point cloud representing the object of interest, $ObjCloud^{Robot}$ (Eq. 3.6). If the smaller cuboid cell is occupied, its value is 1, otherwise it is 0. The state variables associated with the larger cells ($C^B_{I,J,K}$) are: occupied (1), unknown (0.5), or free (0), which are identified based on the smaller cell's value. If more than two smaller cells of a larger cell are occupied, the value of the associated larger cell is 1; else if one or two of the small cells are occupied the value is 0.5; and if all the small cells are free then the larger cell is tagged with 0.

$$C^S_{i,j,k} = S(0 = Free|1 = Occupied)$$
$$C^B_{I,J,K} = S(0 = Free|0.5 = unknown|1 = Occupied)$$

The occupancy grid corresponding to the larger cells is represented by a H × V × P dimensional matrix (Eq. 3.10). Initially all cells are considered as unknown. Therefore, the occupancy matrix, $OccMatrix$, is initialized with a value equal to 0.5 in all cells, where $C^B_{I,J,K}$ contains occupancy information associated with the larger cell *(I, J, K)* and cell (0, 0, 0) is located in the front upper left corner of the grid.

$$\text{OccMatrix} = \begin{bmatrix} \begin{bmatrix} C^B_{0,0,0} & C^B_{0,1,0} & \cdots & C^B_{0,V,0} \\ C^B_{1,0,0} & C^B_{1,1,0} & \cdots & C^B_{1,V,0} \\ \vdots & \vdots & \ddots & \vdots \\ C^B_{H,0,0} & C^B_{H,1,0} & \cdots & C^B_{H,V,0} \end{bmatrix} & \begin{bmatrix} C^B_{0,V,1} \\ \vdots \\ C^B_{1,V,1} \\ \vdots \\ C^B_{H,V,1} \end{bmatrix} \\ \begin{bmatrix} C^B_{0,0,P} & C^B_{0,1,P} & \cdots & C^B_{0,V,P} \\ C^B_{0,0,1} & C^B_{0,1,1} & \cdots & C^B_{0,V,1} \end{bmatrix} & \begin{bmatrix} C^B_{1,V,P} \\ \vdots \\ C^B_{H,V,P} \end{bmatrix} \end{bmatrix} \tag{3.10}$$

The occupancy matrix provides the required information to plan an offline trajectory to completely explore the selected object's surface. Therefore, in the next step the occupancy matrix is used for global path planning. In addition, to preserve the dynamic characteristic of the occupancy grid and to achieve a more accurate and efficient exploration of the object surface, the occupancy grid is updated later, using the instrumented compliant wrist sensory information when the robot reaches in close proximity to the object.

3.2.2.2 Global Trajectory Generation

In order to ensure coverage over the object surface, the robot end-effector should explore all the occupied cells and avoid the free ones. Therefore, the robot end-effector's pose at each time step is defined by a set of points:

$$P_G[k] = (pos_G[k], r_G^{\Phi}[k], dir_G[k]) \tag{3.11}$$

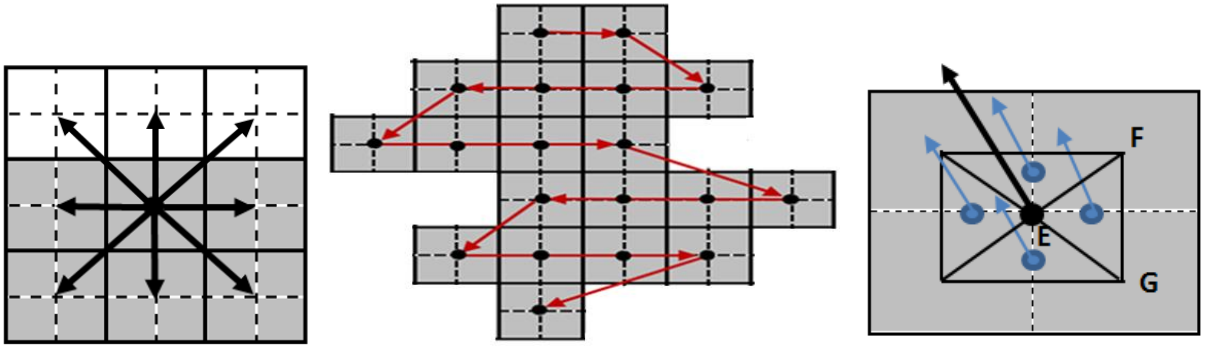
where $pos_G = [\, p_G^x, p_G^y, p_G^z]$, $r_G^{\Phi} = [\, r_k^{\theta}, r_k^{\varphi}, r_k^{\psi}]$, and dir_G respectively determine the end-effector position with respect to the robot base, its orientation, and the direction of motion. The

concatenation over iterations, k, of the set of points to be visited generates a global trajectory for the robot end-effector to follow.

$$Trajectory_{Global} = P_G[0] \frown P_G[1] \frown P_G[2] \frown \cdots \frown P_G[occ] \tag{3.12}$$

where occ is the number of occupied cells in the occupancy grid.

In this work, the start point is always the nearest occupied cell to the upper left corner of the object and the initial moving direction is horizontal. The end-effector position at each moving step ($pos_G[k]$) is defined by extracting the closest vertex in the object point cloud ($ObjectCloud^{\boldsymbol{Robot}}$) to the center of the occupied larger cell ($centre(C^B_{I,J,K})$) in the robot moving direction. However, if the larger cell is not completely occupied ($i.e.\, C^B_{I,J,K} = 0.5$). As shown in Figure 3.8a, the robot can move in 8 possible directions [*horizontally, vertically, and diagonally*] and it must move through all the points to cover the object surface area. This area is covered using a Seed Spreader motion [177] pattern (Figure 3.8b). The robot begins at a start point defined above and moves horizontally to the right towards the center of the next occupied cell. If there is no occupied cell on the right, the motion direction is changed and the robot moves vertically (diagonally) down/up to the next point. Then the robot again moves horizontally but in the opposite direction until it reaches to the last occupied cell on the other side of the object. The process continues until the robot has conquered the last occupied cell and explored the entire area.



(a) (b) (c)

Figure 3.8: a) Accessible directions of motion for the robot end-effector over the surface encoded in the 2.5D occupancy grid model, b) global trajectory planning strategy, and c) vertex normal calculation at the center of the front face of a larger cell.

The set of points (center of the occupied cells), forming a path defined in the previous step, determine the positions for the end-effector in Cartesian coordinates over the object surface. In addition to the robot position, to ensure proper alignment of the end-effector with the surface, it is required to estimate the local surface normal and calculate the end-effector orientation at each point. The local normal at each point is calculated using the neighbor cells (Figure 3.8c). Given that there are four smaller cells forming a larger cell, four points which are situated in the center of each occupied cell (closest point to the center) in three dimensions are obtained. Each set of three points forms a triangle for a total of four triangles that can be generated. First, the normal to each triangle (N_i) is calculated and normalized. The normal of each triangle is computed as the cross product between the vectors representing two sides of the triangle. The following equations define the normal vector, ***N***, calculated from a set of three points, *E*, *F*, *G*, coordinates:

$$N_x = (E_y - G_y \cdot E_z - F_z) - (E_z - G_z \cdot E_y - F_y) \tag{3.13}$$

$$N_y = (E_z - G_z \cdot E_x - F_x) - (E_x - G_x \cdot E_z - F_z) \tag{3.14}$$

$$N_z = (E_x - G_x \cdot E_y - F_y) - (E_y - G_y \cdot E_x - F_x) \tag{3.15}$$

The resulting normals are then normalized such that the length of the edges does not come into account. The normalized vector, $\boldsymbol{N_{norm}}$, is computed as:

$$norm = \sqrt{N_x^2 + N_y^2 + N_z^2} \tag{3.16}$$

$$N_{norm} = [N_{nx} \quad N_{ny} \quad N_{nz}],$$

where $N_{nx} = N_x/\text{norm}$, $N_{ny} = N_y/\text{norm}$, $N_{nz} = N_z/norm$.

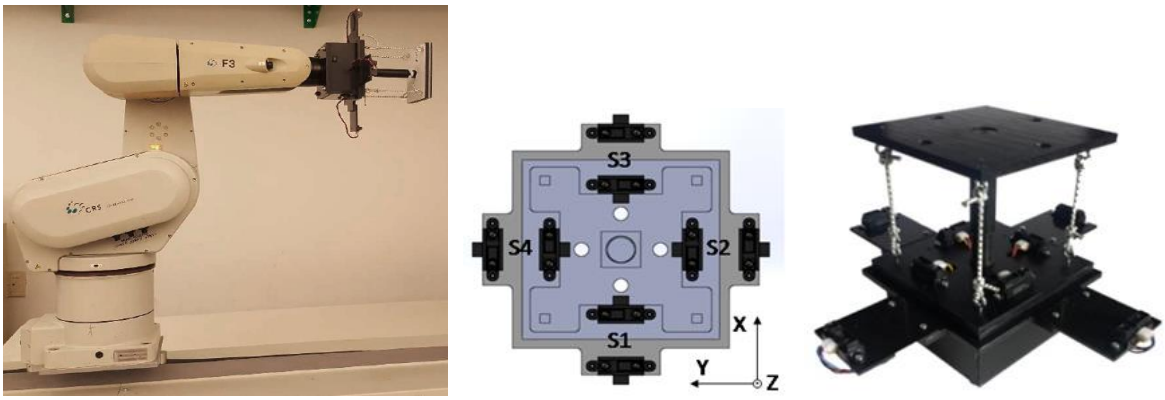
Then, the average of the four triangle normals provides the estimated object's surface normal at the center of the larger occupied cell, and the local object surface orientation is deduced from that normal vector to the surface.

The free-motion mode (f_F) defines the robot position (pos_R), orientation (r_R^{ϕ}) and motion direction (dir_G) at each moving step, k, as follows:

$$f_{Free-motion} = P_G[k] = \begin{cases} pos_R[k] = pos_G[k] = \left[p_k^x, p_k^y, p_k^z \right] \\ r_R^{\phi}[k] = r_G^{\phi}[k] = \left[r_k^{\Theta}, r_k^{\varphi}, r_k^{\psi} \right] \\ dir_G[k] \end{cases} \quad (3.17)$$

3.3 Online (proximity/contact) Path Planning

Embedding human like adaptable compliance is an essential feature for safe interaction in object-robot interaction and surface following applications. Compliance allows for greater margins of uncertainty in sensing technologies meant to monitor the dynamic interaction between the robot and environment which cannot be predicted in advance and to reduce the risk of damages where such measurement errors could be hazardous. The previously active and passive designed compliance devices applied in industrial robots (detailed in section 2.2) cause slower response, reduce position estimation and increase oscillation due to their limitation in providing information only using force/torque sensors during the interaction. Moreover, the precision of the Kinect measurements has been evaluated in [170, 178, 179]. They reported a precision of 1mm-2cm along the X-Y axes and 1mm-4cm along the Z axis (Kinect's axes) which varies based on the object distance from the Kinect. The error may also increase due to inaccuracy in the Kinect/robot calibration process [180]. Furthermore, the object may move or deform under the influence of the physical interaction. Therefore, to address these problems and compensate for errors in the rough 3D profile of the surface provided by the RGB-D sensor, and to further refine the global trajectory, an instrumented compliant robotic wrist device described in Appendix B was designed and built by my colleagues [181] in the SMART research group to support dexterous robotic interaction with live proximity and contact feedback. It is equipped with embedded proximity and contact sensors, and mounted on the robot as its end-effector, as shown in Figure 3.9a.



(a) (b) (c)

Figure 3.9: a) Compliant wrist mounted on the CRS-F3 manipulator, b) internal and external sensors arrangement, c) compliant wrist assembly.

The compliant wrist provides a means of detecting objects both in contact and proximity to the end-effector, as well as adding a degree of compliance to the robot which enables the end-effector to slide on an object without damaging it and adapt to its surface changes when contact happens. As shown in Figure 3.9b, the compliant wrist is equipped with eight infrared (IR) distance measurement sensors. The infrared sensors are arranged in two arrays of four sensors each: an external array and an internal array. The external sensor array allows to measure distances at multiple points between the wrist and the object located in front of the end-effector. The maximum measurement distance of each sensor is 400 mm but since the external sensors are offset from the contacting surface of the compliant wrist, the offset is subtracted from the total distance which results in a 265 mm depth range [181]. The internal sensor array is situated between the base of the compliant wrist structure and a moveable plate, as shown in Figure 3.9c. It allows the device to determine the surface orientation and distance to an object when the robot is in contact with it. The two plates (base and compliant) are attached by a sliding shaft allowing for deflection of the upper compliant plate under externally applied forces which provides a translation of -25 mm when it is completely compressed. The sensing layers estimate an object pose in the form of a 3D homogeneous transformation matrix. The rotation and translation parameters are obtained using the distance measurements from the IR sensors. The wrist was designed as a black box system that can be readily mounted on any type of manipulator, as its embedded electronics is completely independent from the robot controller. The research detailed in this section capitalizes on the live measurements provided by the instrumented wrist to design original control schemes that can also

be integrated as an overlay to any manipulator controller, therefore maximizing the flexibility of the approach.

To interact with the object and closely accommodate its local surface curvature, two self-tuning model-free adaptive position and orientation controllers are developed using online proximity and contact sensory information provided by the compliant wrist (Figure 3.10). These controllers operate from the moment the robot end-effector that is the compliant wrist, reaches in proximity to the considered object surface. The controllers are designed to allow the robot to react to unanticipated events and to quickly correct the path whenever needed and, as such, dynamically refine the offline path with the extra inputs now available. The proposed adaptive on-line trajectory generation is designed to generate continuous command variables and modify the position and orientation of the robot end-effector based on the online sensory information when the robot end-effector is in close proximity of (proximity control mode) or in contact with (contact control mode) an object. As shown in Figure 3.10, distance and orientation deviation values between the robot end-effector and the object's surface are measured simultaneously by the compliant wrist sensors. The values are compared with the desired distance and orientation set by the user as specifications for the robotic interaction (e.g. following a surface at a given distance with normal orientation, or closely following the surface with continuous contact), then the controllers run in parallel to generate position and orientation control signals to compensate the deviations from the specifications. The controllers generate the control signals until there is no change of distance and orientation error between the robot and the object.

The adaptive controllers neither require precise mathematical model of the robot and the environment, nor force/torque calculation or learning procedure to compensate for the unknown system dynamics parameters. As a result, it can be applied to control various robot manipulators of different kinematic and dynamic models.

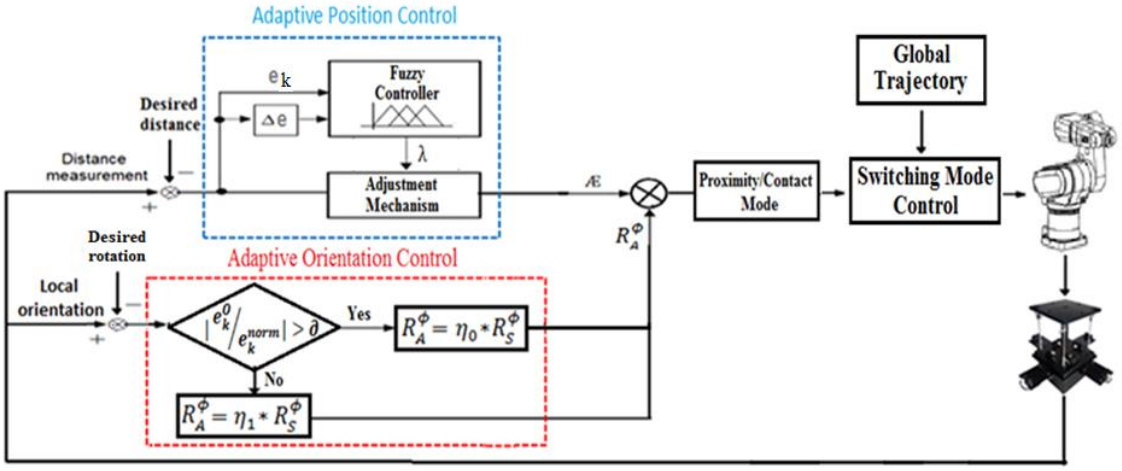


Figure 3.10: Block diagram of the online position and orientation controllers.

3.3.1 Adaptive Position Control

When the robot is in close proximity to or contact with the target points defined by the global trajectory, a position error (distance) between the robot end-effector and the desired set point is calculated based on the currently perceived information by internal (external) sensors. The adaptive position controller generates a position control signal to adjust the robot distance from the object and eliminate the error in proximity and contact phases. The position error, e_k, defined in Eq. 3.18, is determined as a distance between the end-effector and the object. The distance is updated by external sensors or internal sensors when the robot is in close proximity or in contact with the object respectively.

$$e_k = d_k - d_0 \tag{3.18}$$

where d_k is the average of the distance measurements by the four external (or internal) sensors, and d_0 is the desired distance to the object set by the operator. The d_0 value can be positive, negative or zero depending on the application requirement. For example, if it is desired to follow or interact with an object from a specific distance without contact (e.g. spray painting, inspection), then d_0 gets positive values. In some applications such as welding and assembly, it is desired to contact and align the robot end-effector with the object's surface but not to apply much force unto the object, then the value of d_0 is set to zero. On the other hand, for applications such as polishing, particles collection, or live shape (curvature) estimation, a better result will be obtained when the

robot applies force by attempting to move slightly into the object, which causes the compliant plate of the wrist to be deflected, then $d_0 < 0$ is suitable.

3.3.1.1 Design of Proposed Adaptive Position Controller

In order to eliminate the position error, an adaptive position controller is developed to create a closed loop controller with parameters that can modify and correct the end-effector position and react to changes in the environment. From this error, a cost function is formed in which the relationship between the change in error and the cost function is inspired from the MIT rule [182, 183]. The MIT rule is used to develop adaptation mechanism for model reference adaptive control systems. In this rule, a cost function is defined as a function of error between the actual plant and a reference model. The control parameters are adjusted in such a way that the cost function is minimized so that output of the plant becomes the same as the reference model. In contrast to previous works [130-133], where the MIT rule has been applied for model reference adaptive control (MRAC), in this work it is used to design a model-free adaptive controller where the output of the system is compared to a desired distance instead of a reference model. The goal is to minimize the cost related to the error to guarantee that for any given desired distance, the error converges to zero. For this purpose stochastic gradient descent theory [184] is applied. In the typical gradient descent algorithms [185-187], the adaptation gain is either predefined or the gain is derived online by applying learning methods. These methods are computationally expensive and may not be adaptive to unstructured dynamic environments. However, here the adaptation gain is not pre-defined or fixed and no learning procedure is required. Instead, the gains are dynamically updated during the robot-object interaction by gain-adaptation laws developed using fuzzy logic.

From the distance error (Eq. 3.18) a cost function, **J**, can be formed as shown in Eq. 3.19. It should be adjusted in the direction of the negative gradient of **J** to minimize the quadratic cost function and therefore the error.

$$J(Æ_k) = \frac{1}{2}\,(d_k - d_0)^2 = \frac{1}{2}e_k^2 \tag{3.19}$$

$$Æ_{k+1} - Æ_k = -\lambda \frac{\partial J(Æ_{k+1})}{\partial Æ} \tag{3.20}$$

where $Æ$ is the position control signal that defines the amount of movement towards the object, λ is the adaptation gain and $\frac{\partial J}{\partial Æ}$ is the gradient of J.

The control signal $Æ$ is updated and corrected through the gradient descent procedure [184] online to minimize the cost function.

$$Æ_{k+1} = Æ_k - \lambda \frac{\partial \, J(Æ_{k+1})}{\partial Æ_k} \tag{3.21}$$

Using the chain rule [188], the update rule Eq. 3.21 can be rewritten as

$$\frac{\partial \, J(Æ_{k+1})}{\partial Æ_k} = \frac{\partial J(Æ_{k+1})}{\partial e_{k+1}} \frac{\partial e_{k+1}}{\partial Æ_k} = e_{k+1} \frac{\partial e_{k+1}}{\partial Æ_k} \tag{3.22}$$

where $\varphi_k = \frac{\partial e_{k+1}}{\partial Æ_k}$ is called the sensitivity derivative of the system. The control signal $Æ$ is updated and described by

$$Æ_{k+1} = Æ_k - \lambda \, e_{k+1} \varphi_k \tag{3.23}$$

The gradient descent proceeds by finding the optimal solution to minimize the error. However, the control signal, $Æ$, is dictated by the magnitude of the gradient. Therefore, the adaptive controller is very sensitive to the changes in the amplitude of the reference input (e_k), and a high magnitude of the gradient can make the system unstable. In order to reduce the sensitivity of the system and control the moving step's size by the adaptation gain, λ, instead of the gradient magnitude, a normalized update rule is proposed.

$$Æ_{k+1} = Æ_k - \lambda \, e_{k+1} \Phi_k \tag{3.24}$$

where $\Phi_k = \frac{\varphi_k}{\sqrt{\alpha^2 + \varphi_k^2}}$ and α is introduced to remove the difficulty of zero division when φ is small.

The controller designed using this technique is also very sensitive to the amplitudes of the adaptation gain (λ). The adaptation gain (λ) specifies how much the control signal *Æ* can change on each update. Large adaptation gains can make the system unstable. Therefore, λ plays a critical role for the system's stability and convergence. As a general rule, the value of λ is kept small. A small adaptation gain can satisfy the system's stability requirements but it needs many iterations and makes the convergence rate very slow. It is not practical to determine a fixed (optimal) adaptation gain for all moving steps (iterations) due to unpredictable object movement or deformation that can happen during the robot-object interaction. The performance of the system can be improved by adapting the gain, λ, during the interaction. Therefore, it is proposed to determine the adaptation gain, λ, for every moving step, k, based on the error value (Eq. 3.18). The objective is that the farther the robot is from the object surface (or from the desired distance away from the object), the larger should λ be, and therefore results in moving faster toward the object. When the error is reduced (the robot gets closer to the object), then the adaptation gain, λ, gets smaller . Since the convergence rate and stability of the system depend on the adaptation gain, selecting a proper value is crucial.

Fuzzy control is ideal to deal with nonlinear systems where there is a wide range of operating conditions, an inexact model exists and accurate information is not required. In order to obtain the highest adaptation gain that increases the convergence rate and guaranties the stability of the system, it is estimated using a fuzzy controller. The fuzzy controller input variables are the distance error (e_k) and change of error (Δe_k), and the output is the adaptation gain, λ, as shown in Figure 3.11. If the robot is in close proximity of the object surface, the proximity motion mode is activated and the external sensors from the compliant wrist provide the input for the fuzzy controller. Alternatively, when the robot end-effector is in contact with the object, the contact motion mode kicks in and the internal sensors from the compliant wrist provide the distance measurements for the controller. The standard triangular membership functions and Mamdani type inference are adopted for the fuzzification due to their small computation cost which enhances the reaction speed of the system. The error (e_k) in distance is represented by five linguistic terms, as shown in Figure 3.11a: large negative (*LN*), small negative (*SN*), zero (*Z*), small positive (*SP*), and large positive (*LP*). Similarly in Figure 3.11b, the fuzzy set of change of error (Δe_k) is expressed by negative big

(*NB*), negative (*N*), zero (*Z*), positive (*P*), and positive big (*PB*). The output linguistic terms for adaptation gain are (Figure 3.11c): very slow (*VS*), slow (*S*), medium (*M*), Fast (*F*), and very fast (*VF*) over the interval [0, 1].

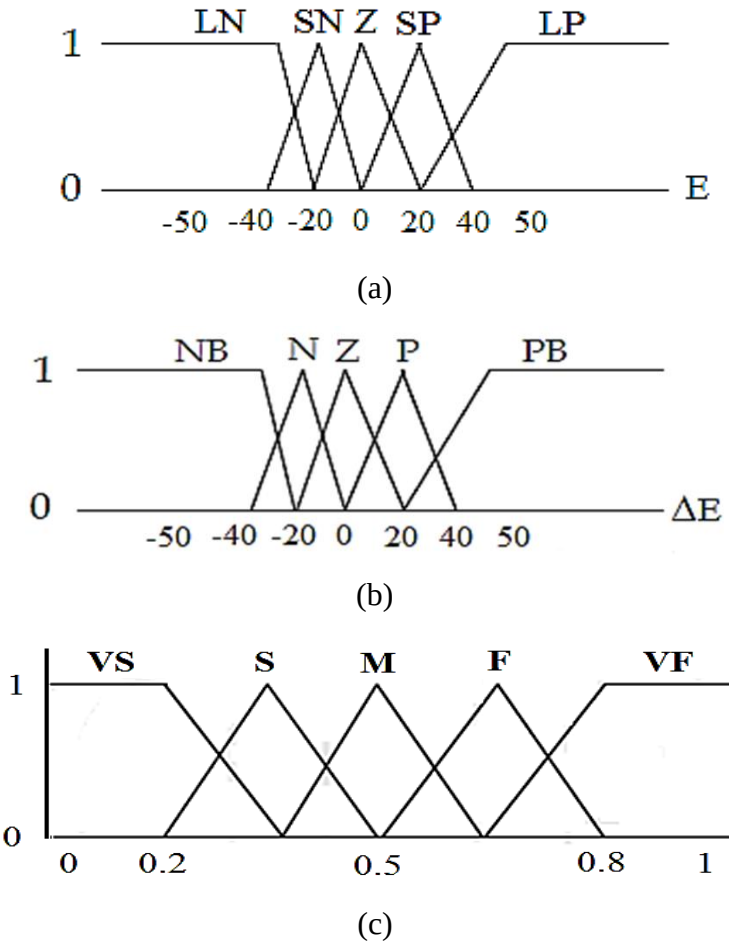


Figure 3.11: Fuzzy membership functions for: a) error, b) change of error, and c) adaptation gain.

Table 3.1 shows the fuzzy rules where the adaptation gain is determined depending on the robot distance to the surface. The rules are developed using the method proposed in [189-190] for fast convergence and minimum oscillations. The center of gravity (COG) method is used to diffuzify the output variable, λ, as defined in Eq. 3.25.

$$\lambda = \frac{\sum_{i=1}^{m} \mu(f_i).f_i}{\sum_{i=1}^{m} \mu(f_i)} \tag{3.25}$$

where μ is membership function, f_i is the numerical values of output set and m is number of the output fuzzy sets.

Table 3.1: Fuzzy rule base.

ΔE \ E	LN	SN	Z	SP	LP
NB	VF	F	M	F	VF
N	F	M	S	M	F
Z	M	S	VS	S	M
P	F	M	S	M	F
PB	VF	F	M	F	VF

3.3.1.2 Stability of Proposed Adaptive Position Controller

To investigate the stability of the adaptive controller, the discrete Lyapunov function candidate is defined as:

$$V_k = \frac{1}{2} {e_k}^2 \tag{3.26}$$

where e_k is the distance error defined in Eq. 3.18.

The change in Lyapunov function is obtained by:

$$\Delta V_k = V_{k+1} - V_k = \frac{1}{2} e_{k+1}^2 - \frac{1}{2} e_k^2 \tag{3.27}$$

where the error at index time k+1 can be written as:

$$e_{k+1} = e_k + \Delta e_k \tag{3.28}$$

Thus Eq. 3.28 can be rewritten as:

$$\Delta V_k = \frac{1}{2} [e_k + \Delta e_k]^2 - \frac{1}{2} e_k^2 = \Delta e_k [e_k + \frac{\Delta e_k}{2}] \tag{3.29}$$

The error difference can be estimated by [191]:

$$\Delta e_k \approx \frac{\partial e_{k+1}}{\partial Æ_k} \Delta Æ_k = \\ -\lambda\, e_{k+1}\varphi_k \Phi_k = -\lambda\, e_{k+1}\frac{\varphi_k^2}{\sqrt{\alpha^2+\varphi_k^2}} \tag{3.30}$$

By substituting Eq. 3.28 in Eq. 3.30,

$$\Delta e_k = -\lambda\,(e_k + \Delta e_k)\frac{\varphi_k^2}{\sqrt{\alpha^2+\varphi_k^2}} \tag{3.31}$$

$$\Delta e_k = -\lambda \frac{e_k \varphi_k^2}{(\sqrt{\alpha^2+\varphi_k^2}+\lambda\varphi_k^2)} \tag{3.32}$$

Finally, the change of the Lyapunov can be defined as:

$$\Delta V_k = \frac{-\lambda\, e_k \varphi_k^2}{\left(\sqrt{\alpha^2+\varphi_k^2}+\lambda\varphi_k^2\right)}\left[e_k + \frac{-\lambda\, e_k \varphi_k^2}{2\left(\sqrt{\alpha^2+\varphi_k^2}+\lambda\varphi_k^2\right)}\right] \\ = \frac{-\lambda\,\varphi_k^2}{(\sqrt{\alpha^2+\varphi_k^2}+\lambda\varphi_k^2)}\left[1 - \frac{\lambda\,\varphi_k^2}{2(\sqrt{\alpha^2+\varphi_k^2}+\lambda\varphi_k^2)}\right] e_k^2 \le 0 \tag{3.33}$$

The candidate Lyapunov function for the position controller is positive definite and the change of the function ΔV_k is negative definite where the adaptation gain, λ, is positive. Therefore the system is (locally) asymptotically stable in the sense of Lyapunov.

3.3.2 Adaptive Orientation Control

The orientation error is determined as the angular deviation between the robot's end-effector and the object surface normal direction (Eq. 3.34). When the robot is in close proximity of (or in contact

with) the object, the orientation error of the object surface is estimated by the proximity/contact sensors embedded in the compliant wrist, in terms of the roll (r_O^{θ}) and pitch (r_O^{φ}) angles (Euler angles) of the object surface with respect to the wrist reference plane. The yaw angle (r_O^{ψ}), corresponding to the rotation around the compliant wrist pointing direction, is not measurable by the compliant wrist sensory stage. But as it does not affect the end-effector alignment with respect to the surface, it is set to zero by default.

$$e^O = [r_S^{\phi}]^T = \begin{bmatrix} r_O^{\theta} \\ r_O^{\varphi} \end{bmatrix} - \begin{bmatrix} r_E^{\theta} \\ r_E^{\varphi} \end{bmatrix} = \begin{bmatrix} r_S^{\theta} \\ r_S^{\varphi} \end{bmatrix} \tag{3.34}$$

where the r_E^{θ} and r_E^{φ} correspond to the roll and pitch angles, respectively, of the end-effector.

Let the rotation matrix of the object surface generic orientation, r_O^{ϕ}, be denoted as R_O^{ϕ} with respect to the wrist (end-effector) frame, and the rotation matrix of the end-effector with respect to the robot base frame be described by R_E^{ϕ}, then the desired rotation of the end-effector with respect to the robot base frame, R_R^{ϕ}, to match the compliant plate with the object surface can be determined by (Eq. 3.35).

$$R_R^{\phi} = R_E^{\phi} * R_O^{\phi} \tag{3.35}$$

In a robot-object interaction task, R_R^{ϕ} is set as the orientation to be tracked by the robot end-effector. However, in order to obtain smooth orientation adjustments, it is not directly sent to the controller, and instead the orientation of the end-effector is updated using an adaptive orientation signal, at each moving step to correct the orientation error between the end-effector and the object's surface.

3.3.2.1 Design of Proposed Adaptive Orientation Controller

The use of classical exponential error decrease is common for error regulation where the error decreases exponentially to minimize the error.

$$\Delta e^{O} = -\eta * e^{O} = r_A^{\phi} \tag{3.36}$$

$$r_A^{\phi} = -\eta * r_S^{\phi} \tag{3.37}$$

where $r_S^{\phi} = \left[r_S^{\Theta}, r_S^{\varphi}, 0\right]$ and η is a proportional gain.

To avoid instability, the proportional gain is generally a small value and it remains constant during the interaction, which increases the convergence time. On the other hand, a large gain value reduces the convergence time but may lead to tracking loss or task failure due to fast motion. Therefore, in order to reduce the convergence time and oscillation near the equilibrium point, while preserving the system stability, the gain, η, should also be tuned according to the error value at each moving step. When the orientation error is very small, the convergence rate reveals to be fairly poor and therefore the final orientation correction becomes very slow and oscillatory. To avoid the problem, the gain must rather switch back to a larger gain near the equilibrium point ($e^{O} = 0$) to ameliorate the performance. A similar adaptive controller as the proposed position controller (Section 3.3.1) can be used for orientation control. But setting up the fuzzy rule base for tuning the orientation adaptation gain that satisfies the above switching strategy is not very efficient because it requires a larger number of membership functions, rules and variables. As the rule base of the controller increases, computation and therefore convergence takes longer for each control cycle, so that it is nearly impossible to set up fuzzy rules for more than three inputs [192]. In this case, exponential error regulation methods [193-194] offer a better performance with a much more reduced set of parameters. Due to this choice, a new exponential error regulation strategy is proposed to compute a proper adaptation gain value for η. In contrary of classic exponential error regulation methods, the adaptation gain is not constant and switches from large to small when the error decreases, and switches back to larger gain when error arrives to a specified threshold. In addition, another formulation of the regulation is used where the error norm is applied in computation of the adaptation gain to accelerate the convergence and avoid instability. The following equations are developed based on the orientation error that is updated online using the information provided by the compliant wrist sensors.

$$\eta_k = \begin{cases} 1 - \mu \exp\left(-\xi \left| e_k^O / e_k^{norm} \right|\right), & \left| e_k^O / e_k^{norm} \right| > \partial \\ \mu \exp(-\xi \left| e_k^O / e_k^{norm} \right|), & \left| e_k^O / e_k^{norm} \right| < \partial \end{cases} \tag{3.38}$$

where $[e_k^O]^T = [r_S^{\Theta}[k], r_S^{\varphi}[k]]$, $e_k^{norm} = \sqrt{r_S^{\Theta}[k]^2 + r_S^{\varphi}[k]^2}$, $\eta_k = [\eta_{\Theta}[k],\ \eta_{\varphi}[k]]$ and where μ and ξ are positive constant scalar values and ∂ is the switching threshold.

As shown in Eq. 3.38, in the first proposition, the adaptation gain is large when the orientation error is large and it gets smaller when the error reduces until the error arrives to a small value. According to the second proposition, the adaptation gain switches back to a larger gain near the convergence. The gain tuning scheme leads to increase the convergence rate and prevent oscillation and instability.

By using the adaptive orientation signal, the end-effector orientation angles necessary to match with the object's surface orientation are calculated by:

$$r_A^{\phi}[k] = \begin{cases} r_A^{\Theta}[k] = \eta_{\Theta}[k] * r_S^{\Theta}[k] \\ r_A^{\varphi}[k] = \eta_{\varphi}[k] * r_S^{\varphi}[k] \\ r_A^{\psi}[k] = 0 \end{cases} \tag{3.39}$$

Let the rotation matrix of the adaptive orientation signal, r_A^{ϕ}, be denoted with R_A^{ϕ}, with respect to the wrist (end-effector) frame. The desired robot end-effector orientation with respect to the robot base frame at each moving step, k, is determined using the orientation matrix, R_R^{ϕ}.

$$R_R^{\phi}[k] = R_E^{\phi}[k] * R_A^{\phi}[k] \tag{3.40}$$

The resulting change in the orientation error for the above mentioned orientation correction is given by:

$$\Delta e_k^o = e_{k+1}^o - e_k^o = -\eta_k e_k^o = -\eta_k \begin{bmatrix} r_S^{\Theta}[k] \\ r_S^{\varphi}[k] \end{bmatrix} \tag{3.41}$$

3.3.2.2 Stability of Proposed Adaptive Orientation Controller

The stability of the orientation controller can be verified using the Lyapunov candidate function defined as:

$$V_k = \frac{1}{2}\, e_k^{o^T} e_k^o \tag{3.42}$$

where e_k^o is the orientation error defined in Eq. 3.34.

The change of the Lyapunov function using Eq. 3.41 is obtained as:

$$\begin{gathered} \Delta V_k = V_{k+1} - V_k = \frac{1}{2}\, {e_{k+1}^o}^T e_{k+1}^o - \frac{1}{2}\, e_k^{o^T} e_k^o = \\ \frac{1}{2}\,[(\Delta e_k^o + e_k^o)^T(\Delta e_k^o + e_k^o)] - \frac{1}{2}\, e_k^{o^T} e_k^o = \; e_k^{o^T}\Delta e_k^o + \frac{1}{2}\,\Delta e_k^{o^T}\Delta e_k^o \end{gathered} \tag{3.43}$$

$$\Delta V_k = -\eta_k e_k^{o^T} e_k^o - \frac{1}{2}{\eta_k}^2 e_k^{o^T} e_k^o = -\eta_k \left(e_k^{o^T} e_k^o - \frac{1}{2}\eta_k e_k^{o^T} e_k^o \right) \tag{3.44}$$

where $\Delta V_k < 0$ if $0 < \eta_k < 1$. Therefore, the orientation control system is stable in the sense of Lyapunov.

3.3.3 Proximity and Contact Motion Mode

As mentioned before, when the robot is in close proximity of the object surface, the *proximity mode* is activated and external sensors provide the distance and orientation of the object surface relative to the compliant wrist mounted on the robot as the end-effector. While the robot follows the global trajectory defined in section 3.2, the adaptive controllers generate proximity position control signal, $Æ_P$ (Eq. 3.24), and orientation control signals, r_{AP}^{φ}, (Eq. 3.39), where subscript *P* holds for *proximity*, to dynamically update and modify the end-effector pose over the object using the new sensory information as follows:

$$f_{Proximity} = \begin{cases} pos_R[k] = pos_E\,[k] + \text{Æ}_P[k] \\ r_R^{\phi}[k] = r_E^{\varphi}[k] + r_{AP}^{\varphi}[k] \end{cases} \tag{3.45}$$

where pos_E, r_E^{φ} are the current robot position and orientation.

When the robot touches the objects surface, the compliant plate is deflected and the *contact mode* is activated. The internal sensors provide contact information for the adaptive controllers to generate contact position (Eq. 3.24) and orientation (Eq. 3.39) using Eq. 3.46 to refine the end-effector pose around the global trajectory defined in section 3.2.

$$f_{Contact} = \begin{cases} pos_R[k] = pos_E\,[k] + \text{Æ}_C[k] \\ r_R^{\phi}[k] = r_E^{\varphi}[k] + r_{AC}^{\varphi}[k] \end{cases} \tag{3.46}$$

where Æ_C, r_{AC}^{φ} are adaptive position and orientation signals in the contact motion mode.

3.4 Hybrid Switching Control Scheme

Hybrid systems in general are composed of two distinct components in which continuous subsystems and discrete subsystems interact with each other. As discussed in section 2.4, hybrid systems have been used frequently in sensor-based robotic control systems where the robot interacts with the environment. A class of hybrid systems that has received significant attention recently in multimodal control consists of switched systems that provide a framework to model and integrate several control modes and feedback control algorithms while ignoring irrelevant details using supervisory logic-based control laws [194]. A hybrid switched system consists of a set of linear or nonlinear functions and a supervisor that selects a particular function to control the system in response to dynamics of the system.

3.4.1 Design of the Proposed Hybrid Switching Control Scheme

This work proposes an original hybrid switched system which consists of four subsystems (interaction modes) and a unique switching control scheme to combine and smoothly switch between these interaction modes. Unlike previously proposed hybrid systems (discussed in section

2.4) that use force/torque sensors and vision sensors to control a robot motion (hybrid vision/force control) where local information from the environment is acquired from a force/torque sensor only when contact occurs between the robot and the environment, here the hybrid system utilizes and combines the RGB-D data provided by a Kinect sensor, extra data from proximity sensors during close approach, and finally data from contact sensors provided by the compliant wrist. This enables the robot to adapt reactively and interact with the environment with or without contact. In addition to the sensory improvement, the contribution of the proposed hybrid switched system is the weighted combinations of hard switch and blend switch between the different motion modes through using logistic functions [195] that orchestrate the transition between the interaction modes and their respective controllers.

The supervisory control scheme is designed to implement a switching law that creates a mixture of hard and blend switches (Figure 3.12) to overcome chattering, oscillation and instability problems when the system switches from one mode to another, when a subsystem fails due to noise, or other possible sensor failures. The three oval containers of the graph in Figure 3.12 represent the three main motion modes of the system and the transitions are represented by the directed arrows. The unidirectional arrows and the bidirectional arrows show the hard switches and blend switches between the motion modes respectively. Blend switching leads to smooth transition between two interaction modes and allows complementary use and data fusion from multiple sensing sources by combining pairs of the main motion modes, and related sensing stages, in what is called here a *blend* motion mode.

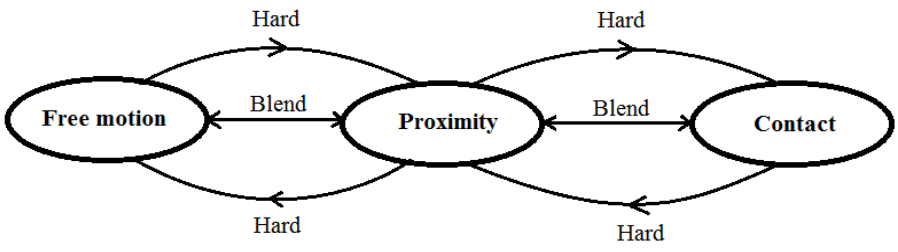


Figure 3.12: Proposed switching control system model.

As shown in Figure 3.13, the switching signal varies when the robot finds itself within certain regions of the workspace based on the end-effector distance to the object surface (the operational

distance of the compliant wrist varies from -25mm to 265mm [181]). The *free* motion mode (f_F) is activated when the robot is far from the object or all the sensors embedded on the complaint wrist cannot detect the object to guide the robot toward it and adjust the end-effector to fit over the object surface. Navigation is then guided from data acquired with the Kinect sensor, which leads to global path planning as detailed in section 3.2. Once the object is successfully detected and the robot reaches within a certain distance from the object, the proximity motion mode (f_P) is activated where the sensory information provided by external sensors of the compliant wrist is used to locally adjust the robot pose over the object surface without any contact. The contact motion mode (f_C) kicks in when the robot first comes in contact with the object. The true contact is detected by the internal sensors when the compliant contact plate is deflected from its rest state and the translational shaft is compressed by a certain amount. Reversely, once the task is completed, the robot moves away from the surface, switches back to *proximity* mode, and eventually reaches far away from the object and then switches back in *free* motion mode. The blend motion modes are respectively a mixture of *free* motion and *proximity* (f_{PF}), or *contact* and *proximity* (f_{CP}), which enable the robot to use two different motion modes simultaneously and switch smoothly between them where a transition from a motion mode to another is required and where using only one sensing modality may not provide accurate enough or adequate information. The *blend* switching also plays an important role under special circumstances that may appear during operation, like loss of contact with surface due to the object movement or deformation, end of object, high curvature, etc. As such the multiple connections and combination of hard and blend switches provide a robust framework that is able to recover from more complex situations than classical force/torque controllers.

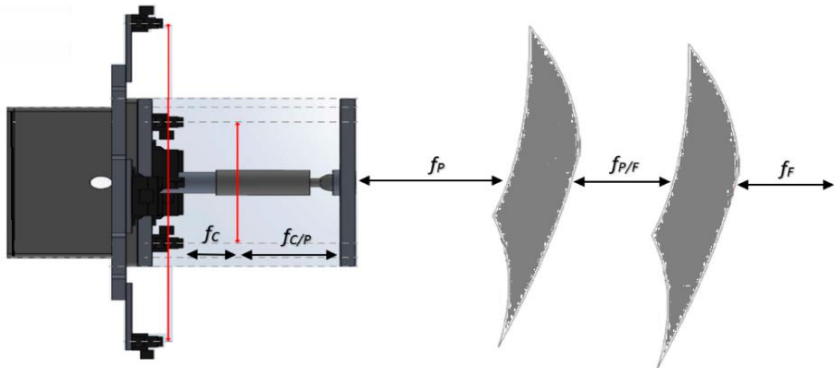


Figure 3.13: Workspace decomposition.

In this work, the switching control is developed based on the position error (Eq. 3.18) between the robot end-effector and the desired set point, here set at a desired distance, d_0, to the object surface at each moving step, k. It is modeled as follows:

$$e_{k+1} = f_{\delta(e)}(e_k, d_k), \qquad \delta(e) \in \{F, P, C, PF, CP\} \tag{3.47}$$

where d_k is the current distance from the object obtained by either the internal sensors (d_k^{int}) in the *proximity* mode, or from the external sensors (d_k^{ext}) of the compliant wrist in the, *contact* mode; $\delta(e)$ determines which function $f_{\delta(e)}(e_k, d_k)$ controls the system behavior at each moving step (*k*), where $f_F, f_P, f_C,$ correspond to *free* (F) motion, *proximity* (P), and *contact* (C) control modes respectively. f_{PF} refers to the proximity and free motion modes combination (called *blended proximity-free,* PF), and f_{CP} is the contact and proximity motion modes combination (called *blended contact-proximity,* CP). The switching signal (δ) is determined using Eq. 3.48, according to the internal error (e^{int}) and external error (e^{ext}). The error values shown in Eq. 3.48 are in mm and are defined experimentally based on the compliant wrist sensor's performance analysis detailed in [181]. For a compliant wrist equipped with different IR sensors, or other robot-mounted distance measurement devices, re-calibration of Eq. 3.48 would be necessary, but the general framework would remain valid as such.

$$\delta(e) = \begin{cases} C, & e_k^{int} \le -15 \\ CP, & -15 < e_k^{int} < -2 \\ P, & -2 \ge e_k^{int}, e_k^{ext} \le 45 \\ PF, & 45 < e_k^{ext} < 80 \\ F, & e_k^{ext} \ge 80 \end{cases} \tag{3.48}$$

$$\text{where} \quad e_k = \begin{cases} e_k^{int} = d_k^{int} - d_0 \\ e_k^{ext} = d_k^{ext} - d_0 \end{cases}$$

The interaction mode selection and combination of the modes is supervised by the supervisor, *S* to smoothly switch among the motion modes (Eq. 3.49).

$$S = \begin{cases} f_{\delta(e)} = (1-\beta)f_C, & \delta(e) = C \\ f_{\delta(e)} = \beta f_P + (1-\beta)f_C, & \delta(e) = CP \\ f_{\delta(e)} = \beta f_P, & \delta(e) = P \\ f_{\delta(e)} = \beta f_P + (1-\beta)f_F, & \delta(e) = PF \\ f_{\delta(e)} = (1-\beta)f_F, & \delta(e) = F \end{cases} \tag{3.49}$$

The blending gain (β) is determined for gain scheduling based on the error, e_k, (Eq. 3.49) to apply one or blend two motion control modes in different operational conditions. The supervisory control framework and the blending gain curve is shown in Figure 3.14.

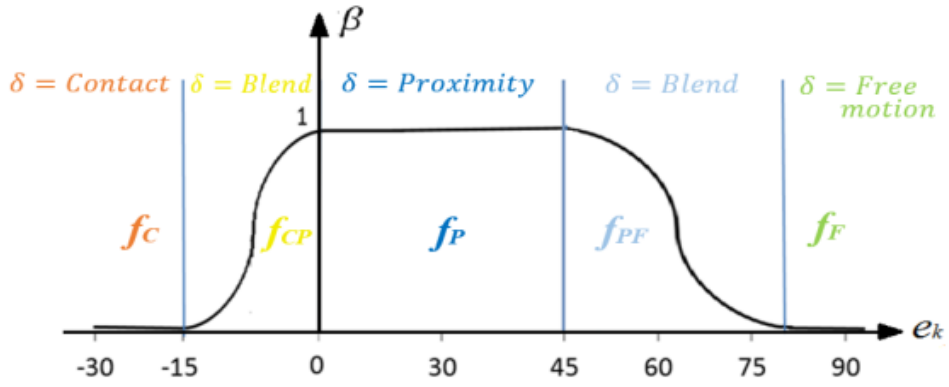


Figure 3.14: Switching signal and supervisory control framework.

The β value is bounded to [0, 1], with β=0 corresponding to the *free* motion and *contact* modes, and β=1 to the *proximity* mode. In order to blend the interaction modes (blend switching), β is computed using the logistic function (Eq. 3.50). The logistic function combines two characteristics of the exponential function in one, that is the exponential (fast) growth first, and the bounded (slow) exponential growth when values are near the limits [195].

$$\beta = A + \frac{C - A}{1 + e^{-\mathrm{k}(e_k - \ell)}} \tag{3.50}$$

where A is the lower asymptote, C is the upper asymptote, k is the growth rate, and ℓ is the value of the curve midpoint. Therefore, to obtain the profile of Figure 3.14, in the two blend zones the blending gain is computed as follows based on the operational characteristics of the compliant wrist considered in this research:

$$\beta = \begin{cases} \dfrac{1}{1 + e^{-0.5\left(e_k^{int}+7\right)}} & -15 < e_k^{int} < -2 \\ \dfrac{1}{1 + e^{0.2\left(e_k^{ext}-62.5\right)}} & 45 < e_k^{ext} < 80 \end{cases} \tag{3.51}$$

The proposed hybrid switched system is designed based on the distance error between the end-effector and the object surface. Therefore, the compliant wrist can be replaced by any other sensors that can provide similar distance measurements. Even force/torque sensors could be applied to control contact forces instead of the internal sensors where the error would be defined as the difference between the desired force and the measured force applied on the object.

3.4.2 Stability of Proposed Hybrid Switching Control Scheme

A number of different theories have been proposed to ascertain stability of hybrid switched systems. One of the frequently used approaches to demonstrate stability of hybrid switched systems is the multiple Lyapunov functions (MLFs) introduced by Branicky [196], where a Lyapunov function, $V_i(x)$, is defined for each control mode, $f_i(x)$, and the stability of the system is guaranteed if: 1) all the subsystems are stable, and 2) their corresponding Lyapunov function is decreasing from the value it had the last time the subsystem left the control mode, i, which can be achieved by exerting some conditions and restrictions on the switching.

In section 3.3.1.2 and 3.3.2.2, it was shown that the Lyapunov functions associated with the position and orientation adaptive controllers are positive definite and have a negative definite derivative that ensure the stability of the subsystems. The switching rules of Eq. 3.48 ensure that a subsystem is activated when its constraints are satisfied and it remains active as long as its Lyapunov function continues to decrease. In order to guaranty the stability of the overall system and satisfy the second stability condition, the supervisory control scheme enforces another rule to restrict the switching conditions. The switching rule (Eq. 3.48) applies only when the interaction mode, $f_{\delta(e)}$, has been previously activated. Suppose that $V_{\delta(e)}$ is a Lyapunov function associated with the subsystem $f_\delta(e)$, our proposed hybrid switched system orchestrates the transition between interaction modes and picks a subsystem $(f_F, f_P, f_C, f_{PF}, f_{CP})$ in a way that at each instant, k, the

corresponding Lyapunov function of the subsystem is less than it was last time the controller operated in that mode.

$$V_{\delta(e)}(k) < V_{\delta(e)}(j) \tag{3.52}$$

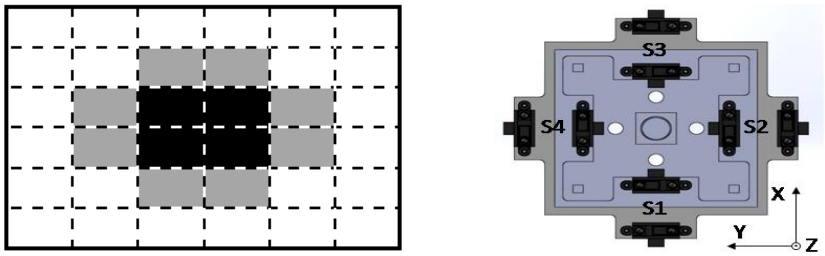
where $k > j$, and j is the last time the subsystem, $f_{\delta(e)}$, left the control mode $\delta(e)$. Therefore, the proposed hybrid switched system controls the transition between the interaction modes and their respective controllers in a way that guarantees the stability of the system in the sense of multiple Lyapunov functions (assuming that the robot manipulator is stable).

3.5 Updating the Occupancy Grid and Object Retrieval

The offline global trajectory initially guides the robot towards the object to interact with and to explore its surface using the modified 3D occupancy grid. As discussed in section 3.2.2.2, the state variables associated with the larger cells of the grid is unknown (0.5) when the cell is not completely occupied. This normally happens when the object has acute edges or has not been acquired well by the Kinect sensor due to its limited range of operation, its accuracy, or lighting problems. Furthermore, the object may move or deform during the interaction. Therefore, to achieve complete exploration of the object and complement the coverage path planning, it is required to update the occupancy grid and react to movements of the object surface.

When the robot is in close proximity to the object, the compliant wrist embedded sensors provide closer and higher accuracy feedback about it. In order to update the occupancy grid, the local information provided by the external sensors of the compliant wrist are used to update the global trajectory. As mentioned in section 3.2.2.1, the larger cells size was selected the same size as the robot's tool plate (compliant plate) size. As shown in Figure 3.15a, if we consider the white region as a part of the object of interest surface decomposed into the smaller cells, given the selected resolution of the occupancy grid, the compliant plate is able to cover four smaller cells (one larger cell), and the external sensors which are protruding over the four sides of the compliant plate can provide online look-ahead information while the robot is moving along the global trajectory.

As shown in Table 3.2, a set of If-Then rules are considered to decide in which direction to steer the robot when operating under guidance of the compliant wrist external sensors (S1, S2, S3 and S4). The directions are defined with respect to the end-effector's reference frame and can be either -X, -Y, +X, +Y or certain combinations of these (Figure 3.15 b). For example when the robot is moving to the right and the next cell value is 0.5 (unknown cell), the sensor situated on the right side (S2) will allow an update of the state of the two cells located ahead on the right. If the sensor detects the object in that position, the cell value changes to 1 and the robot moves to the right, else it will change to zero and the robot moves towards the next occupied cell.



(a) (b)

Figure 3.15: a) Cells coverage by the wrist, b) compliant wrist sensors arrangement.

Table 3.2: Object retrieval rules under guidance from compliant wrist embedded sensors.

If	*Then*	**Direction**	**If**	*Then*	**Direction**
None		N/A	S3 , S4		+X & +Y
S1		-X	S4 , S1		+Y & -X
S2		-Y	S1, S2, S3		-Y
S3		+X	S2, S3, S4		+X
S4		+Y	S3, S4, S1		+Y
S1, S2		-X & -Y	S4, S1, S2		-X
S2 , S3		-Y & +X	S1, S2, S3, S4		N/A

3.6 Chapter Summary

This chapter presented the design and implementation of a hybrid switched control system operating from multi-sensory inputs for automated and rapid object segmentation, proximity/contact interaction and surface following. The hybrid control architecture enables the robot to detect and follow the surface or interact with an object of interest with or without contact using the information provided by a Kinect sensor and augmented locally using a custom designed instrumented compliant wrist with embedded proximity and contact distance sensors. A unique supervisory control scheme is proposed to supervise the robot motion continuously and to smoothly switch between five motion modes ($f_F, f_P, f_C, f_{FP}, f_{PC}$), based on the robot end-effector's distance from the object surface. Each of the 3 main motion control modes (f_F, f_P, f_C) generates a position control signal and an orientation control signal using a specific sensing modality.

In the *free* motion mode, the 2.5D model of the object acquired via a peripheral RGB-D vision stage is used to plan a global trajectory (detailed in section 3.2) and guide the robot from an initial position towards the object. The *proximity* mode then enables the robot to refine the reference (global) trajectory using the proximity sensing information to track the desired trajectory within a specific distance without contact, or to finely adjust the robot pose before contact happens (transition) to damp the system and match its pose with the surface. The contact with surface is triggered by the internal sensors where the contact plate is deflected under externally applied forces. The *contact* motion mode enables the robot to more precisely adapt to the changes and forces generated by the surfaces with which it comes into contact. When the contact is detected, the internal sensors provide the fuzzy inputs (error and change of error) instead of the external sensors to make sure that the robot remains in contact with the object and prevent the robot from pushing too much into the object, even though forces are not formally measured nor estimated.

Although using the compliant wrist increases the flexibility and reduces the risk of damages, the compliance reduces the stiffness required for position control. As a consequence, the position measurement accuracy is decreased. To achieve a better accuracy and smoothly switch from one motion mode to another and prevent chattering, oscillation and instability, the sensory information from different sources, and therefore the motion modes, needed be combined under specific

conditions, which led to the development of *blend* motion modes. Therefore, a hybrid switched control system including a unique supervisory control scheme is designed for dictating an original switching law that is a mixture of hard and blend switches to determine which motion mode(s) should be active at each moving step.

Chapter 4. EXPERIMENTAL RESULTS

In order to validate and demonstrate the performance and efficiency of the proposed methods, experimental validation of robot-object interaction under the free, proximity, contact and hybrid (proximity and contact) control modes are carried out using a 7-DOF CRS-F3 robot (Appendix C) on which the instrumented compliant wrist is mounted as the end-effector. The purpose of these experiments is to explicitly evaluate the performance of the proposed hybrid switched system and of the adaptive controllers for object detection, surface following and reactive interaction with static and moving objects. Five experimental protocols are examined. In the first three experiments, a planar object has been used rather than a curved surface to evaluate the performance and accuracy of the adaptive controllers and of the switching scheme in proximity and contact interactions with a static object. The fourth experiment is dynamic object-robot interaction where a flat panel moves frequently to random positions and orientations in front of the end-effector and it should detect and follow the object pose continuously. Furthermore, the system's performance when achieving reactive interaction and surface following over a curved surface is demonstrated in the fifth experiment presented in this chapter. The various test cases are closely examined to confirm the validity of the proposed approaches and the results are discussed and compared with other methods from the literature.

4.1 Object Segmentation

The first step to guide the robot towards the object of interest in the robot workspace, and efficiently come to interact with it, is object localization and segmentation from the scene (section 3.2). A Kinect sensor is used to collect color and depth information within only a few seconds. As shown in Figure 3.3, the Kinect sensor is located behind the robot. Therefore, the manipulator was initially located at the extremity of its workspace to avoid creating any occlusion during the RGB-D data acquisition phase. The RGB image (Figure 4.1) is presented to the operator who can then click on any points on the desired object in the image from which the color and depth information is then processed and filtered using Algorithm 1 in order to extract and segment the 3D information corresponding to the object of interest.

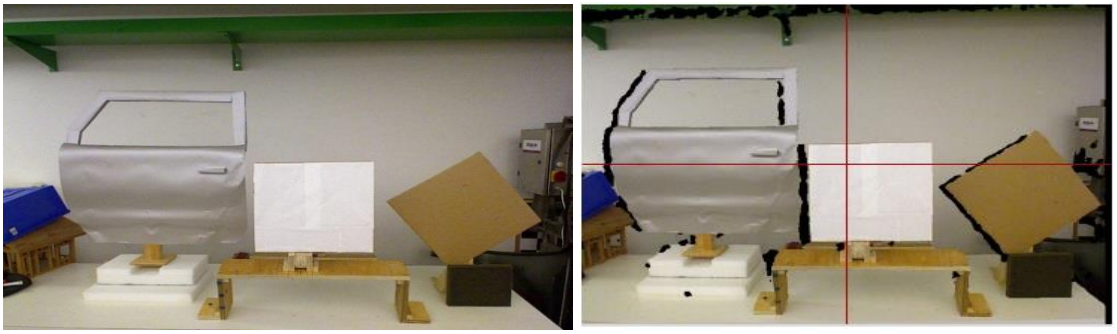


Figure 4.1: RGB-D data captured by the Kinect sensor: a) RGB image, b) depth.

Table 4.1 shows the efficiency of the proposed segmentation algorithm at extracting each object of interest from the scene shown in Figure 4.1, where there are four objects of different shapes and colors. The original point cloud provided by the Kinect sensor after applying the depth filter includes 41204 points from which over 99% of the object of interest have been successfully recovered from the point cloud after the operator selected a point on each of them. The considered exact number of points in each object is extracted using MeshLab software and compared to the number of points extracted with the proposed approach that combines color and depth filters.

Table 4.1: Object segmentation results.

Number of vertices / Objects	Original object	Extracted from the original	Segmentation rate
White	7464	7434	99.59
Brown	8797	8774	99.73
Door	17071	17019	99.69
Black	3085	3068	99.44

4.2 Static Object-Robot Interaction

While interacting with static objects, two scenarios are examined. In both cases, the object is initially placed in a random pose within the robot's workspace (more precisely within the compliant wrist's embedded sensors operational field of view) and remains in that position and orientation until the robot end-effector comes to match its position and surface orientation (here called pose adjustment). The moving steps that the robot performs to reach the desire final pose (here called

iterations) are also closely monitored. The process is repeated 10 times in each test run with the object adopting a new random static pose that the robot end-effector attempts to match. In these scenarios the objective is to evaluate the performance of both controllers in the online trajectory planning phase only, that is using the embedded sensors from the compliant wrist. Therefore, the offline path planning phase that relies on Kinect sensor's data is skipped in the first two experiments, and no surface following and coverage is considered. A single object-robot matching configuration is pursued at a time, before the test resumes with a new random pose to reach to.

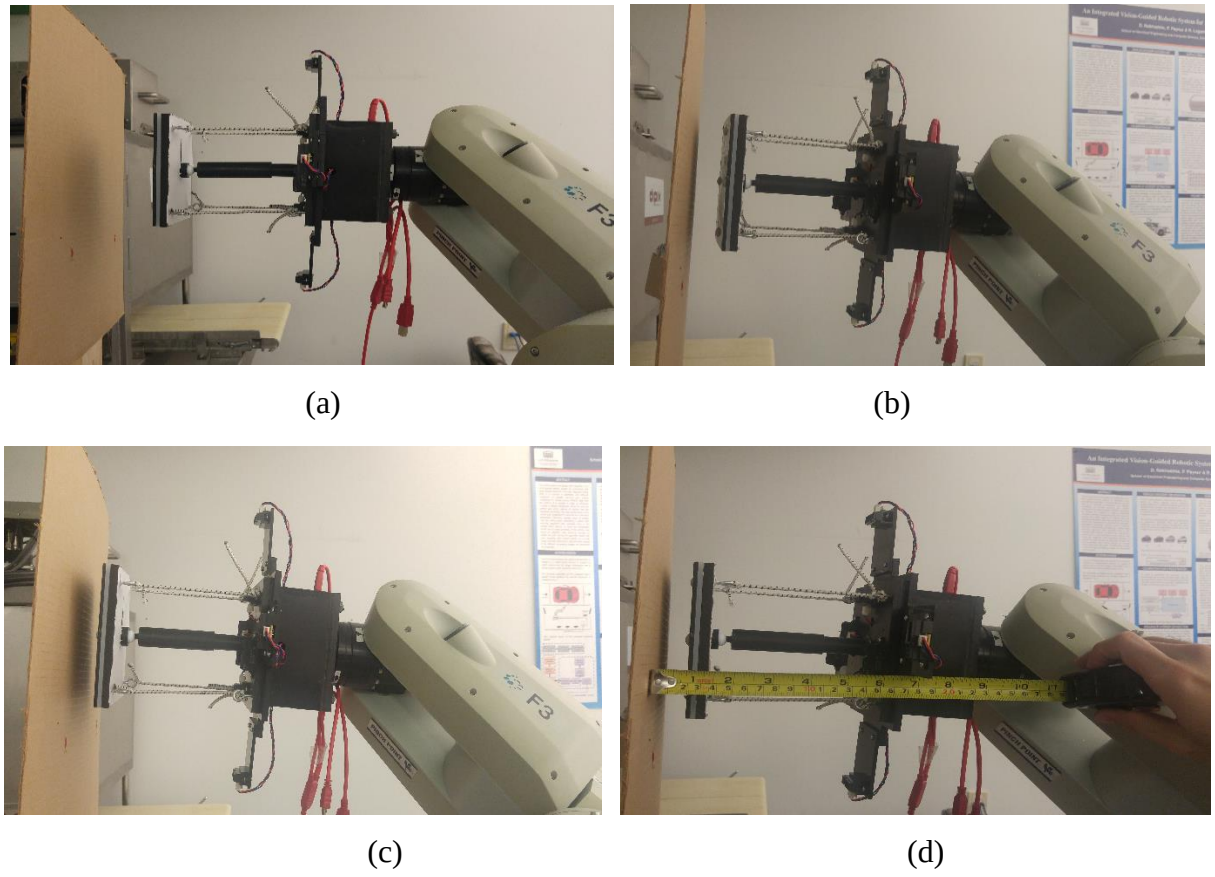
In the first scenario, the robot operates in the *proximity* motion mode only. It does not make any contact with the object but rather maintains a specific distance from it, defined by the user prior to the interaction. As discussed before, the ability of the robot to reactively interact with objects but without reaching contact is critical in modern manufacturing where robots are used for inspection, painting, cleaning and similar tasks. The methodology proposed in this thesis contributes this important additional feature which is not permitted when more traditional force/torque sensors and controls schemes are implemented.

The second scenario is similar but its objective is to evaluate the system performance in the *contact* motion mode, where the robot makes physical contact with the object surface and remains in contact with it while adjusting and finely matching the end-effector position and orientation with the surface. Among many different applications of object-robot physical interaction that involve contact, we can refer to surface following, particles collection, welding and polishing.

4.2.1 Experiment 1: Position Matching with a Static Surface in Proximity Motion Mode

In the first scenario a planar surface is positioned parallel to and at a particular distance from the robot end-effector. The object is fixed and does not move during the interaction. The purpose of the scenario is to only evaluate the accuracy of the proposed adaptive position controller, its stability and convergence rate while correcting the end-effector position and compensating for any deviations from the desired set point defined by user.

In the case reported here, the robot should maintain a distance of 20 mm from the object surface, and since the robot is not to make any contact with the object during the interaction, the proximity motion mode is activated and the external sensors mounted on the compliant wrist provide the required distance information to the controller. Since it is only desired to verify the position controller performance, the wrist is initially positioned parallel to the surface and the desired orientation deviation is set to 0 degrees which means having the end-effector's pointing direction aligned with the normal to the surface. The fuzzy controller presented in section 3.3.1.1 generates the adaptation gain (λ) and the position control signal (Æ) that defines the amount of movement towards the object. An example of the resulting proximity position adjustments is shown in Figure 4.2.



(a) (b)

(c) (d)

Figure 4.2: Proximity interaction with only position adjustment: a) initial pose; b) first iteration; c) second iteration; and d) third and final iteration.

Figure 4.2a shows the relative pose of the compliant wrist with respect to the surface of a planar object before attempting to match the robot's pose to that of the object's surface. The robot moves towards the object and progressively (Figure 4.2b-d) refines its pose over the object and after three iterations the error reaches the desired set point (Figure 4.2d).

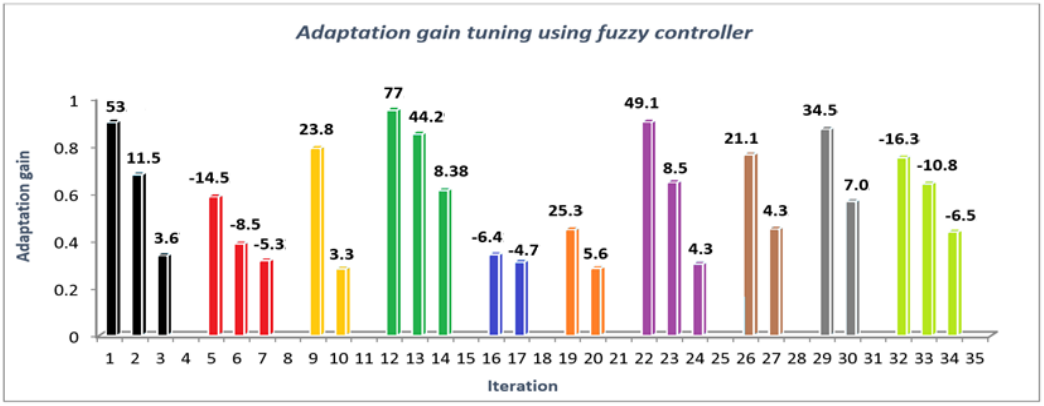
The corrections performed by the adaptive position controller to correct and adjust the robot pose for each configuration during interaction are detailed in Table 4.2 (values in mm) for the test case illustrated in Figure 4.2. At the first configuration, the average distance from the planar surface is 73 mm. Since the desired distance from the object is set to 20 mm, the initial position error is 53 mm. In order to correct the deviation, the adaptive position controller is activated and the control parameters are tuned based on the error value at each configuration. At the first configuration, the adaptation gain ($\lambda = 0.9$) is generated by the fuzzy controller based on the position error and the position control signal ($Æ = 45.02\ mm$) indicates that the robot should move towards the object to compensate the error. In the next configurations, the adaptation gain and therefore the position control signal both get smaller, as expected, when the robot gets closer to the set point after each iteration. The position control signal is generated until the stop condition is satisfied and there is no change in distance error.

Table 4.2: Adaptive position controller performance, proximity mode.

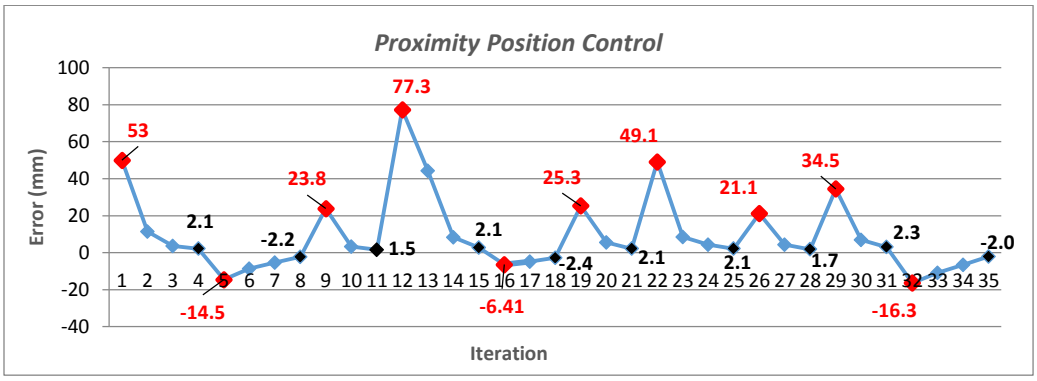
Configuration	Initial Distance	Initial position error (mm)	Final position error (mm)	Position adaptation gain (λ)	Position control signal (Æ)
1	73	53	11.5	0.9	45.02
2	31.5	11.5	3.6	0.679	7.83
3	23.6	3.6	2.1	0.339	1.24

A series of 10 position matching processes were examined where the object is positioned in 10 random positions with different initial distance error from the robot (the first 3 steps correspond to the case detailed above and to values reported in table 4.2). The position control parameters are tuned based on the distance error at each iteration (depicted on top of each bar in Figure 4.3a). The fuzzy adapter presented in section 3.3.1.1 tunes the adaptation gain properly for each scenario (shown in different colors) and the proposed controller generates the position control signal which results in fast but smooth error reduction. The evolution of the position error for all 10 pose adjustments performed solely in the proximity control mode is shown in Figure 4.3b. The red

markers and labels show the initial error at the beginning of each position adjustment process, and the blue markers show how the position error reduces after each iteration, and eventually converges toward the desired set point (black markers).



(a)



(b)

Figure 4.3: Position error corrections during 10 consecutive trials (red dots) of proximity position adjustment: (adaptation gain value and remaining position error over each bar, and b) evolution of position error with respect to set point at each iteration.

4.2.2 Experiment 2: Position and Orientation Matching with a Static Surface in Proximity Motion Mode

The second scenario is similar to the previous one, except that this time the orientation of the object surface is also changed for each test run. As a result, both translation and orientation deviations from the desired set points must be corrected for simultaneously while the robot attempts to match the end-effector's position and orientation with that of the object surface. The object otherwise remains static for the duration of each position and orientation matching process.

An example of the simultaneous position and orientation adjustment process in *proximity* control mode is shown in Figure 4.4. When there are translation and orientation errors between the robot and desired set points, both the position and orientation controllers are activated simultaneously to compute the respective adaptation gains and generate control signals to reposition the compliant wrist using the data provided by the external sensors only on the compliant wrist. The pose adjustment's iterations and the results obtained with the proposed adaptive position and orientation controllers are detailed in Table 4.3 and Table 4.4 for the test case illustrated in Figure 4.4. At the first configuration, the initial distance from the planar surface is 41.6 mm and since the desired distance and orientation from the object are again set to 20 mm and 0 degree respectively, the initial position error is 21.6 mm and the absolute orientation deviation, e^{norm}, (Eq. 3.37) is 0.327 rad (Figure 4.4a). The position adaptation gain (λ) is tuned by the proposed fuzzy controller based on the distance error, and the orientation tuning gains ($\eta_\theta \; and \; \eta_\varphi$) are generated based on the roll (r_S^θ) and pitch (r_S^φ) orientation errors (Eq. 3.38) using the exponential decrease function (Eq. 3.37).

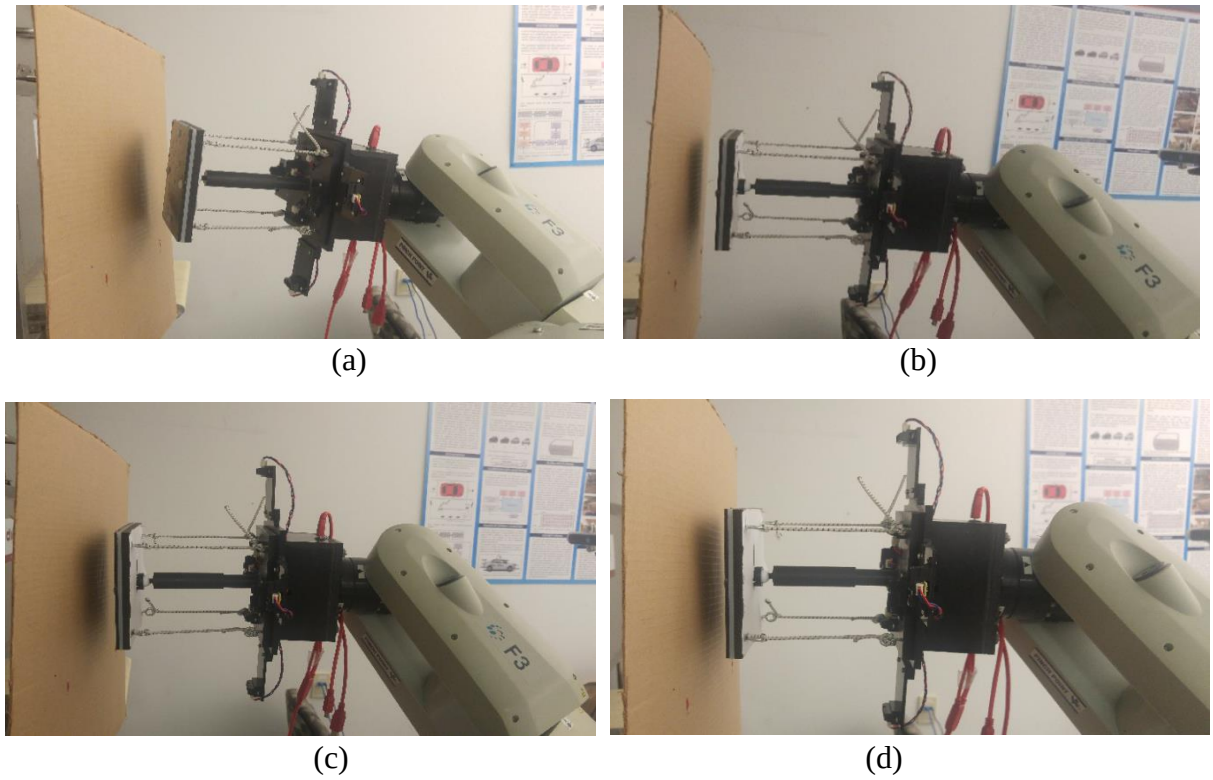


Figure 4.4: Proximity interaction with position and orientation adjustments: a) initial pose; after b) first iteration; c) second iteration; and d) third and final iteration.

Table 4.3: Adaptive position controller performance, proximity mode.

Configuration	Initial Distance	Initial position error (mm)	Final position error (mm)	Position adaptation gain (λ)	Position control signal ($Æ$)
1	41.6	21.6	8.6	0.76	16.65
2	28.6	8.6	4.8	0.411	3.77
3	24.8	4.8	2.6	0.33	1.60

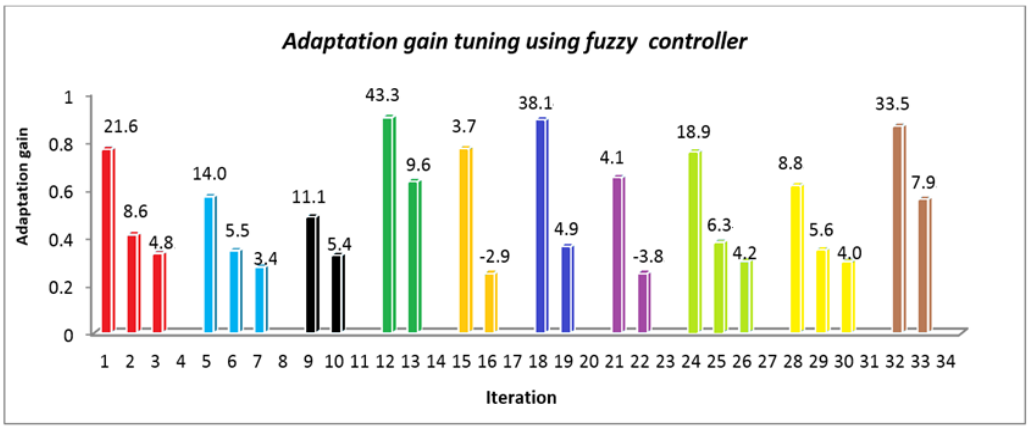
Table 4.4: Adaptive orientation controller performance, proximity mode.

Configuration	Initial error (rad)	Final error (rad)	Orientation error, r_S^{Θ} (rad)	Orientation error, r_S^{φ} (rad)	Orientation adaptation gain (η_{Θ})	Orientation adaptation gain (η_{φ})	Orientation control signal (r_A^{Θ})	Orientation control signal (r_A^{φ})
1	0.32	0.09	0.18	-0.26	0.892	0.959	0.167	-0.258
2	0.09	0.02	-0.01	-0.08	0.448	0.978	-0.006	-0.087
3	0.02	0.01	-0.02	-0.01	0.968	0.830	-0.025	-0.011

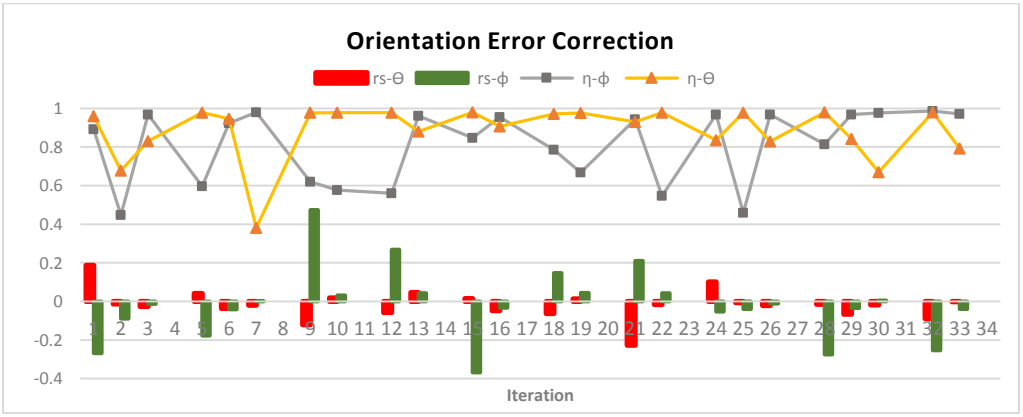
The adaptation gains and the pose control signals are larger when the error is large and get smaller when the error decreases as expected. However, near the equilibrium orientation point, $e^o \to 0$ (Eq. 3.34), the orientation tuning gains switch back to a larger gain when the orientation error becomes very small, which results in a faster convergence. The robot moves towards the object and progressively (Figure 4.4b-d) refines its pose to match that of the object surface. After three iterations, the final end-effector pose reaches the desired set points, while preserving the system stability and ensuring safe interaction.

A series of 10 position and orientation matching processes using only proximity motion mode were performed and closely examined. The tuning of the adaptation gains with the proposed fuzzy controller (Eq. 3.23) and exponential decrease function (Eq. 3.37) for the *proximity* interaction experiments is shown in Figure 4.5, as a sequence of individual operations (the first 3 steps correspond to the case detailed above and to values reported in Table 4.3 and Table 4.4). The position control parameters are tuned based on the distance error at each iteration (depicted on top of each bar in Figure 4.5a) and the orientation adaptation gains ($\eta_{\Theta}, \eta_{\varphi}$) and orientation control signals ($r_A^{\Theta}, r_A^{\varphi}$) are generated to adjust the end-effector orientation based on the orientation error ($r_S^{\Theta}, r_S^{\varphi}$) at each iteration (Figure 4.5b). In contrast to the position adaptation gain, the adaptation

gains $\eta_\Theta, \eta_\varphi$ **(**grey and orange lines**)** do not always vary proportionally to the orientation error r_S^Θ, r_S^φ (red and green bars). The gains are large when the orientation error is large and get smaller when the error decreases, but when the error becomes very small, the gains get back to larger values in order to reduce the convergence time and oscillation near the equilibrium point. Although the orientation adaptation gains do not vary linearly with the angular error value, it is observed that the orientation control signals r_A^Θ, r_A^φ do vary proportionally with and change based on the orientation errors. The behavior observed through these experiments demonstrate the feasibility and efficiency of the proposed adaptive orientation controller.



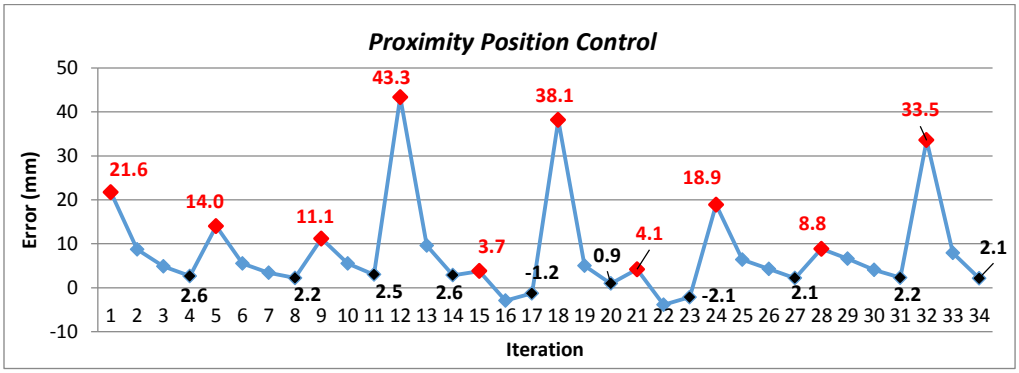
(a)



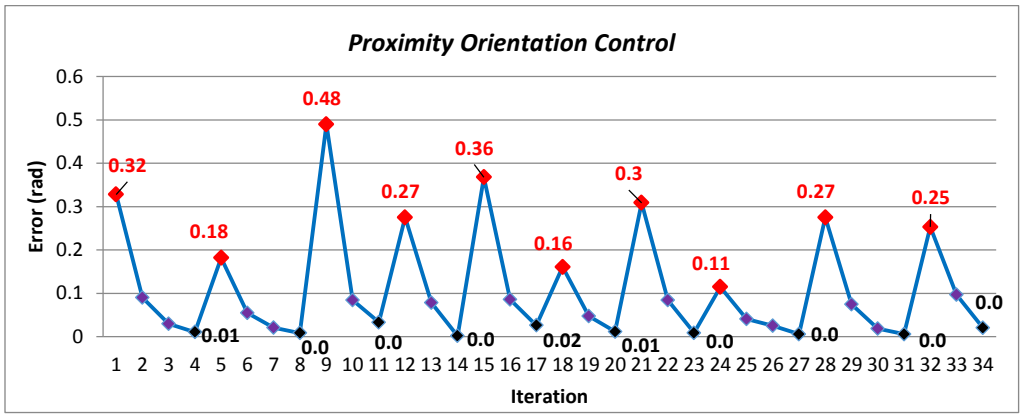
(b)

Figure 4.5: Proximity position adjustment during 10 consecutive trials: a) position adaptation gain value and remaining position error over each bar, and b) rotation gains self-tuning (grey and yellow bars) based on the orientation error (red and green bars).

The evolution of the position and orientation errors for all 10 *proximity* control mode pose adjustment process conducted is shown in Figure 4.6. The red markers show the initial position and orientation error, respectively, at the beginning of each pose adjustment process. The blue markers show how the position and orientation errors is progressively and smoothly reduced after each iteration, and eventually converge toward set point as expected, with black markers depicting final residual errors.



(a)



(b)

Figure 4.6: Position and orientation error corrections over 10 consecutive pose adjustment processes with the proximity control mode.

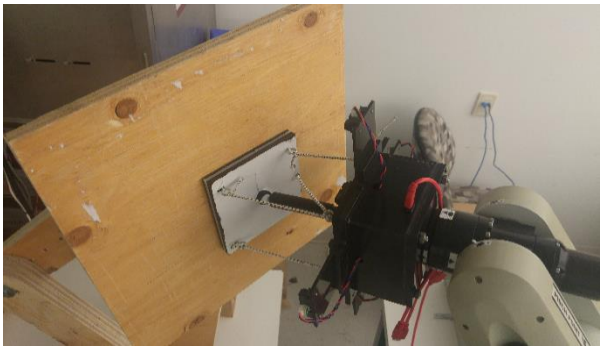
Analysis of the two proximity interaction results show that the number of iterations for each pose adjustment process varies based on the initial distance and orientation from the desired set point. However, it can be slightly different for some cases due to the sensor's measurement variations that influence the system performance especially near the convergence point. The average final position error from the set point over the 10 runs is 2.1 mm for the first experiment and 2.06 mm

for the second experiment. It should be noted that the distance error values are reported by the compliant wrist. The actual final error was measured by a metric ruler after each adjustment. The actual mean final position error for both experiments was about 1.8 mm and the standard deviation was 0.42 mm. The final orientation error after each adjustment was very small so it was not feasible to perform ground truth measurements on those angles. The mean orientation error reported by the compliant wrist over the 10 runs is 0.007 rad and the standard deviation is 0.001 rad.

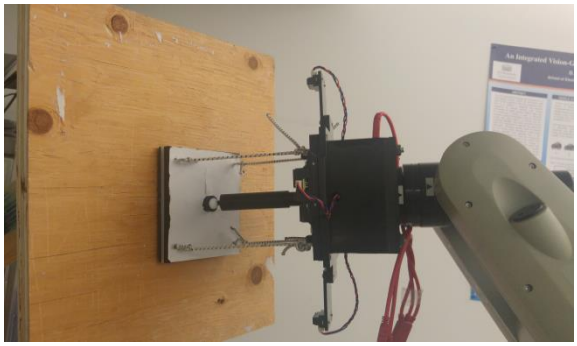
4.2.3 Experiment 3: Position and Orientation Matching with a Static Surface in Contact Motion Mode

The purpose of the third experiment is to evaluate the system performance while operating in the proposed *contact* control mode, that is when the robot end-effector makes contact with the surface of an object and should remain in contact with it while adjusting and matching its position and orientation with that of the surface of the object. The internal sensors of the compliant wrist are used to detect the contact when the compliant plate of the wrist is deflected. In order to make sure that the robot is perfectly in contact with the object and detects the contact via its internal sensors during the interaction, the desired distance to the object is set to -3 mm and the desired orientation is set to 0 rad. The negative value represents the compliant plate deflection amount toward the fixed plate of the wrist due to the force imposed by the object.

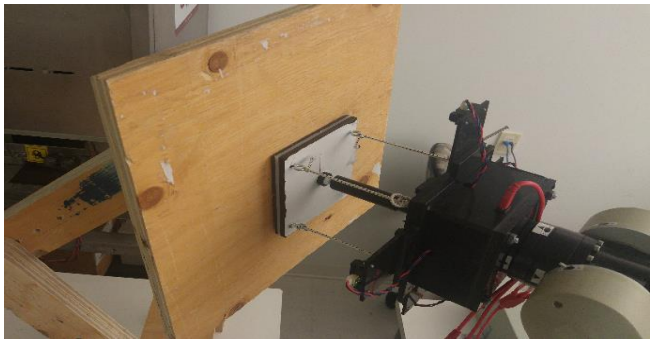
Figure 4.7a shows the initial pose of the robot end-effector and the planar surface considered for this experiment before attempting to match the robot's pose to that of the object's surface. Figure 4.7b and Figure 4.7c illustrate the robot relative position and orientation after the first and second iterations as the manipulator performs its movement up to the desired pose. The results obtained with the proposed adaptive position and orientation controllers while operating with a physical contact are reported in Table 4.5 and Table 4.6 for the test case shown in Figure 4.7. The tables show the control parameters used to define the robot pose at each iteration.



(a)



(b)



(c)

Figure 4.7: Contact interaction: pose at a) initial configuration after contact is reached, b) after first iteration, and c) after second and final iteration.

Table 4.5: Adaptive position controller performance.

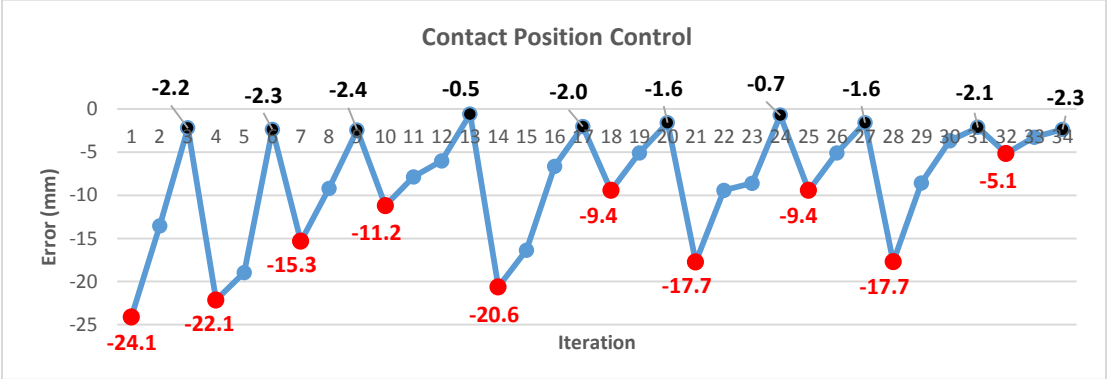
Iteration	Initial position error (mm)	Final position Error (mm)	Position adaptation gain (λ)	Position control signal (Æ)
1	-24.1	-10.5	0.792	-19.11
2	-10.5	-2.20	0.458	-9.05

Table 4.6: Adaptive orientation controller performance.

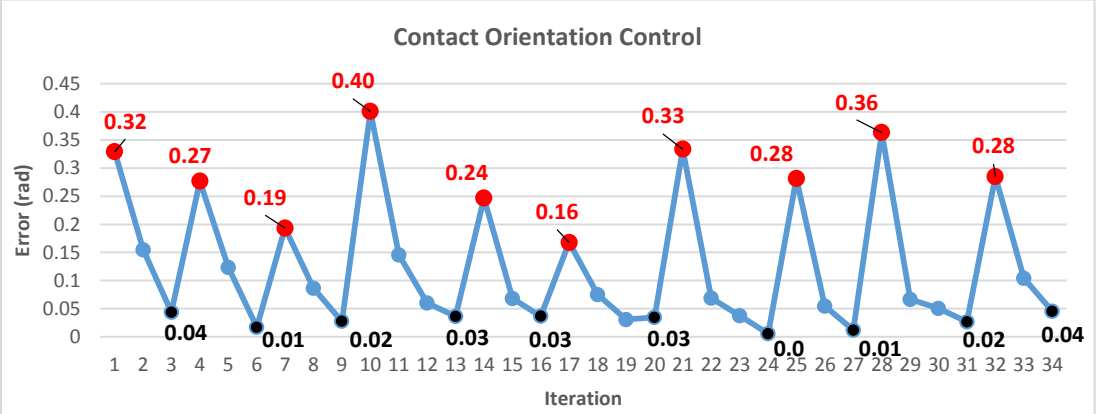
Iteration	Initial error norm	Final error norm	Orientation error, r_S^{Θ} (rad)	Orientation error, r_S^{φ} (rad)	Orientation adaptation gain (η_{Θ})	Orientation adaptation gain (η_{φ})	Orientation control signal (r_A^{Θ})	Orientation control signal (r_A^{φ})
1	0.32	0.15	-0.32	-0.03	0.979	0.342	-0.320	-0.012
2	0.15	0.04	-0.10	-0.11	0.928	0.943	-0.097	-0.107

Figure 4.8 shows the system performance for 10 random contact pose adjustments. Similar to the second experiment reported in section 4.2.2, the planar surface position and orientation is changed in between each test case. But it remains static for the duration of the each entire position and

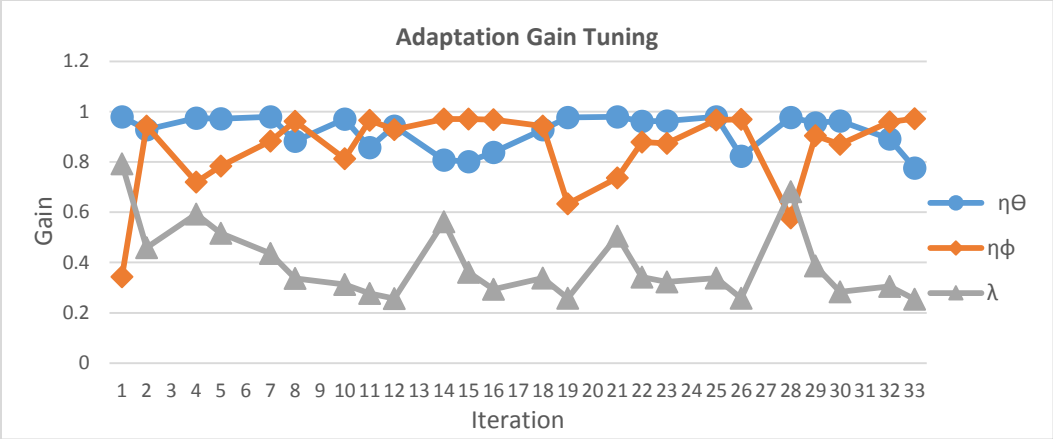
orientation matching process. The initial position and orientation errors at the beginning of each pose adjustment process and the error correction achieved after each iteration are shown by the red markers and blue markers respectively. It can be seen that the control parameters in Figure 4.8c, change based on the online sensory information to make the necessary adjustments. The final position error shown in Figure 4.8a has negative values that confirm the compliant plate deflection (shaft compression) due to the force imposed by the object during the pose adjustment and therefore the ability of the robot to maintain contact after each adjustment as desired. The mean final position and orientation errors from the result obtained by the compliant wrist are -2.09 mm and 0.023 rad. Although the number of iterations and the pose error are influenced by the sensor's signal variability, the robot successfully refines and matches its position and orientation with that of the surface of the object after a two to three iterations only, demonstrating stability and accuracy both in position and in orientation.



(a)



(b)



(c)

Figure 4.8: a) Position and b) orientation error corrections during 10 contact pose adjustments, with c) corresponding adaptation gains.

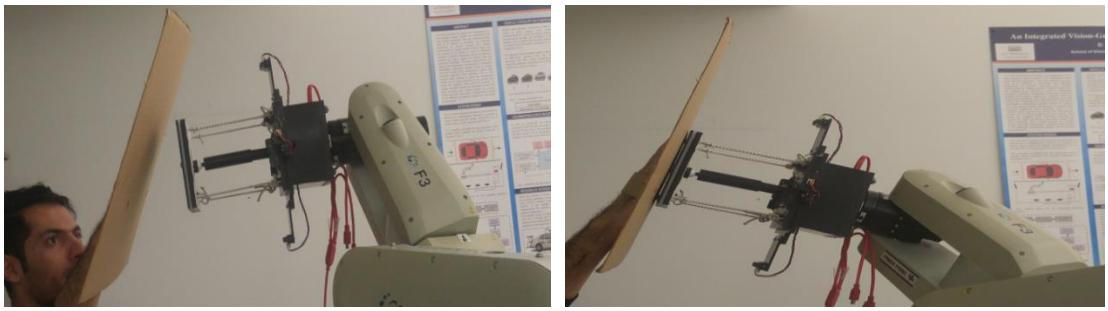
4.3 Dynamic Object-Robot Interaction

This fourth experiment validates the proposed hybrid switching control scheme that integrates and ensures smooth switching between five different motion modes. In order to test the controller up to its full capability, as it adapts the robot configuration to that of a moving target in 3D space, an individual holds a planar surface in front of the robot and moves it frequently to random positions and orientations in an unpredicted way, while remaining within the workspace of the robot manipulator. The robot is required to adapt its pose to continuously match the end-effector's position and orientation with that of the dynamic object (with a desired distance between the end-effector and the surface set to 0mm, and an orientation relative to the surface also set to 0 rad).

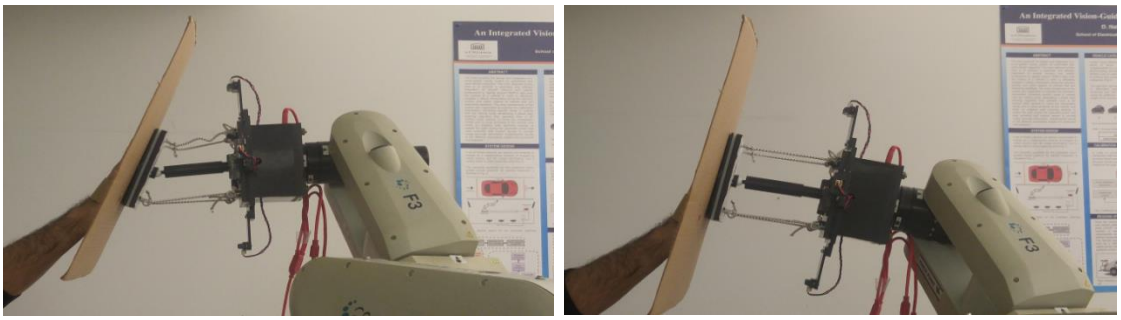
The entire sequence of operation that is described in chapter 3, from RGB-D data acquisition (section 3.2.1), user selection of the object of interest (section 3.2.2), target retrieval using the compliant wrist's embedded sensors (section 3.5), as well as position and orientation controllers (section 3.3), and the hybrid switched controller (section 3.4) must repeatedly transition in between different interaction modes from *free* motion (Figure 4.9a), to *proximity* (Figure 4.9b), and to *contact* (Figure 4.9c), and vice versa. The experiment begins by monitoring the behavior of the manipulator robot when there is no object in front of the robot. First, the RGB-D data is acquired by the Kinect and the RGB image is presented to the operator. The 3D information about the object

of interest is extracted from the corresponding area of the point cloud when the operator clicks on the object in the RGB image. The *free* motion mode is activated (Figure 4.9a) and the robot moves towards the selected object. It should be noted that in this experiment it is not desired to explore the object surface but rather react and follow the object's movement dynamically. Therefore the Kinect information is only used to guide the robot towards the object at the beginning of the procedure. The object may move during the approach phase while it is not yet detected by all the external sensors embedded in the compliant wrist. Therefore, the retrieval algorithm (section 3.5) takes control of the robot movement to guide the robot in the proper direction for all the external sensors to eventually detect the object before any interaction occurs. Once the object is detected by the four external IR sensors, depending on the object's position, a specific motion mode is assigned by the hybrid controller for the end-effector to begin its dynamical interaction with the object.

(a)



(b)

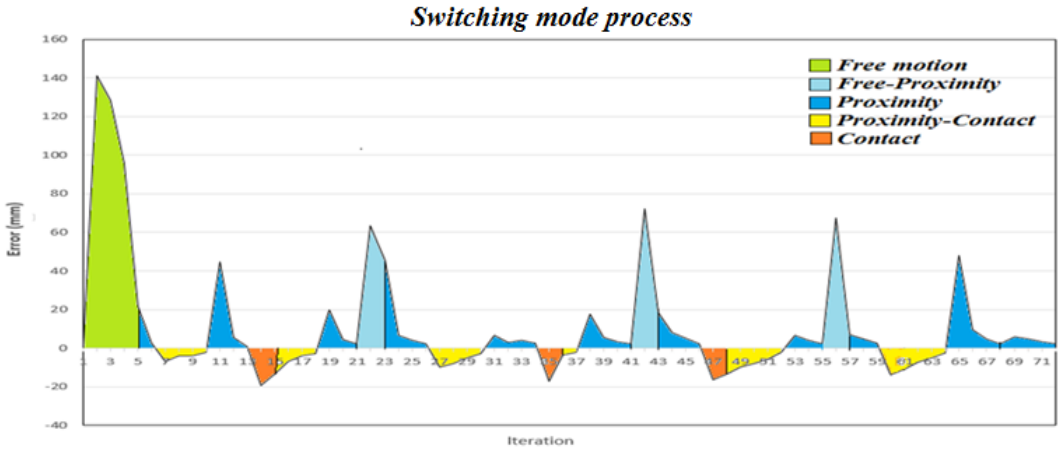


(c)

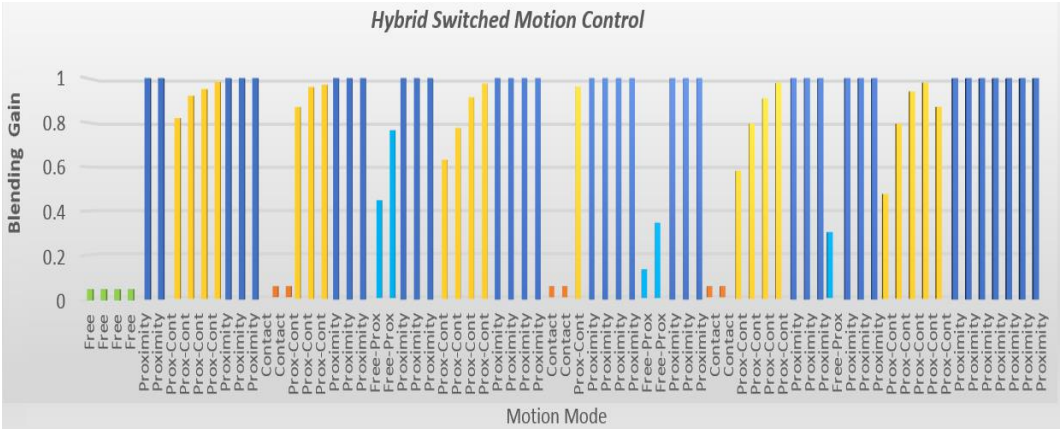
Figure 4.9: Complete process of robot-object interaction with a moving target surface: a) object detection and localization, b) proximity interaction, and c) contact interaction.

As shown in Figure 4.10a for one of the test sequences performed, the position and orientation signals are continuously computed and the hybrid controller switches smoothly between all interaction modes using the data provided by the RGB-D sensor for *free* motion, external compliant wrist sensors for *proximity* interaction, and internal compliant wrist sensors for *contact* interaction. The *blend* motion modes are activated when it is possible for data derived from the different sensors to be combined and to ensure smooth transition from one of the three main interaction modes to another. The blending gain (β) proposed in section 3.4 is used for gain scheduling. As shown in Figure 4.10b, the blending gain value is 0 for *free* motion (depicted in green) and *contact* (depicted in orange) modes, is 1 for *proximity* (blue) mode, and it varies between 0 and 1 to blend the three main motion modes according to the proposed blending weight distribution (Figure 3.14), as desired. In the first phase (depicted in green) the robot moves toward the object (*free* motion). Since at the beginning of the interaction, the object is not completely detected by the external sensors (Figure 4.10a), the error is large but the robot adjusts its position using the retrieval algorithm (section 3.5). The error distance decrease corresponds to the object coming into the line of sight of each of the embedded external infrared sensors. When the object is detected by all four sensors the interaction mode switches automatically and smoothly to the *proximity* mode, or to any other interaction mode based on the online sensory information, to best align the robot end-effector pose with that of the object surface (Figure 4.9b-c). The peaks of error visible in Figure 4.10a happen when the object moves to a new position, as manipulated by an individual, whether it is operating in the *proximity* (Figure 4.9b) or the *contact* (Figure 4.9c) control mode. In all cases, the proposed hybrid-adaptive switched control scheme is successful at recovering the new pose of the target

object and making the end-effector efficiently and smoothly adapt to the new configuration. As shown in Figure 4.10a, the errors on position and orientation are systematically reduced and the deviation in between the end-effector and the surface converges toward zero as a result of effective switching between the different modes of operation considered.



(a)



(b)

Figure 4.10: Hybrid switching control system performance: a) position error, and b) blending gain on position controller.

4.4 Surface Following

In the fifth experiment, it is desired to closely follow the surface of a curved object (Figure 4.11) and to explore the object surface completely while maintaining contact of the end-effector with the surface and accommodating its curvature accurately. This experiment illustrates particularly well

the context of development for the proposed solution, that is particles collection by smoothly wiping surfaces with arbitrary shape for automated security screening. The experiment reported here is also representative of alternative scenarios, including robotic inspection, welding, waxing, polishing, or printing. In the present case, the object considered is a real automotive door panel. The region of interest to be fully covered corresponds to the entire area located below the window line. The upper window area is not considered for these experiments since glass is not imaged well with Kinect sensor technology, and also the robot cannot reach the window area. The desired distance between the contact compliant plate of the end-effector and the surface of the object at any location is set to -3 mm. Parallel alignment between the contact compliant plate of the end-effector and the local orientation of the surface is imposed by setting 0 rad as a constraint on orientation. As shown in Figure 4.11 which shows the entire experimental setup, the Kinect sensor is located behind the robot and the latter moves on a two meter linear track to wipe the full width of the door panel surface.

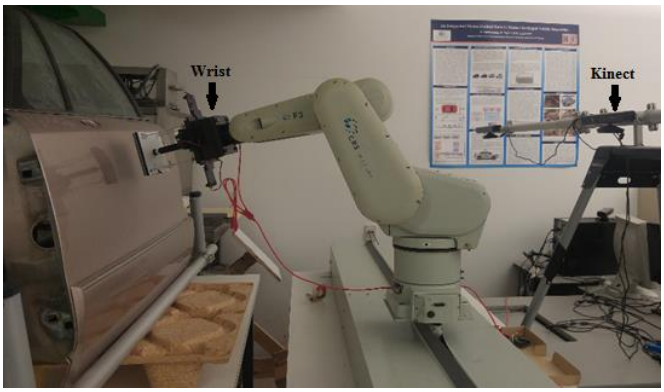
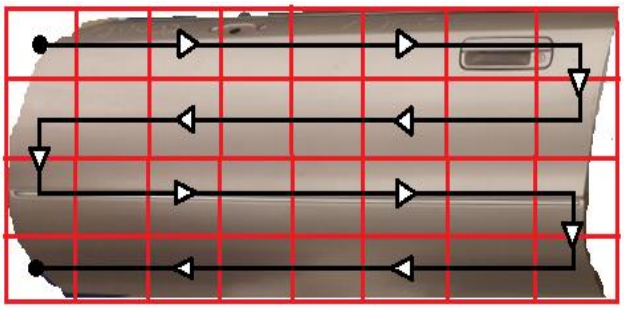


Figure 4.11: Door set up with respect to the robot and the Kinect sensor.

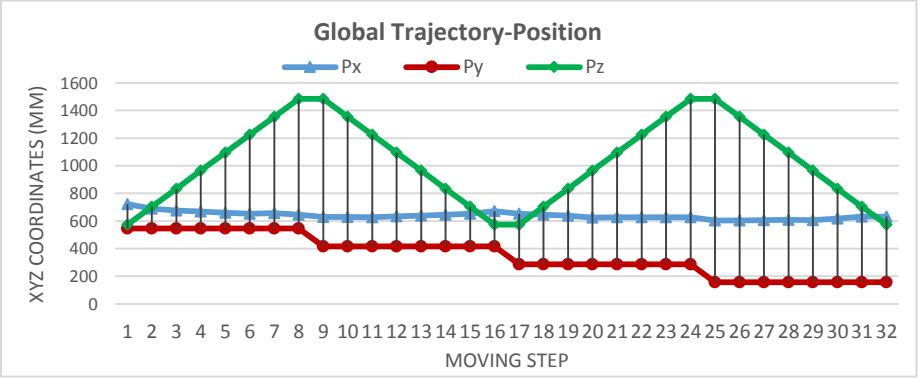
The Kinect sensor collects raw 3D information on the environment. The RGB image captured by the Kinect (Figure 4.12a) is presented to the operator. The object of interest is selected by the operator and the corresponding part of the point cloud is automatically extracted from the scene, as shown in Figure 4.12b. The entire region that is segmented is then decomposed to a set of uniform cells. These are the same size as the compliant plate of the wrist (127x127 mm) which determines the size of the steps for trajectory planning. A global trajectory is then generated using the proposed coverage path planning method (section 3.1) to ensure complete coverage of the surface while scanning over the object of interest's surface. The door panel size is 1018x509 mm, and the global trajectory consists a set of 32 points (32 occupied cells) that determine the successive end-effector

positions $[p_F^x, p_F^y, p_F^z]$ and orientations $[r_F^\theta, r_F^\varphi, r_F^\psi]$ with respect to the robot frame (Figure 4.12c,d) to ensure full coverage of the object surface.

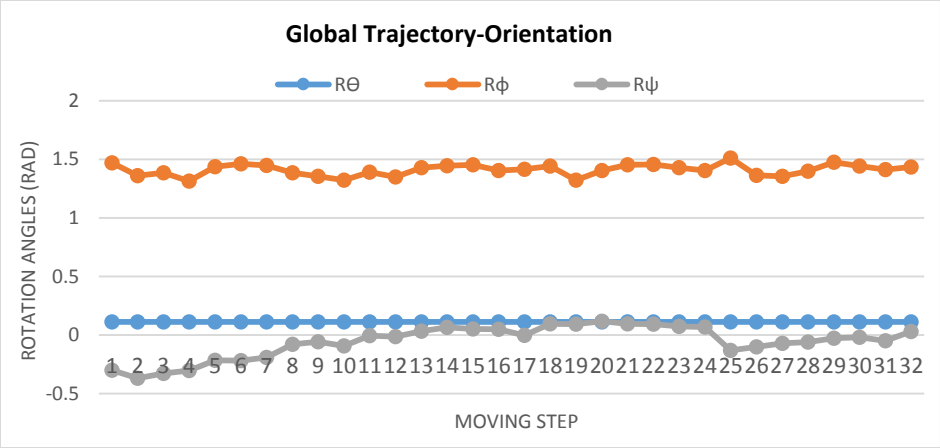
(a)



(b)



(c)



(d)

Figure 4.12: a) A real automotive door panel, b) trajectory planning to ensure full coverage over the region of interest, c) global trajectory: position, and d) global trajectory: orientation.

Once the global trajectory is generated, the *free* motion mode is initially activated to guide the robot towards the start point. When the robot reaches in close proximity of the object, the *proximity* (external) sensors embedded on the compliant wrist detect the object, the switching signal (δ) is determined using Eq. 3.48 and the switching supervisor selects a control mode according to Eq. 3.49. Therefore, the motion control mode is switched to the *proximity* mode where the adaptive position and orientation controllers adjust the robot pose (Eq. 3.45) using the external sensors in preparation for a safe contact with the object. To ensure safe contact with the object and smooth switching between the *proximity* mode and the *contact* mode, the online information provided by the internal and external sensors are combined (Eq. 3.49) and the *blend* mode is activated. When contact happens, the control mode switches to the *contact* mode only and the adaptive controllers refine the robot end-effector's position and orientation over the object using the internal sensory information (Eq. 3.46) and maintain contact with the object during the interaction. The robot then moves to the next pose defined by the global trajectory and the same process as above repeats for each moving step until the robot reaches out to all the 32 points (cells) and fully explores the panel (Figure 4.13).

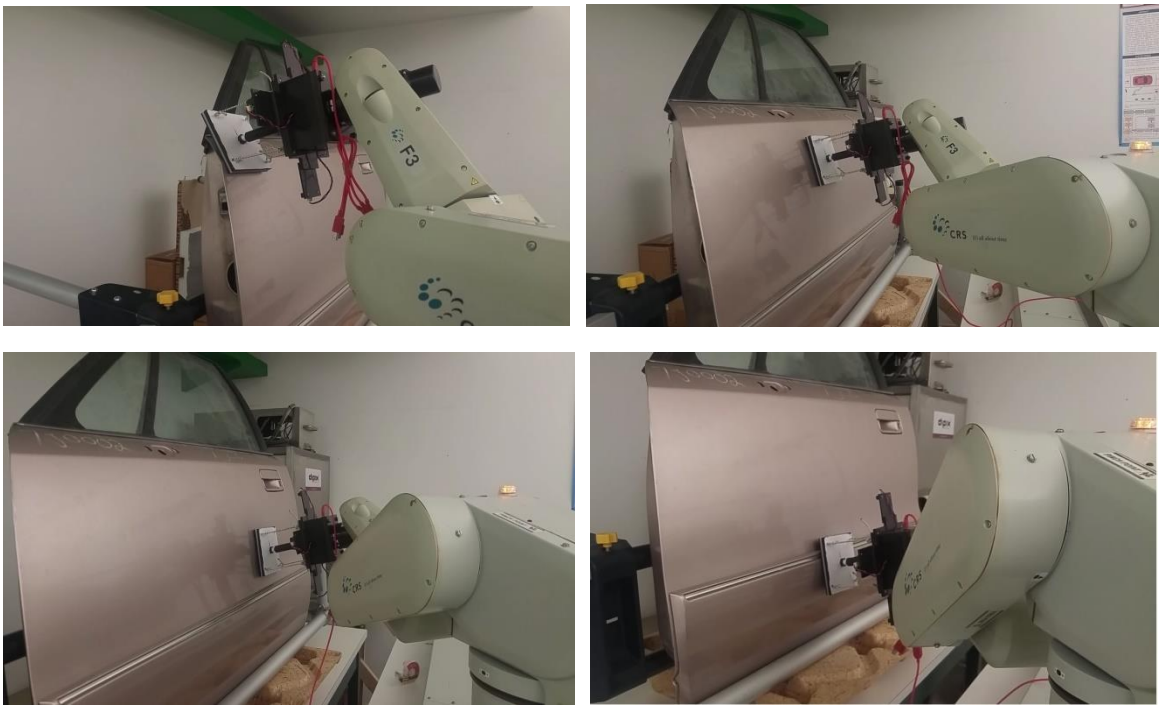
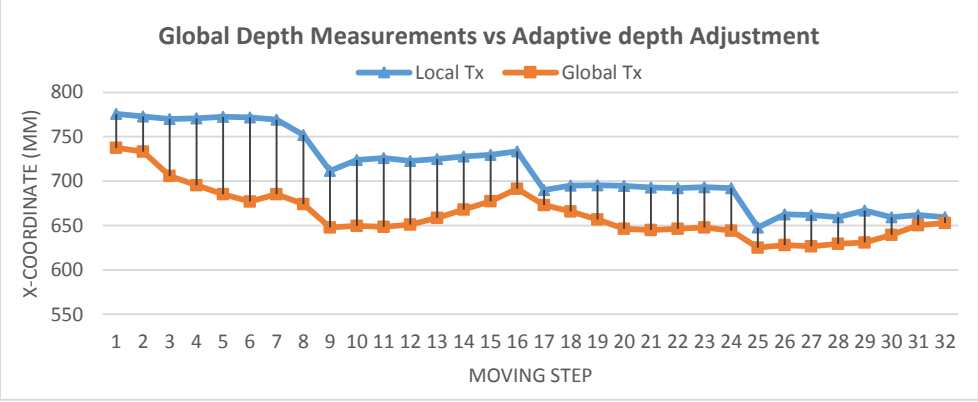


Figure 4.13: Illustration of accurate match between end-effector and curved object surface at selected configurations over a curved automotive door panel.

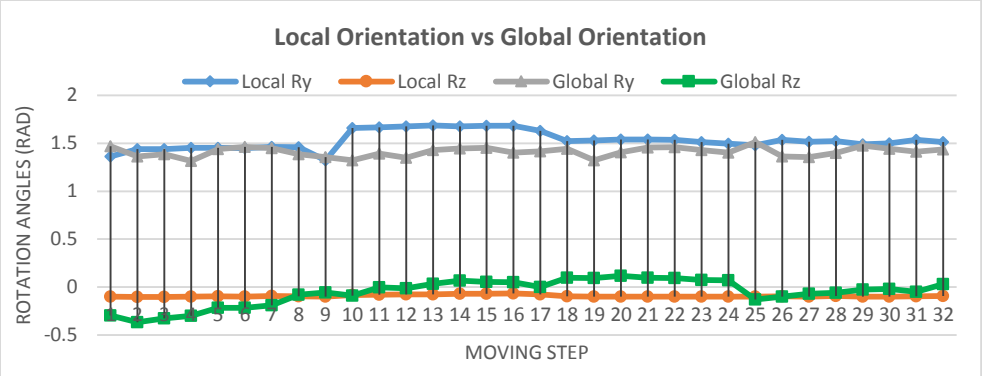
Figure 4.14 shows the adaptive controllers performance in refining the global trajectory during the interaction process. The depth adjustment over the object surface using the adaptive position controllers (depicted in blue) is shown in Figure 4.14a, and compared with the depth measurements estimated from the RGB-D data (depicted in orange). In other words, the orange curve represents the pre-planned trajectory for the end-effector, while following the sequence depicted in Figure 4.14b. It is noticeable that as the door panel is slightly inclined, as can be seen in Figure 4.13, the bottom part, scanned at the end of the trajectory, is closer to the Kinect sensor. As such, distance is progressively reduced as the robot progresses through the 32 moving steps. The robot pose over the object surface varies according to change of the door curvature. When the robot moves horizontally the change in depth (corresponding to the X axis in the robot's reference frame) is small because the curvature of the door in the scan line is consistent, but when the robot moves vertically on the object surface the depth changes more abruptly due to the curvature of the door being more prominent along the vertical direction. The robot moves vertically three times at steps 9, 17 and 25. Conversely, the blue curve represents the same trajectory but as it was refined and actually performed under the more accurate guidance of sensors embedded in the compliant wrist, especially

the internal ones as contact needed to be maintained throughout the experiment. The distance along the depth axis is increased on all locations with respect to that of the orange trajectory. This means that the end-effector reached further, and closer to the door panel, than what was initially planned from the approximate surface location estimates extracted from the Kinect sensor depth map. The general trends of distance variations over the pre-planned and actual trajectories remain consistent. This demonstrates that the original approximate trajectory supports adequately the process of surface following and ensures complete coverage, while the embedded sensing stage provided by the compliant wrist allows for a much more accurate match of the end-effector with the local surface position and orientation at all locations, while also ensuring smooth and stable approach phases and potential recovery of contact. Such a behavior goes beyond what more traditional force/torque control schemes can provide, as the proposed integrated sensing and control framework closely and seamlessly monitors the surface and controls the robot over the entire interaction period, from far away from the surface and up to full contact with it.

The orientation correction is represented in Figure 4.14b, the local object surface orientation with respect to the robot base frame around the y-axis (in grey) and z-axis (in green) estimated as part of the offline path planning process are also updated locally using the online sensory information (in blue and orange) from the compliant wrist. As the results show, the global trajectory is also refined locally in orientation, further refining the complete exploration of the door panel while accommodating its inherent curvature.



(a)

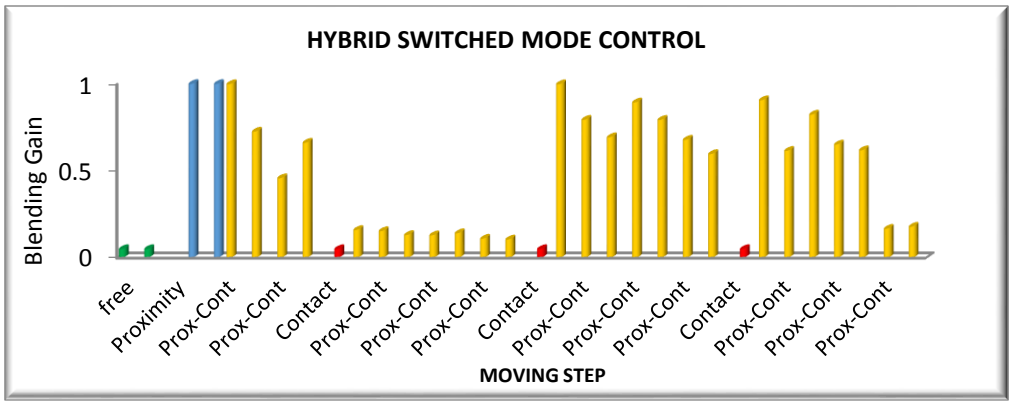


(b)

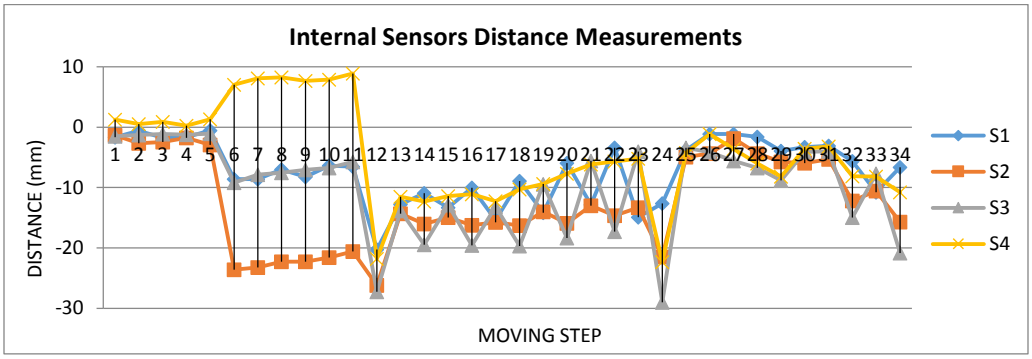
Figure 4.14: Hybrid controller performance in refining the robot a) position and b) orientation.

The switching mode control graph (Figure 4.15a) shows the switching process between different interaction modes for this task. Referring to Eq. 3.51, the system switches between the proposed modes and the blending gain changes based on the robot distance from the object as desired. Once the robot detects the object completely (*free* motion), the interaction mode switches to *proximity*. Once the contact happens, the motion mode switches between the *contact* mode and the *blend* (Proximity-Contact) mode. As mentioned, in this experiment it is desired to follow and slide on the object surface while maintaining contact. For this purpose, the desired distance from object is set to -3 mm. According to Eq. 3.48 and Eq. 3.49, the best results are obtained when the sensory information of internal and external sensors are combined and the system operates in the *blend* mode (Figure 4.15a).

When the robot makes contact with the object, the compliant plate deflection is detected by the internal sensors (when there is no contact, the compliant plate is in an equilibrium condition and the internal sensors return a value of zero). As Figure 4.15b shows the compliant plate has been deflected during the task after a few moving steps (*free* and *proximity* motion) at the beginning of the task and remained deflected, which confirms that the robot maintained contact during the surface following. The sensors S1, S2, S3 and S4 (four internal infrared sensors (Figure 3.9b)) return negative values where the compliant plate is deflected (compressed).



(a)



(b)

Figure 4.15: a) Activated modes at each step by the hybrid switched control system, b) internal sensors dataset, for the fifth test scenario provided full surface following over an automotive door panel.

Analysis of all the different experiments results demonstrate the accuracy, performance and stability of the adaptive controllers in refining the robot pose over the object using the online sensory information. The precision achieved for position and orientation was within ±3 mm and ±0.05 rad respectively. The results obtained in [181] in relation with the development of the instrumented compliant wrist reveal that the pose measurement estimated by the compliant wrist sensors can reach up to ±2.4 mm on the error in distance and ±0.03 rad deviation in rotation, depending on the object pose with respect to the robot. Therefore, it is credible to associate the major sources of error in the proposed control scheme to the embedded sensing stage in the compliant wrist, as the performance of the proposed controllers is commensurate with the sensors accuracy levels.

4.5 Discussion and Comparison with Other Methods

Although much of the literature reviewed in preparation for this work produced interesting and promising results, a direct comparison of results could not be made due to variation in robots, objects and the novel sensing device used for our experiments. However, the capabilities of the proposed approach in data collection, adaptability, decision making and switching control is compared with previous works in this section.

The problem of surface following (tracking) has been addressed by model-based methodologies [116-119] frequently where the object model is assumed to be known a priori. In these approaches force/torque sensors are mainly used to align the end-effector with the object surface and to adjust the contact point by devising adaptive rules based on the robot dynamic model. However, these approaches rely heavily on the predefined object model and the robot dynamics and are therefore not appropriate when there are structural uncertainties in the robot or object model. As discussed in section 2.1, in order to achieve a safe and reliable interaction with objects of different shapes, applying appropriate sensors to model the objects and update local information to refine the accuracy of the representation plays a significant role to enhance the speed, accuracy and performance of the overall robotic application. Various sensors have been used for object modelling (section 2.1.3.3), such as 3D profiling cameras, scanners, sonars, laser or combinations of them. However, these sensors are expensive and acquisition is usually lengthy and requires the processing of massive amounts of information. Alternatively, in this work, the problem was addressed by integrating only a low cost but fast Kinect sensor to collect 3D information on the scene and generate a primary global trajectory for the manipulator to scan and explore the entire surface of an object of interest through using a 3D occupancy grid. To the best of our knowledge, no approach has been proposed for complete surface coverage robot path planning on objects of arbitrary shape, without requiring a detailed model of the said object. In the former surface following approaches, the trajectory was generated using a predefined object model [116-119], or the object model was constructed using online sensory information as the robot moved along the surface in a random way, but no global trajectory was constructed [197, 198] to ensure complete coverage. In some other works, only the contours of the object of interest [199] or predefined contact points [200, 201] were considered and followed rather than the entire surface.

In order to enable industrial robots to interact with the environment and offer flexibility, adaptable compliance has been incorporated to industrial robots. Our review of the traditional and recent robotics technologies (section 2.2) shows that in many cases the modifications are technically challenging, not feasible and require hardware modifications, retrofitting, and new components which result in high costs. We address the flexibility problem by using a low cost custom-designed compliant wrist mounted on the robot end-effector to increase the robot mechanical compliance and reduce the risk of damages, both to the robot and the environment. Using additional sensors for global/local object modelling and incorporating compliancy to the rigid body of standard robots are crucial but not sufficient for safe and flexible interaction with objects in presence of uncertainties. To achieve reliable reactive interaction in such circumstances, adaptive controllers have been proposed. Most studies on adaptive control (section 2.3) for object-robot interaction tasks use force/torque sensors to adjust the robot pose over the object. The adaptive laws are designed based on the manipulator dynamic and contact characteristics. The controller's performance relies heavily on the robot and environment model and is sensitive to modeling errors. Another major limitation of these approaches is that the controllers are only applicable where there is contact between the robot and the object. In this work, two original model-free self-tuning adaptive controllers were proposed to facilitate robot reactive interaction and surface following tasks with or without contact. The adaptive controllers are designed to control the robot's position and orientation respectively using online sensory information provided by the infrared (IR) sensors embedded in the compliant wrist. The adaptive controllers indirectly solve the force control problem in the object-robot interaction in the form of a position control problem where the IR sensors provide the required information for the interaction instead of touch sensors (force/torque, tactile and haptic). In contrast to the other works (section 2.3), they neither require precise mathematical model of the robot and the environment, nor force/torque calculation and learning procedure. Moreover, using two different controllers allows the decoupling of the translational and rotational control loops that result in minimum convergence time, minimum oscillation, and higher accuracy.

Combined together, the proposed global trajectory planning and adaptive control enable the robot to interact with an object in different motion modes under specific sensory guidance. In order to perform more complex tasks and interact with dynamic environments, it is essential to integrate,

combine and switch instantaneously between the control modes because change in the environment happens at unforeseen instants. Most of the previously designed hybrid systems (section 2.4) only use vision (proximity) and force/torque sensors for position and force control where the free motion is handled by the vision (proximity) sensors and force/torque sensor is used to adjust the contact force. Only few works proposed multi-phase control [202, 203]. These approaches use event based (hard) switching scheme in which the controller switches between the phases when certain conditions are satisfied. In contrast to previous works, the proposed hybrid switched control system employs hard and blend switches by taking advantage of logistic functions that lead to smooth transitions and weighted combinations of the different motion modes to fully leverage the sensory stages embedded on the compliant wrist to detect, refine and interact with an arbitrary located and shaped object of interest. Moreover, the proposed adaptive controllers and the hybrid switched approaches are general, in the sense that their concept and formulation are not restricted to the instrumented compliant wrist considered in this research. The embedded sensors scan be substituted by other type of sensors providing distance measurement and the control schemes can readily be integrated in other robotic control systems where interaction with surfaces or surface following is expected.

In addition, comparison of the proposed hybrid switched control system for surface following and interaction with previous works not involving a compliant wrist [173], only using direct sensory information from the compliant wrist but without adaptive controllers [181], applying a classical fuzzy controller [204], and also a self-tuning fuzzy-PID controller [205], shows that incorporating the compliant wrist sensing stage increases the flexibly and accuracy of the performed task.

When the data acquired by the compliant wrist sensors are directly sent to the robot controller [181], the robot oscillate a lot near the set points and convergence tends to be slow. To improve the performance, a fuzzy controller was applied [204] to react adaptively to the object movement and continuously match the end-effector's position and orientation with that of the object. Although the fuzzy controller enables the robot to react fast to the changes, it had weak performance in reducing the oscillation and was not very accurate and stable.

In [205], a self-tuning fuzzy PID controller was applied to control the robot pose over an object (the real car door). As the results show, the self-tuning fuzzy-PID controller provides a fast response to changes and minimizes the error better than the fuzzy controller [204]. However, the controller is very sensitive to changes in the amplitude of the error and PID algorithms do not guarantee optimal control or system stability. In addition, as discussed in section 3.3.2, in order to improve the convergence and reduce the oscillation problem near the desired set points, decoupling the position and orientation controllers is suggested but setting up the fuzzy rule base for tuning the position and orientation adaptation gains is not efficient because it requires a larger number of membership functions, rules and variables and extensive computation time. The proposed hybrid-adaptive switched control scheme addresses those limitation and experiments demonstrate that the weighted combination of four control modes benefitting from the fusion of multi-sensory information results in a faster convergence rate, higher accuracy and stable interaction with an arbitrary object surface.

4.6 Chapter Summary

This chapter presented the experimental results and evaluation of the proposed coverage path planning, adaptive position and orientation controllers as well as the original hybrid switched control scheme, which were tested individually and in an integrated manner on different scenarios. The results from the experiments validate the efficiency, accuracy and performance of the approach in close interaction with static or dynamic objects, and with and without contact. Experimentation also demonstrated the robot's ability to successfully follow, both in position and orientation, the entire curved surface of an object of interest. The two original model-free self-tuning adaptive controllers that are introduced in this thesis were able to dynamically tune the adaptation gains, and generate position and orientation control signals to react to an object's surface configuration or to its movements in order to guarantee safe and stable pose adjustment. In addition, five alternative motion modes were proposed, applied and integrated within the original proposed switching control scheme which utilizes a mixture of hard and blend switches, unlike state-of-the-art literature. It is demonstrated that the hybrid switch approach is capable to supervise the robot motion continuously and smoothly switch between different motion modes when it finds the robot within certain distance ranges from the object. These original concepts can easily be adapted to any robotic and sensing mechanisms, as well as to a variety of robotic applications.

Chapter 5. CONCLUSION

5.1 Summary

Following an extensive review of the related works, the pros and cons of different methods have been discussed and existing problems of the control schemes for adaptive and flexible object-robot interaction and surface following were identified. In order to take advantage of the various control techniques, a hybrid control architecture under multi-sensory guidance is proposed to perform an accurate and reliable interaction with and surface following of objects of arbitrary shape. The proposed hybrid system is a combination of deliberative (global) and reactive (local) path planning methods.

The motion planning task is divided into 3 main interaction modes of *free, proximity* and *contact* motion modes. Each of the three motion control modes uses specific sensory information to guide the robot in different regions of the workspace based on the object of interest's location and orientation. The *free* motion mode operates when the robot is far from any object and it is moving through free space until it reaches in close proximity of an object of interest. In the solution developed in this thesis, this mode is fed with information from a Kinect sensor that provides an approximate 2.5D model of the object of interest from which a global trajectory is generated with the goal to provide full coverage of a defined portion of the surface. This mode manages the robot motion during the approach phase. Once the end-effector reaches in close proximity to the surface, the *proximity* mode takes over to smoothly control the manipulator movement over the object surface where the robot pose is calculated according to the local information provided by an array of infrared sensors embedded on the compliant wrist. The *contact* mode is later on activated when the robot touches the objects surface, and then relies on sensory information from a second array of infrared sensors on the compliant wrist.

Local information made available at and surrounding the contact point supports the development of an innovative form of reactive motion control that provides a manipulator with the capabilities of closely following the surface curvature while maintaining contact with the object. The proposed control scheme includes two unique self-tuning adaptive controllers that refine and stabilize the

robot pose over the object surface. However, using only a specific set of sensors and event-based (hard) switching control system tends to cause chattering, oscillation and system instability. Therefore, a novel hybrid switched control system is proposed to combine the sensory information and switch smoothly between the interaction modes by blending the main motion modes, resulting in a fourth *blend* motion mode. The proposed method was implemented and validated through experimental test runs with a 7-DOF robotic manipulator equipped with the custom instrumented compliant wrist considered for this research. The experimental results demonstrated that the robot can accurately and stably adapt its position and orientation to the configuration of static or dynamic objects, and to successfully scan an entire region of an object of interest under multimodal sensing while closely following the curved surface of an object with or without contact.

5.2 Contributions

This thesis proposes an original hybrid-adaptive switched motion control system for a reliable reactive and adaptive interaction and surface following of objects with arbitrary surface shape with or without contact using an industrial robot. The multi-stage control system utilizes offline and online trajectory planning and a combination of them to address the flexibility and adaptability problem of standard robots interacting with unstructured environment. The approach differs from the literature in terms of data collection, processing, trajectory planning and decision stages by developing a unique coverage path planning, two original self-tuning adaptive controllers and a hybrid switched control scheme. This offers an original self-standing solution for the important market that represent the rigid manipulators currently at work and the ones that will be installed in the near future. The proposed coverage path planning method ensures complete coverage of the selected surface independently from the contour and surface shape of the object. The adaptive controllers solve the force control problem in form of a position control problem where infrared sensors provide the required information for the interaction instead of touch sensors (force/torque, tactile and haptic). The proposed adaptive controllers require no learning procedure, no precise mathematical model of the robot and the environment and does not rely on force/torque calculation. As such the controllers enable the robot to interact with objects with and without contact. The original hybrid switch scheme which utilizes a mixture of hard and blend switches, integrates and

switches smoothly between the five alternative motion modes to supervise the robot motion continuously.

Development of the system resulted in multiple contributions to the fields of surface coverage path planning, adaptive control, hybrid switching, object (human)-robot interaction, security screening and surface following. The work can also be applied for various industrial processes application such as paint removal, polishing, welding, sanding, sand-blasting, stamping, surface writing or printing. The principal original contributions of this thesis are:

1 Design of a strategy for real-time robot trajectory modification via two unique model-free self-tuning adaptive controllers that react to the object movement and its dynamic changes of configuration to achieve an accurate and reliable close proximity and contact interaction;

2 Design of multi-phase control strategies (free-motion, proximity and contact), governed by a novel hybrid switched control approach that applies a mixture of hard and blend switches to seamlessly and efficiently combine the sensory information from multiple sensors and ensure smooth switching between different interaction modes;

3 Development and implementation of a coverage path planning method using cell decomposition technique to ensure complete coverage scan or exploration of an object of interest;

4 Development of innovative means to automatically collect, detect and extract 2.5D information over an object of interest in the robot workspace using fast but low cost RGB-D sensors.

This research also led to a number of publications [159, 172, 173, 180, 204, 205, 206 and 207] and garnered recognition in the form of a Best paper award at IEEE International Systems Conference (2014), an IEEE award (2015) and Electrical and Computer Engineering award (2016) at the Engineering and Computer Science and Graduate Poster Competition held at the University of Ottawa.

5.3 Future Work

This thesis proposes a new methodology for accurate, reliable and stable interaction with objects of arbitrary shape. Since the approach does not rely on the kinematic/dynamic model of the robot and requires no hardware and expensive mechanical modifications, it can be expanded and easily be implemented using any robot for a multitude of more complex applications. Therefore, there are still many opportunities to improve and extend the scope of this thesis:

- The compliant wrist used for the online trajectory modification currently suffers from some problems that limit its practical application. Its current size and the accuracy of the measurements limit the robot movement and adoption in surface following and object tracking. The maximum and minimum rotational range that the internal and external sensors can detect is $\pm 40^0$ and $\pm 58.5^0$ which implies some restrictions in following surfaces with steep curves.

- An eye-in hand 3D camera (Kinect fusion) can be mounted on the robot to provide live 3D local images to extend the work for some applications such as telerobotics, surgical robots and live 3D surface construction.

- The adaptive controllers and the hybrid switched control scheme rely only on the distance measurement from the compliant wrist. Therefore, a better performance could be achieved by incorporating touch sensors (force/tactile) and improving the adaptive controllers to make them capable to regulate and adjust the robot pose and interaction forces simultaneously.

REFERENCES

[1] "World Robotics Report 2016", IFR International Federation of Robotics, 2017. [Online]. Available: https://ifr.org/ifr-press-releases/news/world-robotics-report-2016.

[2] F. Chaumette and S. Hutchinson, "Visual servo control. II. Advanced approaches [Tutorial]", IEEE Robotics & Automation Magazine, vol. 14, no. 1, pp. 109-118, 2007.

[3] H. Kase, N. Maru, A. Nishikawa And F. Miyazaki, "Manipulator Control by Visual Servoing with Stereo Vision", Transactions of the Institute of Systems, Control and Information Engineers, vol. 6, no. 8, pp. 360-367, 1993.

[4] Y. Shirai and H. Inoue, "Guiding a robot by visual feedback in assembling tasks", Pattern Recognition, vol. 5, no. 2, pp. 99-108, 1973.

[5] C. Gao, F. Li and X. Xu, "A Vision Open-Loop Visual Servoing", International Conference on Machine Learning and Cybernetics, 2006 pp. 699-703.

[6] H. Fujimoto, "Visual servoing of 6 DOF manipulator by multirate control with depth identification," 42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475), 2003, pp. 5408-5413

[7] L. Natale, F. Nori, G. Sandini and G. Metta, "Learning precise 3D reaching in a humanoid robot", IEEE 6th International Conference on Development and Learning, 2007, pp.324 -329.

[8] W. Chang and C. Wu, "Hand-Eye Coordination for Robotic Assembly Tasks", International Journal of Automation and Smart Technology, vol. 2, no. 4, pp. 301-308, 2012.

[9] F. Chaumette and S. Hutchinson, "Visual Servoing and Visual Tracking", Springer Handbook of Robotics, pp. 563-583, 2008.

[10] E. Bugarin and R. Kelly, "Direct Visual Servoing of Planar Manipulators Using Moments of Planar Targets", Robot Vision, Ales Ude (Ed.), ISBN: 978-953-307-077-3, InTech, 2010.

[11] F. Miyazaki and Y. Masutani, "Robustness of sensory feedback control based on imperfect Jacobian. Robotics Research, "The Fifth International Symposium. H. Miura and S. Arimoto, Eds. Cambridge, MA: MIT Press, pp. 201-208, 1990.

[12] 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.

[13] R. Kelly, R. Carelli, O. Nasisi, B. Kuchen and F. Reyes, "Stable visual servoing of camera-in-hand robotic systems", IEEE/ASME Transactions on Mechatronics, vol. 5, no. 1, pp. 39-48, 2000.

[14] R. Kelly, "Robust asymptotically stable visual servoing of planar robots", IEEE Transactions on Robotics and Automation, vol. 12, no. 5, pp. 759-766, 1996.

[15] Y. Fang, A. Behal, W. Dixon and D. Dawson, "Adaptive 2.5D visual servoing of kinematically redundant robot manipulators", Proceedings of the 41st IEEE Conference on Decision and Control, 2002, pp. 2860-2865.

[16] E. Zergeroglu, D. Dawson, M. de Querioz and A. Behal, "Vision-based nonlinear tracking controllers with uncertain robot-camera parameters", IEEE/ASME Transactions on Mechatronics, vol. 6, no. 3, pp. 322-337, 2001.

[17] C. Cheah, C. Liu and J. Slotine, "Adaptive Jacobian vision based control for robots with uncertain depth information", Automatica, vol. 46, no. 7, pp. 1228-1233, 2010.

[18] J. de Best, M. van de Molengraft and M. Steinbuch, "Direct dynamic visual servoing at 1 kHz by using the product as 1.5D encoder", IEEE International Conference on Control and Automation, 2009, pp. 361-366.

[19] G. Silveira and E. Malis, "Direct visual servoing with respect to rigid objects", IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007, pp.1963-1968.

[20] G. Silveira and E. Malis, "Direct Visual Servoing: Vision-Based Estimation and Control Using Only Nonmetric Information", IEEE Transactions on Robotics, vol. 28, no. 4, pp. 974-980, 2012.

[21] D. Schuurman and D. Capson, "Robust Direct Visual Servo Using Network-Synchronized Cameras", IEEE Transactions on Robotics and Automation, vol. 20, no. 2, pp. 319-334, 2004.

[22] B. Tamadazte, G. Duceux, N. Le-Fort Piat and E. Marchand, "Highly precise micropositioning task using a direct visual servoing scheme", IEEE International Conference on Robotics and Automation, 2011, pp. 5689-5694.

[23] A. Sanderson, "Image-based visual servo control using relational graph error", in Proceedings of the IEEE International Conference on Cybernetics and Society, Washington, DC, USA, 1980, pp. 1074-1077.

[24] L. Deng, "Comparison of image-based and position-based robot visual servoing methods and improvements", A PHD dissertation of Electrical and Computer Engineering of University of Waterloo, 2003.

[25] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches", IEEE Robotics & Automation Magazine, vol. 13, no. 4, pp. 82-90, 2006.

[26] E. Marchand, A. Rizzo, and F. Chaumette, "Avoiding robot joint limits and kinematic singularities in visual servoing". In Proceedings of the 13th International Conference on Pattern Recognition, Vienna, Austria, 1996, pp. 297-301.

[27] F. Chaumette, "Potential problems of unstability and divergence in image-based and position-based visual servoing," 1999 European Control Conference (ECC), Karlsruhe, 1999, pp. 4549-4554.

[28] Sining Liu, W. Xie and Chun-Yi Su, "Image-Based Visual Servoing using improved image moments", International Conference on Information and Automation, 2009, pp. 577-582.

[29] F. Chaumette, "Image Moments: A General and Useful Set of Features for Visual Servoing", IEEE Transactions on Robotics, vol. 20, no. 4, pp. 713-723, 2004.

[30] O. Tahri and F. Chaumette, "Point-based and region-based image moments for visual servoing of planar objects", IEEE Transactions on Robotics, vol. 21, no. 6, pp. 1116-1127, 2005.

[31] J. Pomares, F. Chaumette and F. Torres, "Adaptive Visual Servoing by Simultaneous Camera Calibration", Proceedings of IEEE International Conference on Robotics and Automation, 2007, pp. 2811-2816.

[32] T. Sato and J. Sato, "Visual servoing from uncalibrated cameras for uncalibrated robots", Systems and Computers in Japan, vol. 31, no. 14, pp. 11-19, 2000.

[33] Y. Mezouar and F. Chaumette, "Path planning for robust image-based control", IEEE Transactions on Robotics and Automation, vol. 18, no. 4, pp. 534-549, 2002.

[34] P. Corke and S. Hutchinson, "A new partitioned approach to image-based visual servo control", IEEE Transactions on Robotics and Automation, vol. 17, no. 4, pp. 507-515, 2001.

[35] E. Malis, "Visual Servoing Invariant to Changes in Camera-Intrinsic Parameters", IEEE Transactions on Robotics and Automation, vol. 20, no. 1, pp. 72-81, 2004.

[36] C. Pérez, N. García-Aracil, J.M. Azorín, J. M. Sabater, L. Navarro and R. Saltarén, "Image-Based and Intrinsic-Free Visual Navigation of a Mobile Robot Defined as a Global Visual Servoing Task, " Informatics in Control, Automation and Robotics II, pp. 87-93, 2007.

[37] E. Cervera, A.P. Del Pobil, F. Berry and P. Martinet, "Improving image-based visual servoing with three-dimensional features", *Int. J. Rob. Res*, vol. *22*, pp. 821-839, 2003.

[38] E. Cervera, and P. Martinet, "Combining pixel and depth information in image-based visual servoing", In Proceedings of the IEEE International Conference on Advanced Robotics, 1999, pp. 445-450.

[39] A. De Luca, G. Oriolo and P. Giordano, "On-Line Estimation of Feature Depth for Image-Based Visual Servoing Schemes", Proceedings of IEEE International Conference on Robotics and Automation, 2007, pp. 2823-2828.

[40] J. Pages, C. Collewet, F. Chaumette and J. Salvi, "Plane-to-plane positioning from image-based visual servoing and structured light", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2004, pp. 1004-1009.

[41] J. Pages, C. Collewet, F. Chaumette and J. Salvi, "Optimizing plane-to-plane positioning tasks by image-based visual servoing and structured light", IEEE Transactions on Robotics, vol. 22, no. 5, pp. 1000-1010, 2006.

[42] J. Pages, C. Collewet, F. Chaumette and J. Salvi, "Robust decoupled visual servoing based on structured light", IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005, pp. 2676-2681.

[43] J. Pages, C. Collewet, F. Chaumette and J. Salvi, "An approach to visual servoing based on coded light", IEEE International Conference on Robotics and Automation, 2006, pp. 4118-4123.

[44] W. Wilson, C. Williams Hulls and G. Bell, "Relative end-effector control using Cartesian position based visual servoing", IEEE Transactions on Robotics and Automation, vol. 12, no. 5, pp. 684-696, 1996.

[45] P. Wira and J.-P. Urban, "A new adaptive Kalman filter applied to visual servoing tasks," Knowledge-Based Intelligent Engineering Systems and Allied Technologies. Proceedings. Fourth International Conference 1, pp.267-270, 2000.

[46] V. Lippiello, B. Siciliano and L. Villani, "Position-Based Visual Servoing In Industrial Multi-Arm Robotic Cells Using Multiple Cameras", IFAC Proceedings Volumes, vol. 39, no. 15, pp. 43-48, 2006.

[47] F. Janabi-Sharifi and M. Marey, "A Kalman-Filter-Based Method for Pose Estimation in Visual Servoing", IEEE Transactions on Robotics, vol. 26, no. 5, pp. 939-947, 2010.

[48] P. Martinet, N. Daucher, J. Gallice and M. Dhome. "Robot control using 3D monocular pose estimation," IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, vol. 4, 1997, pp. 1–12,.

[49] G. Morel, T. Liebezeit, J. Szewczyk, S. Boudet and J. Pot, Explicit incorporation of 2D constraints in vision based control of robot manipulators, Lecture Notes in Control and Information Sciences, vol. 250, pp 99-108, 2000.

[50] B. Thuilot, P. Martinet, L. Cordesses and J. Gallice, "Position based visual servoing: keeping the object in the field of vision", IEEE International Conference on Robotics and Automation, 2002, pp. 1624-1629.

[51] G. Chesi, K. Hashimoto, D. Prattichizzo and A. Vicino, "Keeping Features in the Field of View in Eye-In-Hand Visual Servoing: A Switching Approach", IEEE Transactions on Robotics, vol. 20, no. 5, pp. 908-913, 2004.

[52] T. Murao, H. Kawai, and M. Fujita. "Predictive visual feedback control with eye-in/to-hand configuration via stabilizing receding horizon approach", In Proc. of the 17th IFAC World Congress on Automatic Control, 2008, pp. 5341–5346.

[53] V. Lippiello, B. Siciliano and L. Villani, "A Position-Based Visual Impedance Control for Robot Manipulators", IEEE International Conference on Robotics and Automation, 2007, pp.2068-2073.

[54] N. Gans, A. Dani and W. Dixon, "Visual servoing to an arbitrary pose with respect to an object given a single known length", American Control Conference, 2008, pp.1261-1267.

[55] R. Herrejon, S. Kagami and K. Hashimoto, "Composite visual servoing for catching a 3-D flying object using RLS trajectory estimation from a monocular image sequence", IEEE International Symposium on Computational Intelligence in Robotics and Automation - (CIRA), 2009, pp. 665-670.

[56] Y. Wang, J. Thunberg and X. Hu, "A transformation of the Position Based Visual Servoing Problem into a convex optimization problem", IEEE Conference on Decision and Control (CDC), 2012, pp. 5673-5678.

[57] T.M, Du and H.J. Kang, "Fusion of Vision and Inertial Sensors for Position-Based Visual Servoing of a Robot Manipulator", Lecture Notes in Computer Science. 7995, pp 536-545, 2013.

[58] P. Martinet and E. Cervera, "Stacking Jacobians properly in stereo visual servoing", IEEE International Conference on Robotics and Automation, 2001, pp. 717-722.

[59] E. Cervera, F. Berry and P. Martinet, "Is 3D useful in stereo visual control?", IEEE International Conference on Robotics and Automation, 2002, pp. 1630-1635.

[60] E. Cervera, F. Berry and P. Martinet, "Stereo visual servoing with oriented blobs. In Proceedings of the 11thInternational Conference on Advanced", Robotics, 2003, pp. 977-982.

[61] G. Recatala, P.J. Sanz, E. Cervera and A.P. Del Pobil, "Filter-based control of a gripper-to-object positioning movement", In Proceedings of the IEEE International Conference on Systems, Manand Cybernetics, 2004, pp. 5423-5428.

[62] L. Pari, J.M. Sebastian, C. Gonzalez and L. Angel, "Image based visual servoing: A new method for the estimation of the image jacobian in dynamic environments", In Proceedings of the 3rdInternational Conference on Image Analysis and Recognition, 2006, pp. 850-861.

[63] J.M. Sebastian, L. Pari, and L. Angel, Traslosheros, A. "Uncalibrated visual servoing using the Fundamental matrix", Robot. Auton. Syst. vol. 57, pp. 1-10, 2009.

[64] C. Teuliere and E. Marchand, Direct 3d servoing using dense depth maps. In Proceedings of IROS'12, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012, pp. 1741-1746.

[65] P. Rakprayoon, M. Ruchanurucks and A. Coundoul, "Kinect-based obstacle detection for manipulator," System Integration (SII), IEEE/SICE International Symposium, 2011, pp. 68-73.

[66] I. Siradjuddin, L. Behera, T. M. McGinnity and S. Coleman, "A position based visual tracking system for a 7 DOF robot manipulator using a Kinect camera," The International Joint Conference on Neural Networks (IJCNN), 2012, pp. 1-7.

[67] G. J. García, P. Gil1, D. Llácer and F. Torres, "Guidance of Robot Arms using Depth Data from RGB-D Camera", Proceedings of the 10th International Conference on Informatics in Control, Automation and Robotics, 2013, pp. 315-321.

[68] J. Gamez, A. Robertsson, J. Gomez and R. Johansso, "Improvement of Force Control in Robotic Manipulators Using Sensor Fusion Techniques", Robot Manipulators, 2008.

[69] J. N. Pires, J. Ramming, S. Rauch, and R. Araújo, "Force/torque sensing applied to industrial robotic deburring", Sensor Review, vol. 22, no. 3, pp.232 – 241, 2002.

[70] L. Villani and J. De Schutter, Force control. In Handbook of Robotics; Siciliano, B. & Khatib, O., Eds.; Springer: Berlin, Germany, pp. 161-185, 2008.

[71] F. Almeida, A. Lopes and P. Abreu, Force-Impedance Control: a New Control Strategy of Robotic Manipulators, Recent Advances in Mechatronics (Kaynak, O., Tosunoglu, S. & Ang Jr M., Eds.). Springer-Verlag, Singapore, pp. 126-137, 1999.

[72] L. Biagiotti, H. Liu, G. Hirzinger and C. Melchiorri, "Force-Impedance Control: a new control strategy of robotic manipulators," Proceedings of the IEEE/RSJ, Intl. Conference on Intelligent Robots and Systems, pp. 1-12, 2003.

[73] I. Bonilla, F. Reyes, M. Mendoza and E. González-Galván, "A Dynamic-compensation Approach to Impedance Control of Robot Manipulators", Journal of Intelligent & Robotic Systems, vol. 63, no. 1, pp. 51-73, 2010.

[74] M. Namvar and F. Aghili, "Adaptive force-motion control of coordinated robots interacting with geometrically unknown environments", IEEE Transactions on Robotics, vol. 21, no. 4, pp. 678-694, 2005.

[75] S. Kilicaslan, M. Özgören and S. Ider, "Hybrid force and motion control of robots with flexible links", Mechanism and Machine Theory, vol. 45, no. 1, pp. 91-105, 2010.

[76] A. Jafari, J. Ryu, M. Rezaei, R. Monfaredi, A. Talebi and S. Ghidary, "An adaptive hybrid force/motion control design for robot manipulators interacting in constrained motion with unknown non-rigid environments", IEEE ISR 2013, 2013.

[77] H.N. Nguyen and D. Lee, "Hybrid force/motion control and internal dynamics of quadrotors for tool operation", IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp. 3458-3464.

[78] G. Garcia, J. Corrales, J. Pomares and F. Torres, "Survey of Visual and Force/Tactile Control of Robots for Physical Interaction in Spain", Sensors, vol. 9, no. 12, pp. 9689-9733, 2009.

[79] E. M. Petriu, "Biology-inspired multimodal tactile sensor system,"IEEE International Symposium on Robotic and Sensors Environments (ROSE), 2011, pp. 54-59.

[80] M.R. Cutkosky, R.D. Howe and W.R. Provancher, "Force and tactile sensors", In Handbook of Robotics; Siciliano, B., Oussama, K., Eds., &Springer-Verlag: Berlin, Germany, pp. 455-476, 2008.

[81] R.S. Dahiya and M. Valle, "Tactile Sensing: Definitions and Classification, Robotic Tactile Sensing", pp.13-17, 2013.

[82] A. Berger and P. Khosla, "Using Tactile Data for Real-Time Feedback", The International Journal of Robotics Research, vol. 10, no. 2, pp. 88-102, 1991.

[83] R. Fearing, "Tactile Sensing Mechanisms", The International Journal of Robotics Research, vol. 9, no. 3, pp. 3-23, 1990.

[84] R. M. Murray, Z. Li and S.S. Sastry, "A Mathematical Introduction to Robotic Manipulation", CRC, 1994.

[85] B. Kane, M. Cutkosky and G. Kovacs, "A traction stress sensor array for use in high-resolution robotic tactile imaging", Journal of Microelectromechanical Systems, vol. 9, no. 4, pp. 425-434, 2000.

[86] A. Okamura and M. Cutkosky, "Feature Detection for Haptic Exploration with Robotic Fingers", The International Journal of Robotics Research, vol. 20, no. 12, pp. 925-938, 2001.

[87] M. Charlebois, K. Gupta and S. Payandeh, "On estimating local shape using contact sensing", Journal of Robotic Systems, vol. 17, no. 12, pp. 643-658, 2000.

[88] J. Galvez, P. Gonzalez de Santos and F. Pfeiffer, "Intrinsic tactile sensing for the optimization of force distribution in a pipe crawling robot", IEEE/ASME Transactions on Mechatronics, vol. 6, no. 1, pp. 26-35, 2001.

[89] S. Omata, Y. Murayama and C. Constantinou, "Real time robotic tactile sensor system for the determination of the physical properties of biomaterials", Sensors and Actuators A: Physical, vol. 112, no. 2-3, pp. 278-285, 2004.

[90] V. Maheshwari, "High-Resolution Thin-Film Device to Sense Texture by Touch", Science, vol. 312, no. 5779, pp. 1501-1504, 2006.

[91] J. Yuji and C. Sonoda, "Simultaneous Sensing of Static Contact Pressure and Temperature using a PVDF Film Sensor", The Proceedings of JSME annual Conference on Robotics and Mechatronics (Robomec), 2006, pp. 738-741.

[92] P. Payeur, C. Pasca, A. Cretu and E. Petriu, "Intelligent Haptic Sensor System for Robotic Manipulation", IEEE Transactions on Instrumentation and Measurement, vol. 54, no. 4, pp. 1583-1592, 2005.

[93] P. Schmidt, E. Maël and R. Würtz, "A sensor for dynamic tactile information with applications in human–robot interaction and object exploration", Robotics and Autonomous Systems, vol. 54, no. 12, pp. 1005-1014, 2006.

[94] R. Dahiya, M. Valle and G. Metta, "System approach: A paradigm for robotic tactile sensing", 10th IEEE International Workshop on Advanced Motion Control, 2008, pp. 26-28.

[95] P. Mittendorfer and G. Cheng, "Humanoid Multimodal Tactile-Sensing Modules", IEEE Transactions on Robotics, vol. 27, no. 3, pp. 401-410, 2011.

[96] T. E. Alves de Oliveira, A. M. Cretu and E. M. Petriu, "Multimodal Bio-Inspired Tactile Sensing Module," in IEEE Sensors Journal, vol. 17, no. 11, pp. 3231-3243. doi: 10.1109/JSEN.2017.2690898.

[97] E. M. Petriu, W. S. McMath, S. S. K. Yeung and N. Trif, "Active tactile perception of object surface geometric profiles," in IEEE Transactions on Instrumentation and Measurement, vol. 41, no. 1, pp. 87-92, 1992.

[98] U. Chouinard, S. Achiche, T. Bieze, L. Baron and C. Duriez, "Analyzing design modifications effects on the compliance of deformable hybrid serial-parallel manipulators", CCToMM Mechanisms, Machines, and Mechatronics (M3) Symposium, pp.1-12, 2017.

[99] A. Albu-Schäffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck and G. Hirzinger, "The DLR lightweight robot: design and control concepts for robots in human environments", Industrial Robot: An International Journal, vol. 34, no. 5, pp. 376-385, 2007.

[100] M. Cutkosky and I. Kao, "Computing and controlling compliance of a robotic hand", IEEE Transactions on Robotics and Automation, vol. 5, no. 2, pp. 151-165, 1989.

[101] N. Tsagarakis, M. Laffranchi, B. Vanderborght and D. Caldwell, "A compact soft actuator unit for small scale human friendly robots", IEEE International Conference on Robotics and Automation, 2009, pp. 4356–4362.

[102] A. Bicchi and G. Tonietti, "Fast and "Soft-Arm" Tactics", IEEE Robotics & Automation Magazine, vol. 11, no. 2, pp. 22-33, 2004.

[103] M. Zinn, O. Khatib, B. Roth and J. Salisbury, "Playing it safe", IEEE Robotics & Automation Magazine, vol. 11, no. 2, pp. 12-21, 2004.

[104] M. Laffranchi, N. Tsagarakis, and D. Caldwell, "CompAct Arm: a compliant manipulator with intrinsic variable physical damping." Robotics: Science and Systems, vol. 8, pp. 225-232, 2013.

[105] M. Zinn, O. Khatib and B. Roth, "A new actuation approach for human friendly robot design", IEEE International Conference on Robotics and Automation, 2004, pp. 249–254.

[106] J. Hurst, J. Chestnutt and A. Rizzi, "An actuator with physically variable stiffness for highly dynamic legged locomotion", IEEE International Conference on Robotics and Automation, 2004, pp. 4662–4667.

[107] R. Schiavi, G. Grioli, S. Sen and A. Bicchi, "VSA-II: a novel prototype of variable stiffness actuator for safe and performing robots interacting with humans", IEEE International Conference on Robotics and Automation, 2008, 2171 –2176.

[108] S. Dwivedy and P. Eberhard, "Dynamic analysis of flexible manipulators, a literature review", Mechanism and Machine Theory, vol. 41, no. 7, pp. 749-777, 2006.

[109] M. Tokhi and A. Azad, "Flexible Robot Manipulators", Stevenage: Institution of Engineering & Technology, 2017.

[110] A. Bichi and G. Tonietti, "Design, realization and control of soft robot arms for intrinsically safe interaction with humans", Proceedings of the IARP/RAS workshop on technical challenges for dependable robots in human environments, 2002, pp 79–87.

[111] G. Runge, T. Preller, S. Zellmer, S. Blankemeyer, M. Kreuz, G. Garnweitner and A. Raatz, "SpineMan: Design of a soft robotic spine-like manipulator for safe human-robot interaction", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015, pp. 1103-1110.

[112] G. Pratt and M. Williamson, "Series elastic actuators", IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots, 1995, pp. 399–406.

[113] E. Torres-Jara, "Obrero: a platform for sensitive manipulation", 5th IEEE-RAS International Conference on Humanoid Robots, 2005, pp. 327–332.

[114] Y. Xu and R. Paul, "A robot compliant wrist system for automated assembly", IEEE International Conference on Robotics and Automation, 1990, pp. 1750-1755.

[115] F. Montagnani, M. Controzzi and C. Cipriani, "Preliminary design and development of a two degrees of freedom passive compliant prosthetic wrist with switchable stiffness", IEEE International Conference on Robotics and Biomimetics (ROBIO), 2013, pp. 310–315.

[116] S. Arimoto, Y. Liu and T. Naniwa, "Model-based adaptive hybrid control for geometrically constrained robots", Proceedings IEEE International Conference on Robotics and Automation, 1993, pp. 618-623

[117] R. Araújo, U. Nunes and A. de Almeida, "3D Surface-Tracking with a robot manipulator", Journal of Intelligent and Robotic Systems, vol. 15, no. 4, pp. 401-417, 1996.

[118] Sangbong Park and Cheol Hoon Park, "Model-based path planning and tracking control using neural networks for a robot manipulator", Proceedings of International Conference on Neural Networks, 1997, pp. 1761-1765.

[119] Z. Jiang and T. Ishita, "A Neural Network Controller for Trajectory Control of Industrial Robot Manipulators", Journal of Computers, vol. 3, no. 8, 2008.

[120] B. Paijmans, W. Symens, H. V. Brussel, J. Swevers, "A gain-scheduling-control technique for mechatronic systems with position-dependent dynamics", In Proceedings of the American Control Conference, 2006, pp. 2933-2938.

[121] Z. Vukic. A tutorial on adaptive control: The self–tuning approach. Lecture notes, 2000.

[122] C. Pérez, C. Vivas and F.R. Rubio, "PID - Gain Scheduling Controller for a Robot Manipulator", IFAC Proceedings Volumes, vol. 33, no. 4, pp. 505-509, ISSN 1474-6670, 2000.

[123] K. Helal, M. Atia, M. El-Sebah, "Gain scheduling control with multi-loop PID for 2- DOF arm robot trajectory control", International Conference on Innovation Engineering Technologies ICIET, 2016, pp. 19-25.

[124] A. Chahkoutahi, M. Rahimi, S. Ashja, M. Gholami and M. Moradipour, "Novel Adaptive Fuzzy Inference Controller for Highly Nonlinear System", International Journal of Hybrid Information Technology, vol. 7, no. 6, pp. 211-222, 2014.

[125] M. Alata, M. Jarrah, K. Demirli, and A. Bulgak, "Fuzzy gain scheduling for position control of a robot manipulator", Journal of Intelligent and Fuzzy Systems, vol. 8, no. 2, pp. 111-120, 2000.

[126] H. P. H. Anh and N. T. Nam, "A new approach of the online tuning gain scheduling nonlinear PID controller using neural network" in PID Control Implementation and Tuning, Rijeka, Croatia: InTech, 2011.

[127] V. Mallapragada, D. Erol and N. Sarkar, "A New Method of Force Control for Unknown Environments," IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006, pp. 4509-4514.

[128] Kamalasadan, S, "A new generation of adaptive control: An intelligent supervisory loop approach (Unpublished doctoral dissertation) ", University of Toledo, Toledo, OH, 2004.

[129] M. Nazemizadehl, M. Taheri and Sh Nazeri. "The application of fuzzy-logic method to control of robots: review study". International Journal of Mechanical Engineering and Robotics Research, vol. 3, no. 2, pp. 229-238, 2014.

[130] R. Kamnik, D. Matko and T. Bajd, "Application of model reference adaptive control to industrial robot impedance control", Journal of Intelligent and Robotic Systems, vol. 22, pp. 153 -163, 1998.

[131] S. Dubowsky and D.T. DesForges, "The Application of Model-Referenced Adaptive Control to Robotic Manipulators," Journal of Dynamic Systems, Measurement and Control, vol. 101, no. 3, pp. 193–200, 1979.

[132] J. J. Craig, P. Hsu, and S. S. Sastry, "Adaptive Control of Mechanical Manipulators," IEEE International Conference on Robotics and Automation, 1986, pp. 190-195.

[133] S. Nicosia and P. Tomei, "Model Reference Adaptive Control Algorithms for Industrial Robots," Automatica, vol. 20, no. 5, pp.635–644, 1984.

[134] S. Ulrich and J. Z. Sasiadek, "Modified Simple Adaptive Control for a two-link space robot," Proceedings of the American Control Conference, 2010, pp. 3654-3659.

[135] E. Wilson and S. Rock, "Reconfigurable control of a free-flying space robot using neural networks", In Proceedings of the American control conference, vol. 2, 1995, pp. 1355–1359.

[136] J. R. Layne and K. M. Passino, "Fuzzy model reference learning control," The First IEEE Conference on Control Applications, Dayton, OH, vol. 2, pp. 686-691, 1992.

[137] E. Pereira, S. Aphale, V. Feliu and S. Moheimani, "A hybrid control strategy for vibration damping and precise tip-positioning of a single-link flexible manipulator", IEEE International Conference on Mechatronics, 2009, pp. 1-6.

[138] M. Bassil, Al-Hadithi, A. Jimenez and F. Mati, "Fuzzy Optimal Control for Robot Manipulators", Robot Manipulators New Achievements, Aleksandar Lazinica and Hiroyuki Kawai (Ed.), InTech, 2010.

[139] K. Åström, U. Borisson, L. Ljung and B. Wittenmark, "Theory and applications of self-tuning regulators", Automatica, vol. 13, no. 5, pp. 457-476, 1977.

[140] A. Tzes, "Intelligent self-tuning controllers for robot manipulators", IEEE International Conference on Systems Engineering, 1990, pp. 5-8.

[141] R. H. Mohammed, F. Bendary and K. Elserafi, "Trajectory Tracking Control for Robot Manipulator using Fractional Order-Fuzzy-PID Controller", International Journal of Computer Applications, vol. 134, no. 15, pp. 22-29, 2016.

[142] M. Llama, R. Kelly and V. Santibanez, "Stable computed-torque control of robot manipulators via fuzzy self-tuning", IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics), vol. 30, no. 1, pp. 143-150, 2000.

[143] A. Marwan, N. Farrukh, K. Sahari and S. Hanim, "Real-time on line tuning of fuzzy controller for two-link rigid–flexible robot manipulators", Transactions of the Institute of Measurement and Control, vol. 35, no. 6, pp. 730-741, 2012.

[144] J. Yuh, "A Self-Tuning Type Neural Net Controller For Robotic Manipulators", Intelligent Automation & Soft Computing, vol. 1, no. 2, pp. 221-230, 1995.

[145] S. Z. S, Al-Khayyt, "Tuning PID Controller by Neural Network for Robot Manipulator Trajectory Tracking," Al-Khwarizmi Engineering Journal, vol. 8, no. 1, pp. 19-28, 2013.

[146] P. Joel Perez, J. P. Perez, R. Soto, A. Flores, F. Rodriguez, and J. L. Meza, "Trajectory tracking error using pid control law for two-link robot manipulator via adaptive neural networks," Procedia Technology Journal, pp. 139-146, 2012.

[147] H. Chaudhary, V. Panwar, S. N and R. Prasad, "ANFIS PD+I Based Hybrid Force/ Position Control of an Industrial Robot Manipulator", International Journal of Materials, Mechanics and Manufacturing, vol. 2, no. 2, pp. 107-112, 2014.

[148] R. S. Dahiya and M. Valle, "Tactile Sensing for Robotic Applications", Sensors: Focus on Tactile Force and Stress Sensors, Rocha, J.G. & Lanceros-Mendez, S. (Ed.), ISBN: 978-953-7619-31-2, InTech, 2008.

[149] R. Araujo, U. Nunes and A.T. De Almeida, "Robot 3D force-based surface-tracking. 20th International Conference on Industrial Electronics", Control and Instrumentation, vol. 2, pp. 788-793, 1994.

[150] B. Yao and M. Tomizuka, "Adaptive Robust Motion and Force Tracking Control of Robot Manipulators in Contact With Compliant Surfaces With Unknown Stiffness", Journal of Dynamic Systems, Measurement, and Control, vol. 120, no. 2, pp. 232, 1998.

[151] P. Pagilla and B. Yu, "Robotic Surface Finishing Processes: Modeling, Control, and Experiments", Journal of Dynamic Systems, Measurement, and Control, vol. 123, no. 1, pp. 93, 2001.

[152] Y. Yin, H. Hu and Y. Xia, "Active tracking of unknown surface using force sensing and control technique for robot", Sensors and Actuators A: Physical, vol. 112, no. 2-3, pp. 313-319, 2004.

[153] E. Li and Z. Li, "Surface Tracking with Robot Force Control in Unknown Environment", Advanced Materials Research, vol. 328-330, pp. 2140-2143, 2011.

[154] X. Papageorgiou, S. Loizou and K. Kyriakopoulos, "Motion tasks for robot manipulators on embedded 2-D manifolds", Joint IEEE Conference on Control Applications (CCA), IEEE Computer Aided Control Systems Design Symposium (CACSD) & IEEE International Symposium on Intelligent Control, 2006, pp. 3047–3052.

[155] X. Papageorgiou, S. Loizou and K. Kyriakopoulos, "Motion Tasks and Force Control for Robot Manipulators on Embedded 2-D Manifolds", Proceedings IEEE International Conference on Robotics and Automation, 2007, pp. 4202–4207.

[156] X. Papageorgiou, H. Tanner and K. Kyriakopoulos, "Motion tasks for robot manipulators on embedded 2-D manifolds under input constraints", European Control Conference, 2007, pp. 3783–3789.

[157] J. Wang and Y. Li, "Surface-tracking of a 5-DOF manipulator equipped with tactile sensors", 11thInternational Conference on Control Automation Robotics & Vision, 2010, pp. 2448- 2453.

[158] J. Wang and Y. Li, "Traching Control Of A Redundant Manipulator With The Assistance Of Tactile Sensing", Intelligent Automation & Soft Computing, vol. 17, no. 7, pp. 833-845, 2011.

[159] D. Nakhaeinia, R. Fareh, P. Payeur and R. Laganière, "Trajectory planning for surface following with a manipulator under RGB-D visual guidance", 2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), 2013.

[160] A. Pichler and M. Jagersand, "Uncalibrated hybrid force-vision manipulation", IEEE/RSJ International Conference on Intelligent Robots and Systems, 2000, pp.1866, 1871.

[161] J. Baeten, H. Bruyninckx and J. De Schutter, "Integrated Vision/Force Robotic Servoing in the Task Frame Formalism", The International Journal of Robotics Research, vol. 22, no. 10-11, pp. 941-954, 2003.

[162] D. Xiao, B. Ghosh, N. Xi and T. Tarn, "Sensor-based hybrid position/force control of a robot manipulator in an uncalibrated environment", IEEE Transactions on Control Systems Technology, vol. 8, no. 4, pp. 635-645, 2000.

[163] T. Olsson, R. Johansson and A. Robertsson, "Flexible force-vision control for surface following using multiple cameras", IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004, pp.789-803.

[164] N.A. Ramey, J.J. Corso, W.W. Lau, D. Burschka, G.D. Hager, "Real-time 3D Surface Tracking and Its Applications," Computer Vision and Pattern Recognition Workshop, Conference, pp.34-40, 2004.

[165] K. Hosoda, K. Igarashi and M. Asada, "Adaptive hybrid visual servoing/force control in unknown environment", Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 1996, pp.1097-1103.

[166] C. Cheah, S. Hou, Y. Zhao and J. Slotine, "Adaptive Vision and Force Tracking Control for Robots with Constraint Uncertainty", IEEE/ASME Transactions on Mechatronics, vol. 15, no. 3, pp. 389-399, 2010.

[167] V. Lippiello, B. Siciliano and L. Villani, "Robot Force/Position Control with Force and Visual Feedback", Proc. of European Control Conference, 2007, pp. 3790–3795.

[168] R. Macknojia, A. Chavez-Aragon, P. Payeur and R. Laganière, "Experimental characterization of two generations of Kinect's depth sensors", IEEE International Symposium on Robotic and Sensors Environments Proceedings, 2012, pp. 150-155.

[169] K. Mankoff and T. Russo, "The Kinect: a low-cost, high-resolution, short-range 3D camera", Earth Surface Processes and Landforms, vol. 38, no. 9, pp. 926-936, 2013.

[170] R. Macknojia, A. Chavez-Aragon, P. Payeur and R. Laganière, "Calibration of a network of Kinect sensors for robotic inspection over a large workspace", IEEE Workshop on Robot Vision (WORV), 2013, pp. 184-190.

[171] K. Khoshelham, "Accuracy Analysis Of Kinect Depth Data", Isprs - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 512, pp. 133-138, 2012.

[172] D. Nakhaeinia, P. Payeur and R. Laganière, "Adaptive Robotic Contour Following from Low Accuracy RGB-D Surface Profiling and Visual Servoing", Canadian Conference on Computer and Robot Vision, 2014, pp. 48-55.

[173] D. Nakhaeinia, P. Payeur, A. Chávez-Aragón, A. M. Cretu, R. Laganière, and R. Macknojia. "Surface following with an RGB-D vision-guided robotic system for automated and rapid vehicle inspection," The International Journal on Smart Sensing and Intelligent Systems, vol. 9, no. 2, pp. 419-447, 2016.

[174] S.G. Pierce, C.N. Macleod, G. Dobie and R. Summan, "Path planning & measurement registration for robotic structural asset monitoring", 7th European Workshop on Structural Health Monitoring, La Cité, Nantes, France, 2014.

[175] P. N. Atkar, D. C. Conner, A. Greenfield, H. Choset, and A. A. Rizzi, "Uniform coverage of simple surfaces embedded in IR for auto body painting," Proc. Workshop on Algorithmic Foundations of Robotics, Utrecht/Zeist, The Netherlands, pp. 383–398, 2004.

[176] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull, "Exact cellular decompositions in terms of critical points of morse functions for sensor-based coverage tasks, " Int. J. Robot. Res., vol. 4, pp. 331–344, 2002.

[177] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull, "Morse decompositions for coverage tasks", International Journal of Robotics Research, vol. 21, no. 4, pp. 331-344, 2002.

[178] M. Krimmel, S. Kluba, M. Bacher, K. Dietz and S. Rein-ert, "Digital Surface Photogrammetry for Anthropometric Analysis of the Cleft Infant Face", Cleft Palate Craniofa- cial Journal, vol. 43, no. 3, pp. 350-355, 2006.

[179] B. Shin, R. Venkatramani, P. Borker, A. Olch, J. Grimm and K. Wong, "Spatial Accuracy of a Low Cost High Resolution 3D Surface Imaging Device for Medical Applications" In International Journal of Medical Physics, Clinical Engineering and Radiation Oncology, vol. 2, no. 2, pp. 45-51, 2013.

[180] R. Fareh, P. Payeur, D. Nakhaeinia, R. Macknojia, A. Chávez-Aragón, A.-M. Cretu, P. Laferrière, R. Laganière and R. Toledo, An Integrated Vision-Guided Robotic System for Rapid Vehicle Inspection, 8th Annual IEEE International Systems Conference, 2014, pp. 446-451.

[181] P. Laferrière, "Instrumented Compliant Wrist System for Enhanced Robotic Interaction," M.A.Sc. thesis, University of Ottawa, 2016.

[182] C. Adrian and A. Corneliu, "The Simulation of the Adaptive Systems using the MIT rule", International Conference on Mathematical Methods and Computational Techniques in Electrical Engineering, 2008, pp.301-305.

[183] P. Swarnkar, S. Jain, R. K. Nema, "Application of Model Reference Adaptive Control Scheme To Second Order System Using MIT Rule", International Conference on Electrical Power and Energy Systems, 2010.

[184] D. P. Bertsekas, J. N. Tsitsiklis, "Gradient convergence in gradient methods with errors ", SIAM J. Optim., vol. 10, no. 3, pp. 627-642, 2000.

[185] G. Oke and Y. Istefanopulos, "Gradient-descent based trajectory planning for regulation of a two-link flexible robotic arm", IEEE/ASME International Conference on Advanced Intelligent Mechatronics, vol.2, pp. 948-952, 2001.

[186] N. ThanhQuyen and Y. Wang, "Self- Structured Organizing Single-Input CMAC Control for Robot Manipulator", International Journal of Advanced Robotic Systems, vol. 8, no. 4, pp. 110–119, 2011.

[187] D.P. Bertsekas and J.N. Tsitsiklis, "Gradient convergence in gradient methods with errors". SIAM J. Optim, vol. 10, pp. 627–642, 2000.

[188] S. Lang, The Chain Rule and the Gradient, Part five, pp 599-624, 1986. doi: 10.1007/978-1-4419-8532-3-18.

[189] P. MacVicar-Whelan, "Fuzzy sets for man-machine interaction", International Journal of Man-Machine Studies, vol. 8, no. 6, pp. 687-697, 1976.

[190] İ. Eksin, M. Güzelkaya and F. Gürleyen, "A new methodology for deriving the rule-base of a fuzzy logic controller with a new internal structure", Engineering Applications of Artificial Intelligence, vol. 14, no. 5, pp. 617-628, 2001.

[191] M. Marey and F. Chaumette, "A new large projection operator for the redundancy framework", IEEE International Conference on Robotics and Automation, 2010, pp. 3727-3732.

[192] P. Albertos, A. Sala, "Fuzzy Logic Controllers. Advantages and Drawbacks", XIII Congreso de la Asociation Chilena de Control Automatico, vol. III, pp. 833-844, 1998.

[193] P. Patre and S. Joshi, "On using exponential parameter estimators with an adaptive controller", IEEE International Conference on Control Applications (CCA), 2011, pp. 107-112.

[194] A. Moughlbay, E. Cervera and P. Martinet, "Error regulation strategies for Model Based visual servoing tasks: Application to autonomous object grasping with Nao robot", 12th International Conference on Control Automation Robotics & Vision (ICARCV), 2012, pp. 1311-1316.

[195] A. Tsoularis and J. Wallace, "Analysis of logistic growth models", Mathematical Biosciences, vol. 179, no. 1, pp. 21-55, 2002.

[196] M. Branicky, "Multiple Lyapunov functions and other analysis tools for switched and hybrid systems", IEEE Transactions on Automatic Control, vol. 43, no. 4, pp. 475-482, 1998.

[197] C. Pudney, "Surface Following for Manipulators with Proximity Sensors," 1991.

[198] Y. Karayiannidis and Z. Doulgeri, "Robot force/position tracking on a surface of unknown orientation [C], European Robotics Symposium", Heidelberg: Springer, vol. 44, pp. 253–262, 2008.

[199] C. C. Peng and C. L Chen, "A contour following strategy for 2-link manipulator using coordinate transformation approach", Proceedings of the 13th IASTED International Conference on Robotics and Applications, 2007.

[200] B. Yu and P. R. Pagilla, "Design and implementation of a robust switching control scheme for a class of constrained robot tasks", International Journal of Systems Science, vol. 37, pp. 303–321, 2006.

[201] D. Xiao, "Sensor-based hybrid position/force control of a robot manipulator in an uncalibrated environment", IEEE Trans. Contr. Systems. Technol., vol. 8, pp. 635-645, 2000.

[202] X. Papageorgiou and K J. Kyriakopoulos "Controlling the Motion of Robot Manipulators on Constrained Surfaces", Proceedings of the 17th Mediterranean Conference on Control and Automation, 2009, pp. 1331-1336.

[203] B. Allotta, G. Buttazzo, "Impact handling by proximity and force sensing", Proc. IEEE Int. Conf. Robot. Auto., pp. 2032-2037, 1992.

[204] D. Nakhaeinia, P. Laferrière, P. Payeur, and R. Laganière, "Safe Close-Proximity and Physical Human-Robot Interaction Using Industrial Robots", 12th Conference on Computer and Robot Vision, pp. 237–244, 2015.

[205] D. Nakhaeinia, P. Payeur, and R. Laganière, "A Mode-Switching Motion Control System for Reactive Interaction and Surface Following Using Industrial Robots", IEEE/CAA Journal of Automatica Sinica (accepted), 2017.

[206] D. Nakhaeinia, P. Laferrière, P. Payeur and R. Laganière, "Hybrid Adaptive Control of Industrial Robots for Surface Exploration of Arbitrary Objects", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2017- Poster presentation).

[207] D. Nakhaeinia, P. Laferrière, P. Payeur and R. Laganière, "Hybrid-Adaptive Control of Industrial Robots Interacting With Objects under Multimodal Guidance", IEEE/ASME Transactions on Mechatronics (submitted).

[208] M. Tölgyessy and P. Hubinský, "The Kinect Sensor in Robotics Education" In Proceedings of 2nd International Conference on Robotics in Education", 2011.

[209] Z. Zhang, "A Flexible New Technique for Camera Calibration," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1330-1334, 2000.

[210] A. Chávez-Aragón, R. Macknojia, P. Payeur, R. Laganière, "Rapid 3D Modeling and Parts Recognition on Automotive Vehicles Using a Network of RGB-D Sensors for Robot Guidance", Journal of Sensors, Hindawi (ed.), pp.1- 16, 2013.

[211] W.M. Keck Virtual Factory Lab, "Virtual Robot System version 2, Technical report", School of Industrial and Systems Engineering, Georgia Institute of Technology, pp. 25-39, 2002.

Appendix A – Microsoft Kinect Technology

The Kinect sensor was built by Microsoft to revolutionize the gaming industry. It enables a computer to sense the third dimension (depth) of the environment and let people interact with computer games. Most of the sensors used in manipulator robots navigation are not capable to capture a color and a depth image simultaneously, and those that acquire depth require a considerable amount of time to perform range acquisition over an entire surface. In contrast, the ability in measuring and monitoring surfaces in 3D and the extreme acquisition speed of Kinect technology motivated the adoption of the Kinect sensor for rapidly acquiring objects of various shape and color and to develop a wide range of applications.

The Kinect sensor, version Xbox 360 used in this research, contains an infrared (IR) projector, one color camera, and one IR camera (Figure A.1), as well as a microphone array. Its field of view is 57° horizontally and 43° vertically, and the operational range of the sensor is between 0.8 to 3.5 m, while it can be extended up to 6 m.

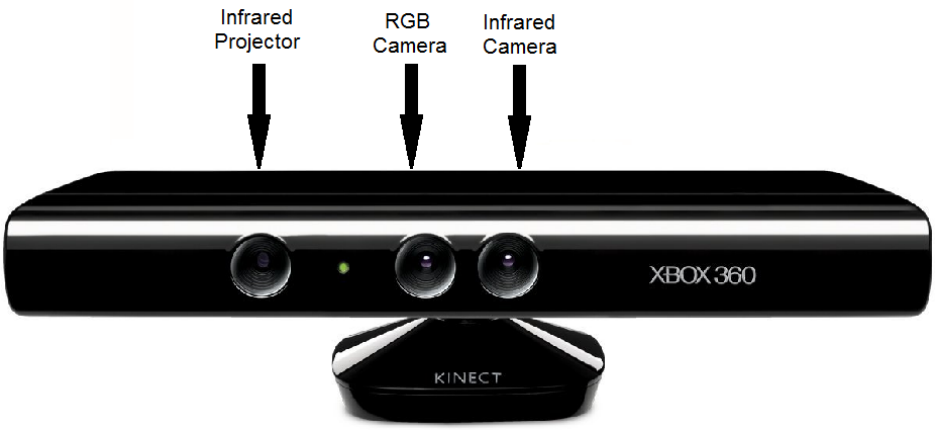


Figure A.1: Microsoft Kinect sensor.

The infrared projector generates electromagnetic waves and projects them on objects in form of a set of infrared dots (light pattern). The lights are reflected from the objects and captured by the infrared camera to create a depth map through correlation and triangulation. The information from the depth sensor and the RGB camera are combined to capture the surrounding world in 3D with

surface textures. The result of the combination is a RGB-D image at 640x480 pixels resolution where each pixel contains color and depth information.

However, the depth values distribution is not the same over the entire field of view and the depth accuracy decreases with the distance due to calibration between the infrared projector and the infrared camera [208]. The accuracy can be improved by using specific calibration procedures and software correction of sensor outputs but it still might not be suitable for some robotic tasks.

The color and depth images provided by the Kinect sensor are not registered. To merge the color and depth, first the Kinect's internal intrinsic and extrinsic parameters should be determined. The intrinsic parameters for the color sensor are the focal length and the principal point, and the infrared sensor intrinsic parameters are lens distortion coefficients. The intrinsic parameters can be determined using the standard chessboard recognition method proposed by Zhang [209]. Extrinsic parameters for built-in Kinect sensors (color and depth) within each Kinect unit are determined by stereo calibration. The position between both cameras is defined by Eq. A.1:

$$H = H_{RGB} H_{IR}^{-1} \tag{A.1}$$

H_{IR} is the homogenous transformation matrix from the depth (IR) camera to a checkerboard target used for calibration, and H_{RGB} is the homogenous transformation from the color (RGB) camera to the checkerboard target. The internal extrinsic calibration parameters allow to accurately relate the color to depth data collected by a given Kinect device. However, the Kinect sensor does not directly provide the registered color and depth images. Therefore, the 3D coordinates corresponding to each point in the depth image are computed as follows:

$$\begin{aligned} X_{IR} &= (x - O_{x\_IR}) du(x,y) / f_{x\_IR} \\ Y_{IR} &= (y - O_{y\_IR}) du(x,y) / f_{y\_IR} \\ Z_{IR} &= du(x,y) \end{aligned} \tag{A.2}$$

where (X_{IR}, Y_{IR}, Z_{IR}) are the 3D point coordinates of pixel (x, y) in the depth image with respect to the IR camera reference frame, (x, y) is the pixel location in the depth map, (f_{x_IR}, f_{y_IR}) is the focal

length of the IR camera, (O_{x_IR}, O_{y_IR}) is the optical center of the IR camera, and $du(x, y)$ is the depth of a pixel (x, y) in the depth image. Next, the color is assigned from the RGB image to each 3D point P_{IR} (X_{IR}, Y_{IR}, Z_{IR}) as follows:

$$P_{RGB}\,(X_{RGB}, Y_{RGB}, Z_{RGB}) = R \cdot P_{IR} + T$$
$$x = (X_{RGB}.\, f_{x\_RGB}/Z_{RGB}) + O_{x\_RGB} \quad \text{(A.3)}$$
$$y = (Y_{RGB}.\, f_{y\_RGB}/Z_{RGB}) + O_{y\_RGB}$$

where P_{RGB} is the 3D point with respect to the color camera reference frame, R and T are the rotation and translation parameters from the color camera to the depth camera obtained from the calibration procedure, (f_{x_RGB}, f_{y_RGB}) is the focal length of the color camera, (O_{x_RGB}, O_{y_RGB}) is the optical center of the color camera, and (x, y) is the pixel location of color information in the color image.

Kinect sensors can also be used for building dense 3D maps and modeling of large scale indoor environments. For this purpose, a multi-view vision system is required to rapidly acquire and reconstruct a full scale model of the environment. A calibration methodology was proposed in [168, 170] to estimate the internal and external calibration parameters over a network of Kinect sensors to achieve accurate alignment between the respective point clouds and textured images acquired by each Kinect sensor that are distributed in a collaborative network of imagers to provide coverage over the large objects [210].

Appendix B – Instrumented Compliant Wrist

Taking inspiration from humans interacting with their environment, compliance is a key aspect for manipulation tasks that increase the flexibility of a robotic platform and reduce the risk of damage or injury. Various approaches and compliant devices were found in literature (section 2.2) to provide the required information for robot manipulators for safe and flexible contact interaction with objects. To prevent the need for sophisticated algorithms, expensive force and vision sensors, or costly modifications, an instrumented compliant robotic wrist device was designed and built by my colleagues [181] in the SMART research group. The device provides an adaptable interface between the environment and a robot. Moreover, it collects the required local information to support dexterous robotic interaction prior to initial contact, and also while contacting with an object using live proximity and contact feedback.

The compliant wrist is composed of two rigid plates separated by springs which provide enough freedom for the upper compliant plate to adapt to the position and orientation of a surface that the robotic wrist might touch (Figure B.1). It provides a means of detecting objects both in proximity and in contact to the end-effector, as well as adding a degree of compliance to the end-effector which enables the latter to touch or slide on the object without damaging it. With the extra sensing capabilities provided, a robot controller can dynamically adapt to surface changes or displacement that may occur when contact with the robot happens.

Figure B.1: Compliant wrist.

As shown in Fig. B.2, the compliant wrist is equipped with eight infrared distance measurement sensors. The infrared sensors are arranged in two arrays of four sensors each: an external array and an internal array. The external sensor array allows to measure distances at multiple points between the wrist and the object located in front of the end-effector, supporting control in proximity to objects. The internal sensor array is situated between the base of the compliant wrist structure and a moveable plate that allows the device to estimate the surface orientation and distance to an object when the robot is contacting with it. The combination of the two sensory layers provides the necessary information for fine tuning the robot pose while maneuvering in close proximity or contact with an object. To make use of the measurements provided by the compliant wrist, it is required to create a closed feedback loop between the compliant wrist module and a robot. This thesis focuses on the development of the corresponding control schemes to fully take advantage of the extra sensing data available through the wrist.

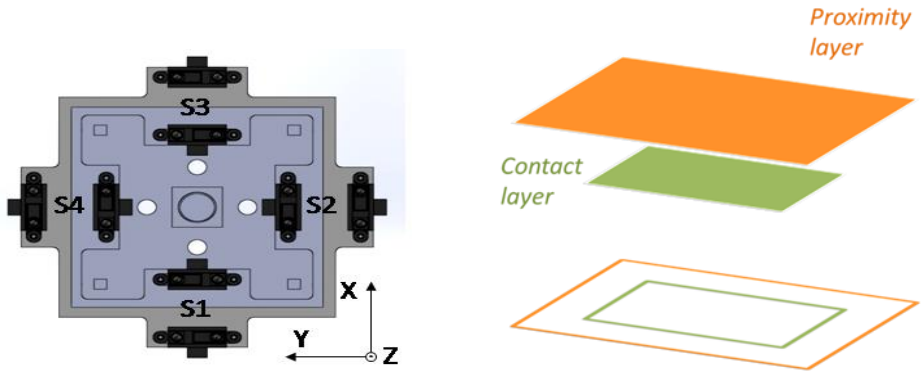


Figure B.2: a) Compliant wrist sensors arrangement and b) dual sensing layers of the compliant instrumented device (reproduced with permission from [181]).

The sensing layers estimate an object's pose in the form of a 3D homogeneous transformation matrix. The rotation and translation parameters are obtained using the distance measurements from the IR sensors. Eq. B.1 shows how the transformation matrix is calculated using distances provided by the internal sensors (contact sensing layer). A similar calculation can be made from the external sensors (proximity sensing layer) using values E, F, G and H instead, as measured by external sensors (Fig. B.3). The transformation matrix determines the object pose with respect to the compliant wrist frame which is transferred to the robot's base frame.

$$Q_{endeff/object} = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha & 0 \\ \sin\beta\sin\alpha & \cos\beta & -\sin\beta\cos\alpha & 0 \\ -\cos\beta\sin\alpha & \sin\beta & \cos\beta\cos\alpha & \frac{A+B+C+D}{4} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(B.1)}$$

where $\beta = atan2(^{D-B}/_{W})$, $\alpha = atan2(^{A-C}/_{W})$

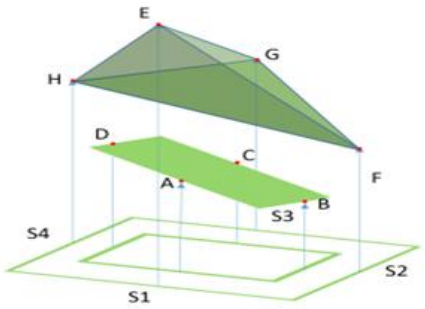


Figure B.3: Distance measurements from internal and external sensors (reproduced with permission from [181]).

Fig. B.4 illustrates the key components of a closed feedback loop involving the compliant wrist. The blue arcs in the vicinity of the compliant wrist module and the transceiver connected to the robot show how all information generated by the compliant wrist is delivered wirelessly to the robot controller to direct the motion of the robot. The communication channel is bidirectional which enables the robot controller to send commands to the compliant wrist module to make data requests.

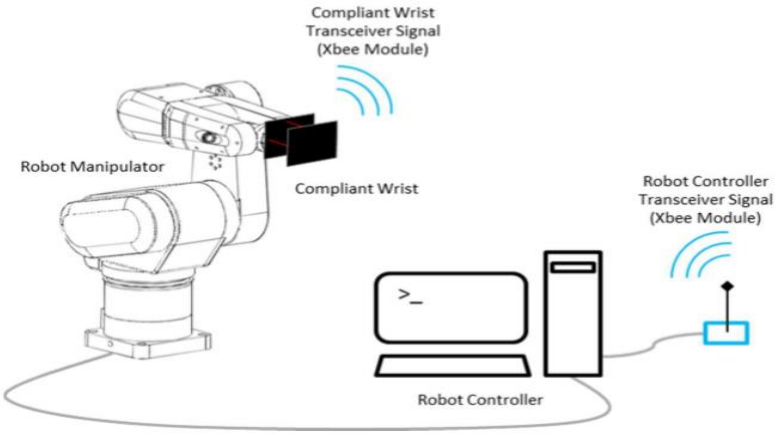


Figure B.4: Wireless communication between the compliant wrist and robot (reproduced with permission from [181]).

Appendix C – CRS F3 Manipulator

The CRS F3 manipulator used for this research is a 6-DOF robot arm mounted on a 2-m long linear track (Fig. C.1), which adds a 7th DOF.

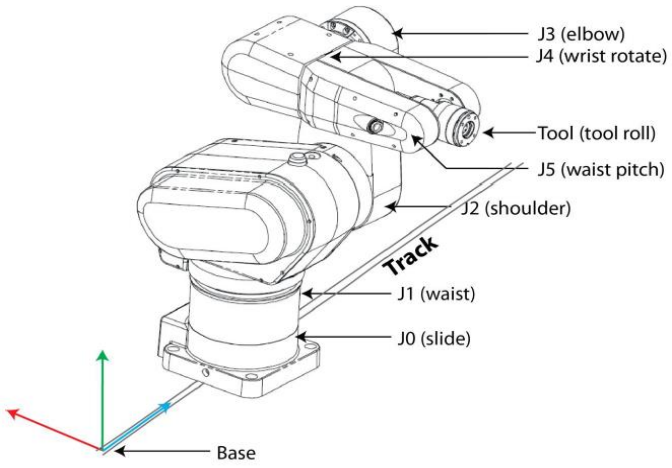


Figure C.1: CRS F3 manipulator.

The Denavit-Hartenberg parameters of the CRS F3 robot are given in Table C.1.

Table C.1: Denavit-Hartenberg parameters.

i	a_i	d_i	α_i	θ_i
0	0	D_0	90	180
1	a_1	d_1	90	$\theta_1 + 90$
2	a_2	0	0	$\theta_2 + 90$
3	0	0	90	$\theta_3 + 90$
4	0	d_4	90	$\theta_4 + 180$
5	0	0	90	$\theta_5 + 180$
6	0	$d_6 + L$	0	θ_6

where $a_1 = 100$; $d_1 = 350$; $a_1 = 270$; $d_4 = 265$; $d_6 = 75$ and $L = 70$.

The forward kinematic model and the inverse kinematic model of the robot can be deduced using the DH table in order to transform the desired trajectories planned in the Cartesian space to the joint space. If the transformation matrix of the end-effector related to the base is given by the following expression:

$$M_{06} = \begin{bmatrix} r_{11} & r_{21} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{C.1}$$

The position of the center of the end-effector is given as follows:

$$\begin{aligned} o_x &= T_x - (d_6 + L) r_{13} \\ o_y &= T_y - (d_6 + L) r_{23} \\ o_z &= T_z - (d_6 + L) r_{33} \end{aligned} \tag{C.2}$$

There are different techniques to calculate the closed loop inverse model. Here, the inverse model was inspired from [211] and modified for the 7 DOFs, CRS-F3 manipulator. As a general solution, the first three joint's angles can be deduced as follows:

$$\theta_1 = atan2(o_y, o_x) \; or \; \theta_1 = atan2(o_y, o_x) + \pi \tag{C.3}$$

$$\theta_3 = 2atan\left(k_1 \pm \sqrt{(k_1^2 - k_3^2)/k_3}\right) \tag{C.4}$$

where

$$k_1 = 2a_2 d_4$$

$$k_3 = o_x^2 + o_y^2 + o_z^2 - 2o_x a_1 c_1 - 2o_y a_1 s_1 + a_1^2 - a_2^2 - d_4^2;$$

$$c_1 = \cos(\theta_1) \; and \; s_1 = \sin(\theta_1).$$

$$\theta_2 = atan2(s_2, c_2)$$

and

$$s_2 = \sin(\theta_2) = (c_2 - c_1 \mu_2/\mu_1)/(v_2 - v_1 \mu_2/\mu_1);$$

$$c_2 = \cos(\theta_2) = (c_1 - v_1 s_2)/\mu_1$$

$$\mu_1 = a_2 + d_4 s_3,\ \mu_2 = -d_4 c_3$$

$$v_1 = d_4 c_3,\ v_2 = a_2 + d_4 s_3$$

$$c_1 = o_x c_1 + o_y s_1 - a_1,\ c_2 = o_z$$

θ_4, θ_5 and θ_6, are calculated using the transformation matrix of the last three joints M_{36}, while θ_1, θ_2 and θ_3 calculated previously allow to determine the matrix M_{03}. As M_{06} is already known from the position and orientation of the Cartesian desired point, the matrix M_{36} can be defined as follows:

$$M_{36} = M_{03}^{-1} M_{06} = \begin{bmatrix} r_{11} & r_{21} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(C.5)}$$

Using the elements of matrix M_{36}, the angles θ_4, θ_5 and θ_6 are calculated as follows:

$$\theta_4 = atan2(r_{23}, r_{13}) \quad \text{(C.6)}$$

$$\theta_5 = arccos(-r_{33}) \quad \text{(C.7)}$$

$$\theta_6 = atan2(-r_{32}, r_{31}) \quad \text{(C.8)}$$