TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

SIAVASH DANAEE BARFOROOSHI

Developing a control Architecture for a vision based automatic pallet picking

Master's Thesis

Examiner: prof. Kalevi Huhtala
Prof.Reza Ghabcheloo
Examiner and topic approved by the
Faculty Council of the Faculty of
Engineering sciences
on 4th November 2015

ABSTRACT

Siavash Danaee.B: Developing a control Architecture for a vision based automatic pallet picking
Tampere University of technology
Master of Science Thesis, 44 pages, 0 Appendix pages
January 2016
Master's Degree Program in Machine Automation
Major: Mechatronics
Examiners: Prof Reza Ghabcheloo, Prof Kalevi Huhtala

Keywords: visual servoing, pallet picking, control architecture, stateflow

The current project covers the problem of controlling an autonomous machine in the application of pallet picking by vision camera sensor. This work is under the Generic Intelligent Machine program at Tampere University of Technology.

The goal of this project is to enhance the controlling performance of an articulated-frame-steering autonomous machine. The current system employs the vision sensor for the detection of the target object and controlling the machine movement by the data obtained by that. The problem we face is that at a far distance when location of the object is detected for the first time, the unreliable data specially the orientation of target will be acquired. So relying on this type of data for heading the machine can distract the whole system behavior, and also there should be a logic for switching between different states of machine in order to control the different stages of performance.

In order to solve the problem, a smooth switching logic will be defined to control the machine. This switching logic should be in a case that operational coordinates with the visual servoing be synchronized and it should be planned to control robot degrees of freedom in each step. The MATLAB/Simulink is employed to execute the idea since this software is capable in the logic-based planning in a graphical way in *stateflow*

The results of the implemented method would be increasing the accuracy of picking the object and enabling the machine to have more space to have any needed maneuver, moreover the environmental disturbances will have the minimum effect on the final result. The accuracy of the system is shown in figures.

PREFACE

This thesis was performed at Intelligent Hydraulic & Automation department of Tampere university of Technology. I would like to thank Dr. Reza Ghabcheloo , my supervisor, for his great help, advises and his patience , and Prof. Kalevi Huhtala for his support. Furthermore, I appreciate Mr. Mohammad M.Aref for his help in project and sharing honestly his knowledge with me.

Finally, I would express my love to my parents and my brothers for their fantastic support during my life and I will always owe them.

Tampere, 20.01.2016

Siavash Danaee.B

CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

FSM	Finite State Machine
GPF	Global Path Following
GVS	Global Visual Servoing
LVS	Local Visual Servoing
GPS	Global Positioning System
NED	North-East-West coordinate system
PBVS	Position Based Visual Servoing
IBVS	Image Based Visual Servoing
AFS	Articulated Frame Steering
IMU	Inertial Measurement Unit

.

1. INTRODUCTION

1.1 Overview:

The robotic science is an activity in smart combination of different branches of science that considers designing, construction and implementation of automatic controllable devices. Service robots which act in actual environments populated by humans are so popular during recent years specially in hospitals, stores, factories and so on. Beside the different types of robots, autonomous mobile robots are one the most interesting with high complexity, as they need high intelligence level to handle different skill and enhance in different environmental situations. These robots are not only applied for their mobility but also, because of their autonomy, and so their capacity in maintaining a sense of position and to navigate without human interference is superior [1]. In the last decade, both industry and academia have put numerous effort in developing the autonomous machines technologies to improve their accuracy and efficiency.

Automatic pallet picking is one of the functionalities and abilities of an autonomous machines. It can be used in industrial or urban locations to reduce the human resources and reduction in financial and energy waste. This mission cannot be completed without controlling the machine navigation toward the object and detecting and observing the target perfectly. This is considered as our main goal for picking the object, so correct information from the camera used for observation and controlling strategy can help us to overcome this problem.

Section 1.2 will represent the basic information about the GIM machine promoted at Tampere university of Technology used as our mobile robot for this task. Section 1.3 represents the camera used for the vision and detection of the environment based on marker detection methods and the different camera installation on robots.

1.2 GIM machine

The generic Intelligent Machine (GIM) is a project by the collaboration of Tampere University of Technology and Aalto University and the project goal is to develop the methods and technologies in robotic intelligence field and enhance the automatic mobile machines functionalities[3]. Several machines have been developed under this project. The current project is defined on the so called “GIM machine”, shown on Figure 1. This machine is now located at the Intelligent Hydraulic & Automation (IHA) department laboratory of TUT. The autonomous GIM is based on AVANT 635 Articulated-Frame-Steering (AFS) machine. An AFS machine consists of two separated parts connected to each other by a hinge and each part has freedom in rotation angles and steering is executed by the linear extension or contraction of a hydraulic cylinder attached on both units making the whole body steering left or right[4]. According to the physical characteristics of AVANT[1], they are so common in agricultural and industrial locations both in manual or automatic ways, and because of these high applications, defining autonomous project on this machine is logical and financially convincing, although controlling this type of heavy machines has more complexity since physical limitations known as kinematic constraints.

There have been defined different projects on this autonomous machine, and they been modified during years to be up-to-dated and increase its accuracy and efficiency, so the current project is also defined to satisfy this roadmap.

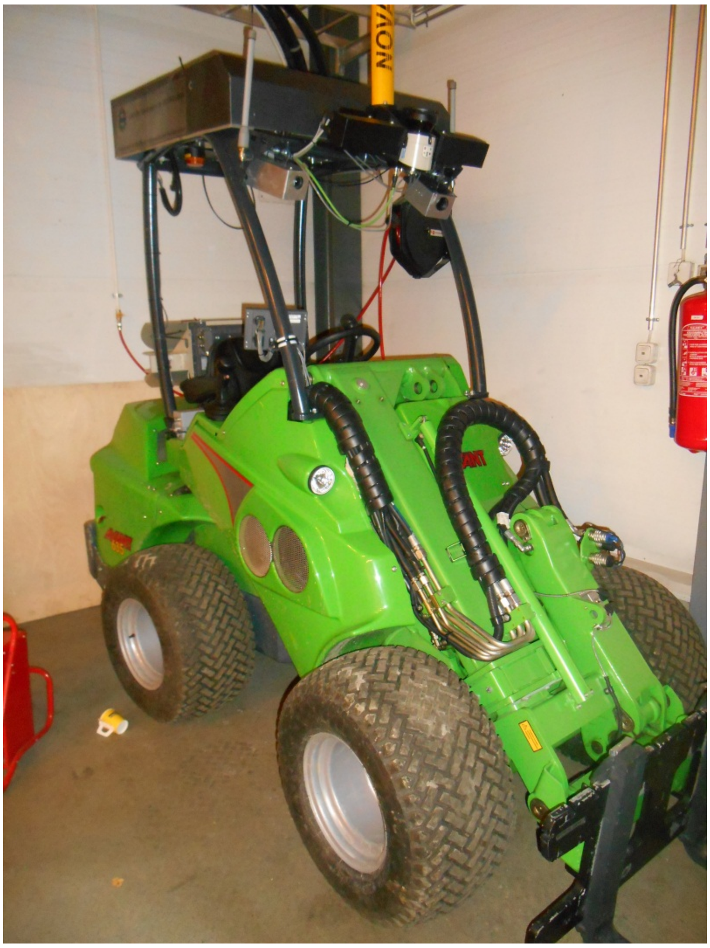


Figure 1- GIM Mobile Machine

[1] http://www.avanttecno.com/www/global/machines/600-series/avant-635/

1.3 Camera

Human being are equipped by their perfect eye in order to monitor the environment. The scientists have used this idea in order to capable the robots to recognize their around side just like humans. In order to reach that goal, the cameras have been used. Employing the camera itself does not solve the problem because camera does not understand what it sees, so the algorithms have been developed to make the camera intelligent in object detection of the environment. To do so, one of the easy and straightforward ways, is use of fiducial markers . A fiducial marker is an external object with a clear and constant model which is located in a scene, and approaches have been developed in order to obtain a reasonable data based on the natural features of the marker, and accordingly several interest point detection and matching techniques have been developed so far[5]. Different fiducial markers such as concentric circles, Intersene, ARTag, ARToolkit are used in scientific area that in each of them different interest points are considered, and in our experiment ARTag type is employed. Generally, in the current system, the pallets which is used as the target is a special type of pallet called Euro-pallet. This type of pallet has special size and it is standardized by the European Pallet Association (EPAL)[2]

The use of camera in a robot control system can be accomplished in two architecture types: camera is installed directly on the robot manipulator or on the mobile robot or it is called eye-in-hand case (Figure 2A); in this case the motion of the robot induces the camera motion, in other words the robot manipulator movement according to the camera is zero, and this type has a partial but precise sight of scene [6], In other case, the camera can be fixed in the workspace area or a space outside of the robot working area, so that camera can observe the robot motion from an stationary configuration; this case is also called eye-to-hand; and this case is less precise but it has global sight of view. In figure 2, these two types of camera installation are shown graphically in a simple robotic manipulator. On the left side, the camera is attached to robot, but on the right, camera is located in a special position where it has view to the robot movements.

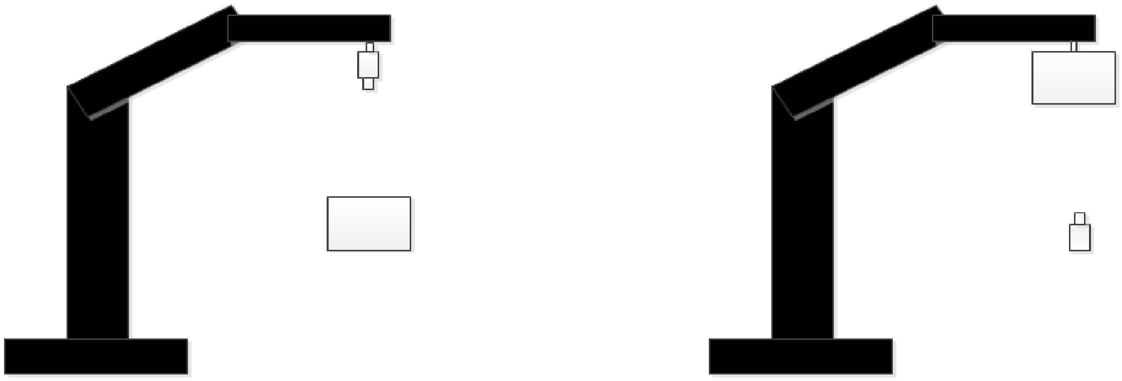


Figure 2- Camera installation techniques (3A- Left side, 3B- Right side)

[2] http://www.epal-pallets.org/uk/home/main.php

The camera installed on the GIM machine has the fixed model to the robot body. As it can be seen in figure 1, because the height of machine is high, the camera is installed in a tilted angle to be sure that the camera can recognize the pallet markers on the ground. In these cases, the plant camera should be mounted as high as possible on the field robot. The camera model in system is Gima GO423C and for fiducial markers detection, the ROS version algorithm has been used. The defined task is calculating the position and orientation of the pallet with respect to the camera, known as vision data.

2. THEORETICAL BACKGROUND :

2.1 Visual Servo

Vision is a powerful and practical robot capability as it simulates the human vision sense and enables noncontact measurement of the surrounding environment. The visual sensing system can be stand on typical camera and laser or the new technology radio frequency identification (RFID), and all these methods try to satisfy three procedures sensation, estimation, and matching, in order to have the best sense of vision from the objects in scene. The machine vision system is commonly employed in many industrial systems, such as robotic manufacturing systems, in order to compensate the robotic positioning error[8], or object tracking like pedestrians [9], and many other industrial applications.

Visual servoing is defined by the use of visual feedback in controlling a robot. So, the vision is a control system part that provides the feedback loop about the environment. The robot system closed loop control, where the vision is employed usually consists of two interconnected processes: tracking and control. Tracking allocates a continuous estimation during the robot motion and updates visual features, and the robot control system is based on the sensory input. Based on this input process, the controlling sequence is generated[10]. Commonly the general task of vision sensing is positioning and tracking. Positioning means that the object should be located at the desired position, and tracking refers to a constant distance between robot and target object, that in both states the error feedback signals should be considered.

According to the visual information, visual servoing system can be divided into these classes:

- Position-based visual servoing:

In this method the 3-Dimentional information about the target object is known, and this 3D data is used to estimate the position and orientation of the object based on the frame of the camera, in other words, this type of system the estimated pose and rotation of target would be sent to controller for following the path to target. In this method, the vision output can integrate with other sensor modalities. Moreover, the calibration of the camera extrinsic parameters with respect to the body has an important role in the accuracy of manipulation [10].

The marker detection algorithm used in our camera system is ALVAR[3] computer vision library. This open-source software has been developed at VTT[4] Technical research center of Finland.

- Image based visual servoing:

According to this method, the 2D image measurement features are used for position estimation. The control law is based on the error between current and desired features on the image plane, and this method does not consider any estimation for the position of the target [17]. An image feature is a real-valued parameter that can be extracted from one or more image features such as interest points and lines. this method does not need any pre knowledge about 3d structure of object.

2.2 Bezier curve

In the year 1950's, a designing smooth shape was a very boring task. An employee at the Citroen automobile company; paul de Faget de Casteljau; represented the idea of iterating affine combination of polygon meshes to obtain smooth surfaces suitable for modeling car chassis. Almost at the same time, Pierre Bezier at Peugeot worked on the intersection of partial cylinder to achieve the same goals. Both produce equal curves which are known today as the Bezier curve [19]. The Bezier curve was a very common way to represent the smooth curves in computer and mathematic graphics.

Typically, the Bezier curve is denoted by several consequent points called *control points*. The points start from $P0$ to Pn as the end point, for example for a linear curve $n = 1$, and for a quadric curve $\mathrm{n} = 2$. The Bezier curves are generally defined based on time; as we start from$\mathrm{t} = 0$, which is $P0$, and we would reach $P1$ at $\mathrm{t} = 1$. The middle points of the Bezier curve are used to specify the endpoint tangent vectors, since the curve passes through $P0$ and Pn (end-point), while approximating the other middle control points.

For example the linear and quadric interpolation will be :

$$B(t) = (1 - t)P_0 + (t)P_1 \;, \qquad 0 < t < 1$$

$$B(t) = (1 - t)^2P_0 + 2(1 - t)tP_1 + t^2P_2 \;, \qquad 0 < t < 1$$

[3] http://vitual.vtt.fi/virtual/proj2/multimedia/alvar/

[4] http://vttresearch.com/

As a general rule the Bezier curve formulation will be :

$$B(t) = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^i P_i \quad , 0 < t < 1$$

Where, i is the control point to be considered (weighted), and n is the degree of curve.

In order to have a smooth path, a 6-order degree of Bezier curve has been chosen, thus six control points; each point with two dimensions, x, y; should be defined. Moreover as the Bezier is defined according to time, we can control the velocity of the machine during each consecutive points, by defining 5 velocity values. Using a high order of a Bezier curve will enable us to have more freedom in choosing the best control points and also it provides higher degree of continuity between points of the path.

2.3 Coordinate Frames

In robotics, in order to define the position and orientation of objects, there has been defined coordinate frames concept. A coordinate system consists of a plane or space by axes from a fixed point called the origin, and the robot targets positions are allocated by measuring along the axes of coordinate systems. The coordinate frames can change to the other frames by a matrix which is transformation matrix. Coordinate frames and the transformation of data between different coordinate frames are widely used in many applications. In a robot, coordinate frame can be divided into base frame, tool frame, global frame (target frame) and for each part a special frame can be defined.

2.3.1 Global navigation coordinate frame

The robot Localization is a basic and crucial problem in making a robot fully autonomous. If a robot does not know where it is, the next activity will be impossible. Typically in localization, the current position and orientation according to the references would be defined. In order to localize the robot there have been defined different methods [14] and [15]. There are different methods in navigation technique solution like Global navigation satellite system (GNSS) which utilize the satellite information for positioning of objects [14]. Mainly this method can be used in open area locations and as the current GIM system is located in such an area, this method can be utilized.

In the GIM system, different coordinate frames are defined in order to have the position and orientation of each part of machine. Moreover the position or orientation make sense with respect to an original frame. To do so, an original frame is defined. Basically, the first original frame is shown by $\{\tau\}$; denotes the original global frame, this is a fixed coordinate frame defined by North-East-Down (NED) coordinate frame. This NED frame is based on the geographical directions of the ground, widely used is GPS based navigation systems. In this frame against the usual Cartesian frames, the X axis is along the geodetic north direction, and Y along to the east and down goes to the center of the earth. See Figure (3).

2.3.2 Local navigation coordinate frame

The local coordinate frame has been defined in the navigation system in order to have high frequency location. In the GIM system, the local coordinate frame has been defined on the target object, since it can be easier for calculation that in the local mode, the target would be to reach the new origin point; pallet. One of the disadvantageous of this coordinate frame is that, there might be unexpected drift or distract in the machine movement.

In the current system, both global and local information are considered. As it can be seen in figure below, there have been defined two coordinate reference, that the machine would switch to each reference frame in special switching time. It is clear that the origin global frame is fixed on the ground and the local coordinate frame is located on the center of the target pallet.

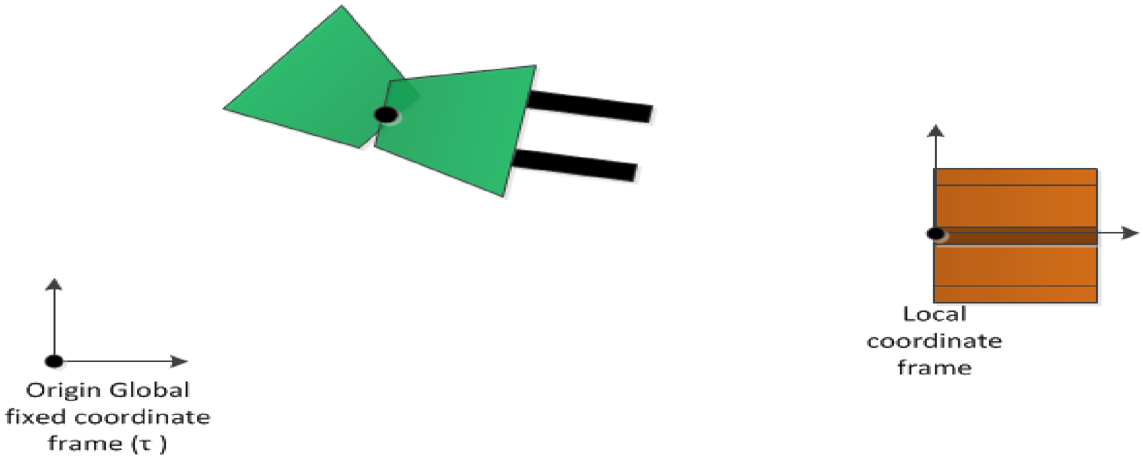


Figure 3- Global and Local coordinate frames

2.3.3 All system coordinate frames

Beside the original reference frame, all the main parts of GIM machine; such as body, camera, fork; and the targets used in our models should be defined by coordinate frames. Hence, The machine; combination of body, hydraulic actuators, camera; and pallet frames are independently defined. For example, the body frame $\{B\}$, is located at the middle of the front axle in a case that the x axis direction goes forward and z axis direction goes downward, and similarly camera frame is on the tilted camera, and pallet frame has the same frame. All the exact coordinate directions are shown in picture below (Figure 4).

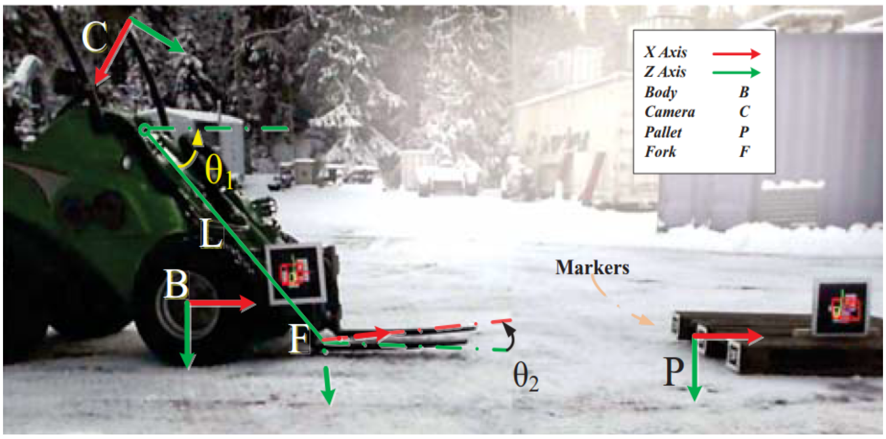


Figure 4- All the coordinate frames [11]

According to the defined coordinate systems, the transformation matrix of body frame with respect to the original frame can be :

$${}^{\tau}T_B\ (x, y, z, \Phi, \theta, \Psi) = \begin{bmatrix} {}^{\tau}R_B\ (\Phi, \theta, \Psi) & {}^{\tau}P_B \\ 0 & 1 \end{bmatrix}_{4\times4}$$

$R(\Phi, \theta, \Psi)$ denotes the rotation matrix of the body as a function of roll(Φ), pitch(θ), and yaw(Ψ). ${}^{\tau}P_B$ is the position of the body origin in the inertial frame by x, y, z parameters. The pallet and fork frames are shown respectively by $\{P\}$ and $\{F\}$. The desire of this project for picking up the pallet perfectly, will be obtained when the fork coordinate frame coincides with the pallet frame or mathematically: ${}^{\tau}P_P = {}^{\tau}P_F$ [2].

The manipulator of the machine, is used for aligning to pick up the pallet. Manipulator consists of a boom and a fork. It has three degrees of freedom. Boom can rotate around y-axis of body shown by θ_1 and elongate shown by L. Also fork can rotate around y-axis of body by angle of θ_2.

The manipulator parameters are also shown in Figure 4.

2.4 Sensor information:

In robotics, one of the most important and also basic tasks is to extract the appropriate information from robot, in other words we should be aware that how the robot can see or reacts according to the input. In order to get the data we need sensors. Usually, The data obtained from sensor should be filtered , analyzed or processed to get the best results by removing the unwanted ones. In our system model the major sensors types are as follow (although we have more detailed sensors):

2.4.1 Vision:

In order to observe the pallet frame , $\{P\}$, a set of fiducial markers have been installed on the front side of pallet. Camera would represent the position and orientation of pallet with respect to its reference frame, and as explained in introduction part (1.3), the location of the camera is fixed with respect to the body frame (Figure1), so it can be noted that the output of the camera will be, position and orientation of the pallet with respect to the body.

$$ {}^{B}T_{P} = {}^{B}T_{C} * {}^{C}T_{P} $$

In equation above, there is a calibration parameter ${}^{B}T_{C}$, That shows the constant transformation matrix between camera and body coordinates. The output data ${}^{B}T_{P}$ denotes the visual feedback information [11].

2.4.2 IMU sensors :

The Gyroscope and accelerometers are common sensors in electrical devices, but usually in robotics they do not give us sufficient information about the measurement of velocity, orientation and position, so to solve the problem these two sensor types are combined in an Inertial-Measurement-Unit (IMU) to provide more degrees of freedom, or in other words to have access to all needed directions; mainly for each direction there is one sensor. In robotic science, the IMU box can have different uses such as navigation. In navigation, the data output (roll, pitch, yaw rotations and accelerations) are fed into a computer to calculate the velocity and position of the current time. In GIM, IMU is installed at the center of the front axle. It includes 3-axes gyros and 3-axis accelerometer. Also there two other IMUs, installed on the boom and on the fork. Each include one gyro and two accelerometers. The IMUs measure the boom and fork angles with respect to the body [11]. (moreover, the telescope extension is measured using potentiometer.)

2.4.3 Wheel encoders:

Encoder is an electrical device attached to a rotating object to measure the rotation and covert that for calculating the linear velocity of machine or its acceleration or displacement. Each wheel of the GIM machine is equipped with two hall sensors to construct an encoder with a 17-teeth plate. The speed measurement is performed at 20Hz [11].

2.5 State Machine:

Finite state machine (FSM) is a mathematical model for computation employed in deigning computer programs and sequential logic cycles. It is considered as an abstract machine that can represent all the states of machine. The machine is in only one of the states at a time and it can change from one state to another when invoked by an event trigger or condition which is called transition. The concept of state machine can be observed in many real devices around us, that the device can perform different sequential actions as if a special event is triggered. A simple example can be an elevators, that two states of “going upward” and “going downward” can be executed when we press keys, or if it is overloaded, it is in state of “out-of-service”. In figure below (Figure 5) we can switch between states when appropriate transition for each state in executed.

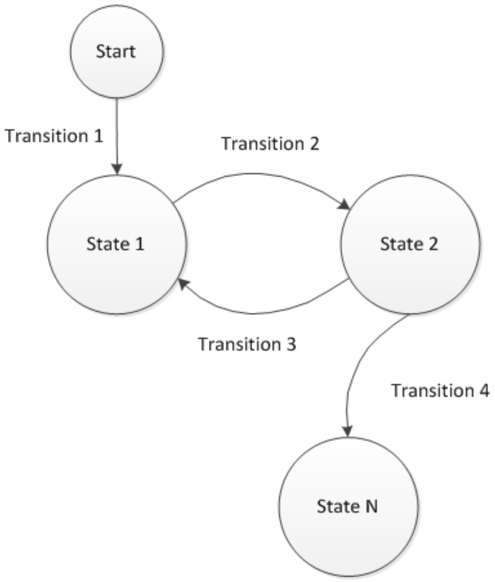


Figure 5-General structure of a simple FSM

In MATLAB/Simulink we have *stateflow* has the same role, in which we can define states and transitions graphically . Stateflow is an environment for modeling and simulating combinatorial and sequential decision logic based on state machines and flowcharts. In defining MATLAB stateflow the first step is to define all the expected states that can occur by considering all the system conditions. The next step is defining the appropriate commands which should be executed when each state is activated. To do so, the MATLAB has 3 main concepts in defining a command line in a state.

- Entry : by using entry command, the commands will be executed just one time

- During : by during, the commands will be executed as long as the state is active. So we can execute a command for than one time. This command can be used as a feedback control signal while it is sending command regularly.
- Exit : The Exit command would occur when the external or the output transitions are activated. By this command the current state will be passed and move to another state.

By defining all the necessary commands inside the states, the state connections or transitions should be announced. The transitions can be any data or an event that by occurring them the new commands will be executed. Generally, defining a stateflow can act as a graphical structure that makes sense to the users as a roadmap of algorithm.

2.6 General view for architecture of the system

The Autonomous systems are the combination of different subsystems. These parts are defined as separate and independent blocks in the architecture of the whole system, but enables the entire model to work well when they interact with each other or technically receive and send out appropriate messages to get information about interconnected block states. The current autonomous machine has the same structure, that different parts have been as a separate block at first, then connected to each other according to the data they need as input or output.

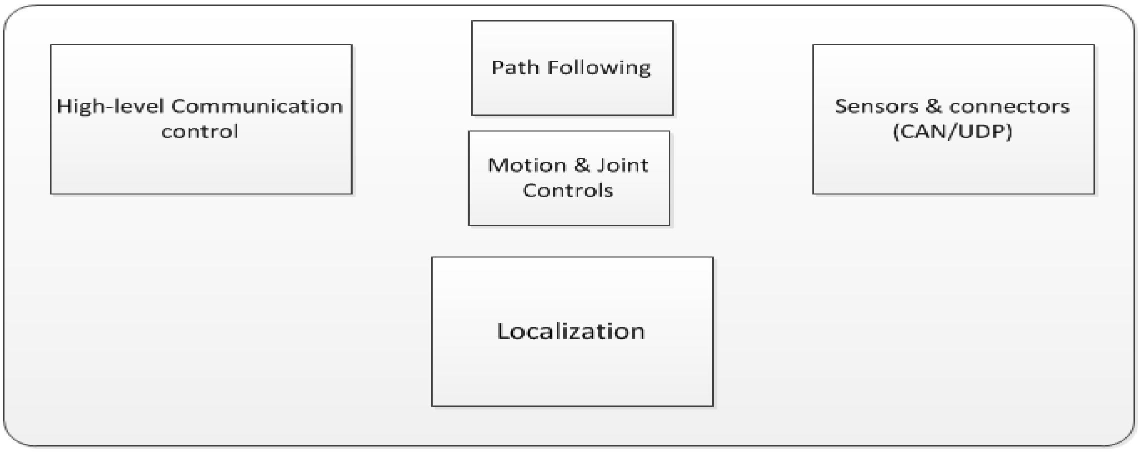

High-level Communication control
Path Following
Motion & Joint Controls
Sensors & connectors (CAN/UDP)
Localization

Figure 6- General structure of an Autonomous machine

As it can be seen above, the general structure of the GIM machine consists of different parts, that in real they have many subsystems inside and they all have connected signals as input or feedback control. In this system model, high level communication block receives the input command (it is entered manually by the operator; command is approximate location of target pallet) from the low level control (= a Linux PC) and the appropriate command types are produced, and also the state of the machine can be defined. Then the produced commands which are classified according to their type (part 2.9), would go through algorithms to follow the desired paths and also generate the controlling signals for the actuators such as driving parts like wheels or hydraulic parts; manipulators. All the necessary data is collected from the external or internal sensors (part 2.3), and it would act as a feedback control signal to measure the error signals in order to compensate them. The communication is done by CAN; CAN is a serial bus communication with multi capabilities by priority in message and message identifier [15]. The localization block has a critical role in detection of the machine position in the navigation process, just like all the autonomous machines the localization has the main and crucial role in system architecture.

3. SYSTEM FEATURES DESPRIPCTION

3.1 overview

The general behavior of the machine while receiving a command is described in this chapter and the steps that should be followed from the start point are described briefly.

At start point, the machine will be located in an arbitrary location far from the target pallet, called initial starting point. The approximate location of the pallet is sent to the machine and will be stored in the memory of the machine, which allows the machine to have a basic information about its destination. Based on the start and end points, the low level control system would plan a smooth path toward pallet, thus the machine starts moving to the pallet. Machine considers this path, the desired path, that is all planned in the global coordinate frame (chapter 2.3.1). As the machine approaches target, the pallet marker is detected for the first time, the machine plans a new smooth path to the pallet before any coordinate frame switch, so this new path will stay in global frame. Finally, as the machine gets nearer to the pallet, the coordinate frame will change to local mode. Typically, the idea of generating a new path while first marker detection, refers to the camera vision data. As the camera detects the markers, the position and orientation of the target pallet with respect to body frame will be acquired, it will be shown in result chapter that this detection might happen in far distances (distance between the camera and pallet) so because of this, the rotation estimation of pallet frame (consists of roll, pitch and yaw angles) will be unreliable, in other words based on this inaccurate data the orientation of pallet frame would not be correct in order to set the heading of the machine at this time. Thus this would be a good idea to stay in the previous coordinate operational system and follow the object just by the position data. Moreover, in order to control the system defined structure, a switching logic is considered for satisfying the desired steps and control each region part.

In figure below, the whole path since the initial start point until the pallet is picked, is shown graphically. The First marker detection point and the plan after this point can be observed. The whole process will be explained later.

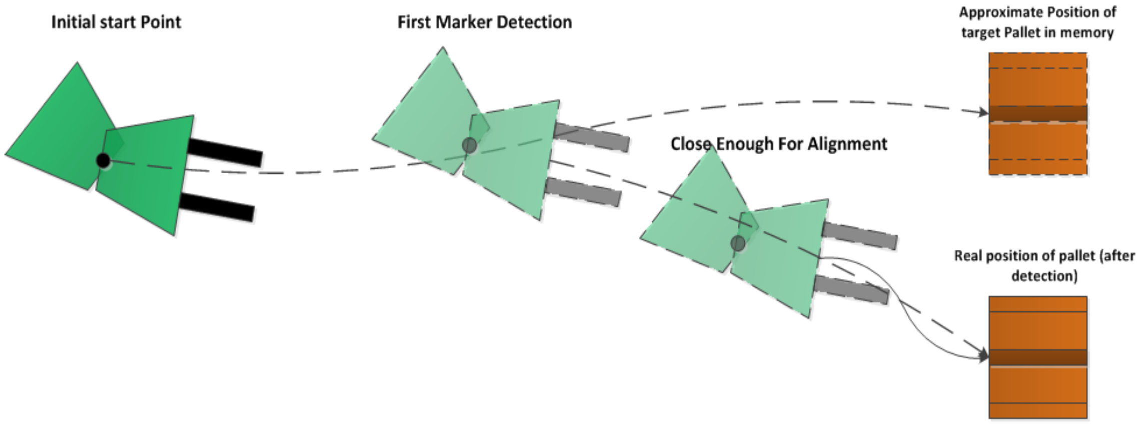

Initial start Point
First Marker Detection
Approximate Position of target Pallet in memory
Close Enough For Alignment
Real position of pallet (after detection)

Figure 7- General Procedure of Pallet picking steps

Visual servoing is the main research method for obtaining information about the location of the target pallet and getting data about the exact position and orientation of pallet. Application of visual servoing in a daily life for picking an object can be found easily, for example asking a person to take an object, Interpret this sentence into a logical sequential steps is: have general idea or the approximated initial position of the object, moving to the location, search the area for the target object and find it, and as we are close enough to the object pick it up [2].

The above steps that are used in daily life can be interpreted to the machine language into a sequential and logical executive program, that the whole procedure of pallet picking from start point to the end of that is controlled (Figure 7). It is necessary to mention that the real machine; AFS type; have the kinematic constraint, so defining the logical steps should be in a manner that we have enough space or have enough time to maneuver and avoid any unexpected distractions and have enough time to compensate the errors. Below, we will extend the above daily steps generally into how the real machine should be planned, more details will be discussed in implementation chapter.

3.1.1 Global Path following state

In the first step the machine moves toward the pallet as soon as the approximate location of the target pallet is given to machine. There have been defined an algorithm for path following toward the pallet. According to this algorithm, 6 control points $P0, P1, \ldots P5$;as Bezier control points; which show dimension of each point in *x-y* plane according to the fixed original base coordinate frame; are given into the path generation code. Moreover the speeds between each two consecutive points is defined by $V0, V1, \ldots V4$ .All these values are combined represented by “Drive command”. All the numerical data is fed into the Path following block. In path following block, the algorithm uses a 6-order Bezier curve to follow the desired path by calculating the error from the current position of machine from its desired one, and tries to reduce the error. So in the first step of logic sequence, the machine receives only “Drive” command that invokes the body for following the path.

3.1.2 Global Visual Servoing state

As the machine approaches the target pallet, the vision camera installed on the top of machine can see the markers of pallet. While this time, the camera gives an approximate position and orientation of the pallet and because of the distance, the orientation data is not reliable, however the position data seems more accurate. Thus based on the position of the pallet from camera and the current position of the machine, a new smooth path can be defined (figure 7). The new generated path is defined in a way that the focus is just going toward pallet, without any concern about the heading of pallet. Heading will be postponed to the next step as the rotation matrix data is more accurate, since rotation data is needed for alignment. According to previous step (2.8.1), new control point commands should be sent, and the path following algorithm block (figure 6) controls the machine for following generated path, and the corresponding control points are in global coordinate navigation, so this state is called *Global-Visual-Servoing* (GVS). Thus, as shown graphically in figure (7) the new path is supposed to be generated. The basic ideas and implementation of this step will be discussed more in implementation chapter (4.2.2).

3.1.3 Local Visual Servoing state

As the robot follows the new generated path in GVS, it approaches the pallet. While following the path, as soon as the visual object pose estimation becomes consistent, there will be a switch into the Local mode coordinate frame. The switching occurs for aligning the machine heading to the pallet orientation using the yaw angle, so a path is defined for picking and placement called Local-Visual-servoing (LVS). In local state, the origin frame is going to be changed to pallet, it means that the pallet position will be considered as zero. This step is shown in Figure (7) ; the last part.

3.1.4 Manipulation state

As the machine reaches to vicinity of pallet, the hydraulic manipulator of machine can be adjusted for picking the pallet. At first the robot puts the fork into the predefined desired position that is close to the ground, because the pallet height is so low and the fork must go into the pallet holes to grab it. The data to adjust the hydraulic part of robot still comes from the visual servoing sensor camera.

3.2 Commands:

In autonomous machines, sending or receiving the appropriate commands has the main role to have the desired behavior of machine. In order to control these commands, the first task is to define all the commands that the robot needs to interact correctly with the environment and its control units. Generally in this robot system, the actuation commands of the machine are divided into four different types, that machine send and receive each type in special states:

1- Drive command : This command is sent when we need normal movement of machine, or when we need the whole body of machine to have physical displacement. Actually, this command type invokes the machine to follow the path.

2- Visual servoing command : This command is sent when the object is detected, it means that when the pallet markers are recognized at least *n* times, the pallet tracking filter is initiated and VS is started[11]. This number *n* can be changed to make the vision accuracy higher or lower.

3- initialization command : This command is sent while resetting all the states of the system into the initial states. By initializing all the states defined in system model, all variables will be set to default values and system waits for the next new command.

4- Work Hydraulic command: This command is sent when the manipulators actuators of machine is needed to act. The manipulator consists of the boom rotation, telescopic elongation and fork rotation; 3 degrees of freedom. Generally, the manipulator actuations are activated when we need to align the forks to the pallet

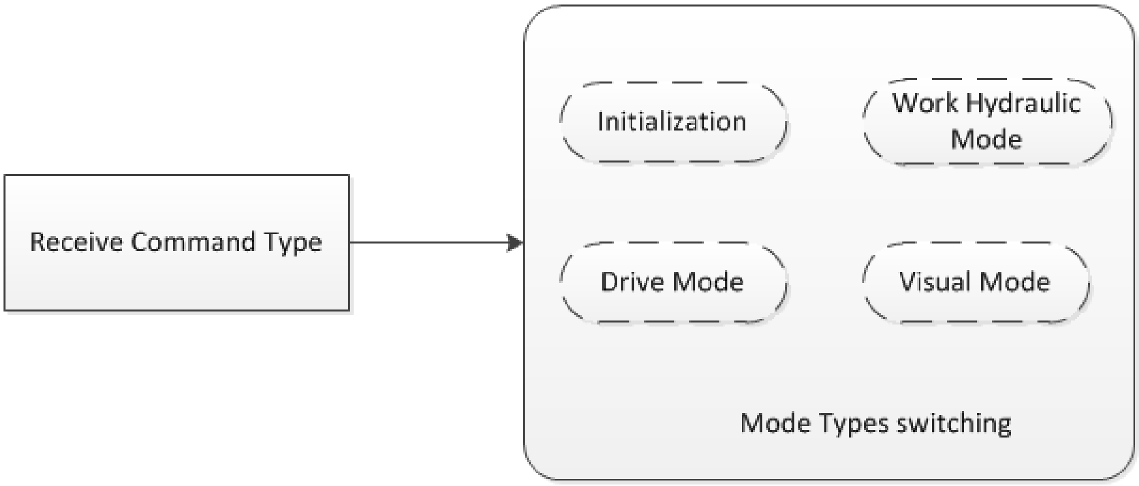


Figure 8- Command types witch modes

Generally, The numerical commands are received from low-level into high-level control, and as these values do not have any meaning for the system, they should be clarified according to the type command. However, by classifying the commands, we can switch between different actuation types, and the output reaction of the machine is changed by each type. Also all the robot commands are received by the xPC target control device, and while the system receives a command, robot controller would execute it immediately as if robot is free to do that, in simple words, for example while the robot is driving to the target, if it receives the next drive command or receives another command type, the command would be saved in robot buffer state, and when the previous command is done completely the new command will be executed. So, the robot can do any action just one time at a time. This can be solved by counting the internal commands. In order to count, all the commands are accompanied by an *ID* number, and each command is allocated by a special ID number and ID will increase after each command. Thus for example if command with $ID = 1$ is still running, the second command should wait until command one is finished. All these structures can be performed by a correct *stateflow* architecture. More details will be shown in implementation chapter.

4. IMPLEMENTATION :

4.1 Overview

This chapter will cover different aspect and features of the implementation in the project and the use of different previously introduced tools to cause an executable project. All the predefined theories and logics and hypothesis should be put into the experiment and test, to understand their capabilities and deficiencies. In order to do so, the first step would be implementation of all the ideas and logics in the programing language to prepare the machine to understand that. Simulation of the program is executed in MATLAB Simulink and the logical steps have been defined by *stateflow.* In 4.2.1 the general structure as the basis for implementation has been introduced, then the main focus areas are discussed, in 4.2.2, in more detail and the ideas are presented to handle the defined task.

4.2 control Architecture

As explained in the previous chapter (2.6), the autonomous system is the combination of different interconnected blocks that provide the necessary information for the final precise behavior of the whole system. One the these blocks is the high level control communication system, that generates the appropriate command signals for the execution of the machine and all the machine states can be defined here. Below, the main architecture of this block and the executed implementation on for this block is represented.

4.2.1 High level control communication

As mentioned above, the high level control block has the main role in producing appropriate command signals and clarifying machine state. This high level block consists of two main subsystems. In figure below the general structure of this block is shown. In the first sub-block; *command block*; numeric data is received from the low level control system, and it should be classified according to its type. To do that there has been defined a stateflow architecture that classifies the input data according to their type.

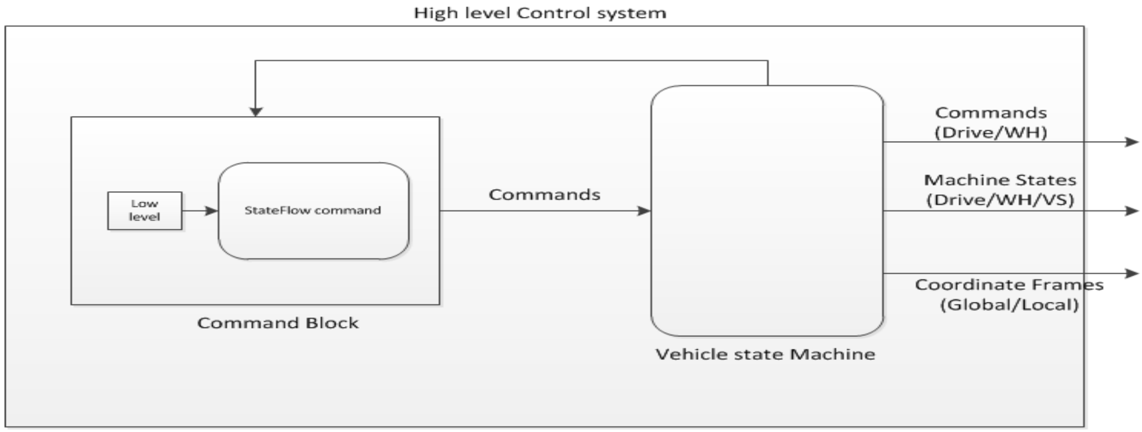


Figure 9- General structure of high level communication

In order to satisfy the defined role for the *command block* (sub-block shown on figure 9), the start task is to receive the commands from low level computer, this can be done by using UDP block; UDP block will be explained in section 5.2. Then, it should be noted that in order to generate the classified command, from the stateflow, it should be considered that if the machine is free to receive a command or being saved; in section 3.2 it was mentioned that one command is executed at each execution period. So according to figure 10 an input should be considered to be aware of machine being busy or free. This input is fed as a feedback from *vehicle state machine* block.

According to figure below, the input command and machine states would enter a stateflow structure to be evaluated according to their *command type*. The output as it can be seen, the appropriate command according to if it is drive command or manipulation command is obtained. Below the structure of the *command block* is represented.

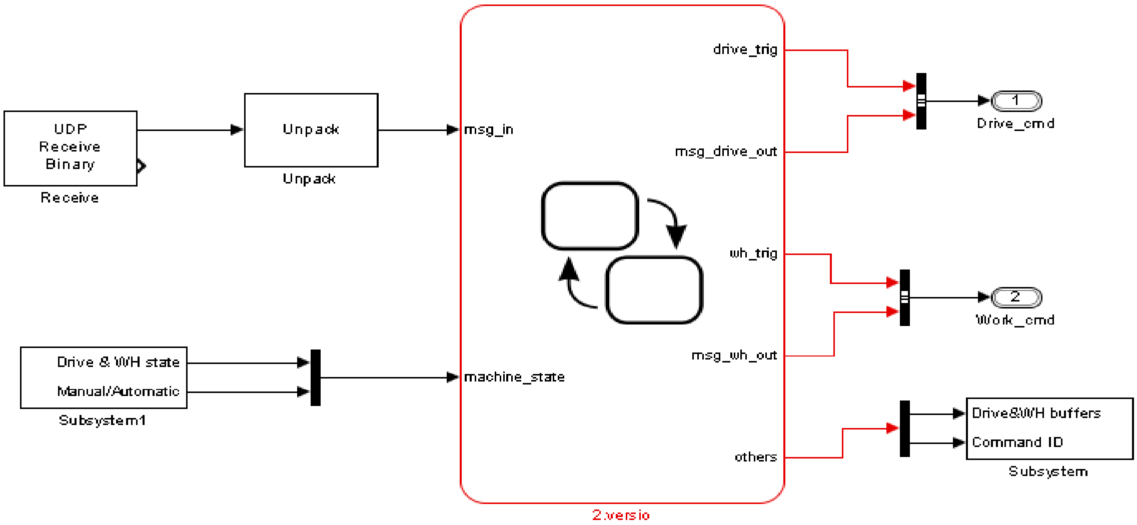


Figure 10- Structure of command block for classifying the commands

On figure above the command inputs and machine current state would go through a stateflow chart for classification of command, and accordingly the outputs would generate drive or manipulation commands step by step. Also it can be mentioned that the commands sent to xPC target can be monitored to be sure which command is sent to machine.

Another sub-block of the high level communication is *vehicle state machine*. This block has been promoted beforehand in IHA group and there are some improvements, as mentioned in 3.1.2 , the *global-visual-servoing* in pallet picking process is developed, that details and all the algorithms implemented for this step will be explained in the next part 4.2.2. Here, only the structure and architecture of how to design the controller is explained. In GVS, the camera data is being used as to follow the pallet target, so in the first step the vision sensor data should be input in this block; all the sensors data are defined in CAN/UDP block (figure 6); and based on the vision data, pallet position should be defined in global frame. The *vehicle state machine* stateflow can define the state of machine and each state has more substates, so to define a new sub-state; GVS; the structure of this stateflow should be enhanced.

The first task is to add a new input for the vision data in high level control block system. Then the vision data should be processed in order to find the visual servoing data with respect to the global coordinate frame; since the vision data shows the transformation of pallet to body frame as reference, so in order to change the reference there should be code for that, Also there should be defined a condition that when and how the machine states should be switched.

In the following figure below, the general structure of the purpose has been represented. The vision data-Bus is allocated as input, and it would enter a function block to change the reference frame (body frame) to a fixed frame; that is inertial global frame. And the next stateflow has the main role in how the states should be planned. As it is shown below, *vehicle state machine* stateflow receives the localization data, duration time; generated as feedback from outside blocks. The inside structure of the *vehicle state machine* and the concept of the mentioned function block will be discussed in the next section.

In the next part the implementation inside the *vehicle state machine* and the related codes will be discussed.

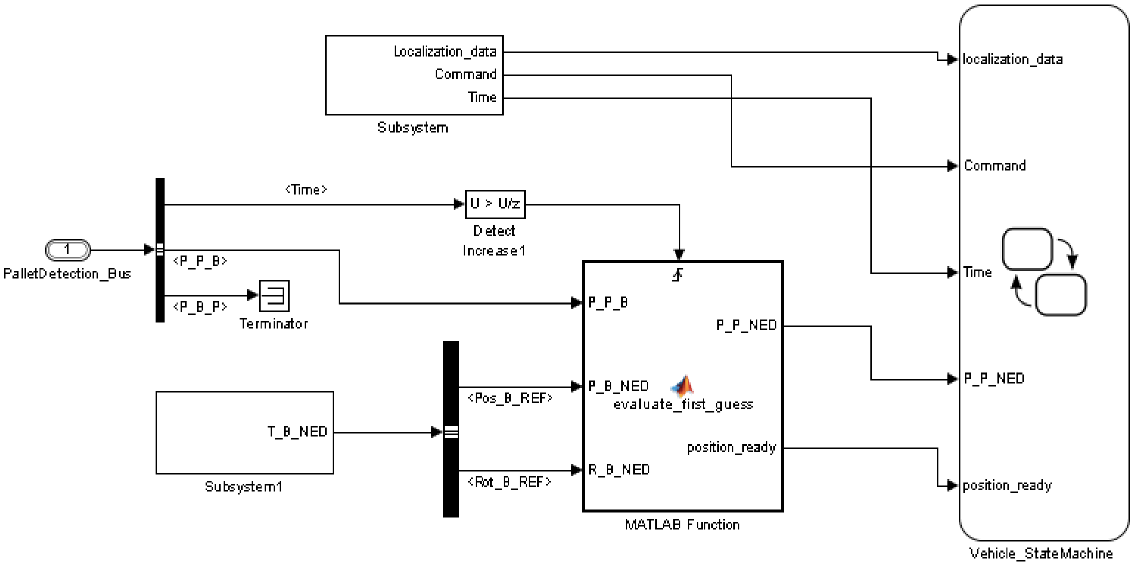


Figure 11- defined Structure for GVS step

4.2.2 Global Visual Servoing state

As explained in system features description section (3.1.1), while the machine approaches to the pallet, the installed fiducial markers can be detected. At this step; when the first detection occurs; here the pallet position with respect to the body frame is obtained; it has been explained in 3.1.2 that only the position data of the camera would be considered. All the coordinate frames should be defined in fixed origin inertial coordinate "τ". Therefore, the output of vision data should be converted to origin coordinate frame. To do so, we convert all vector transformations to one origin:

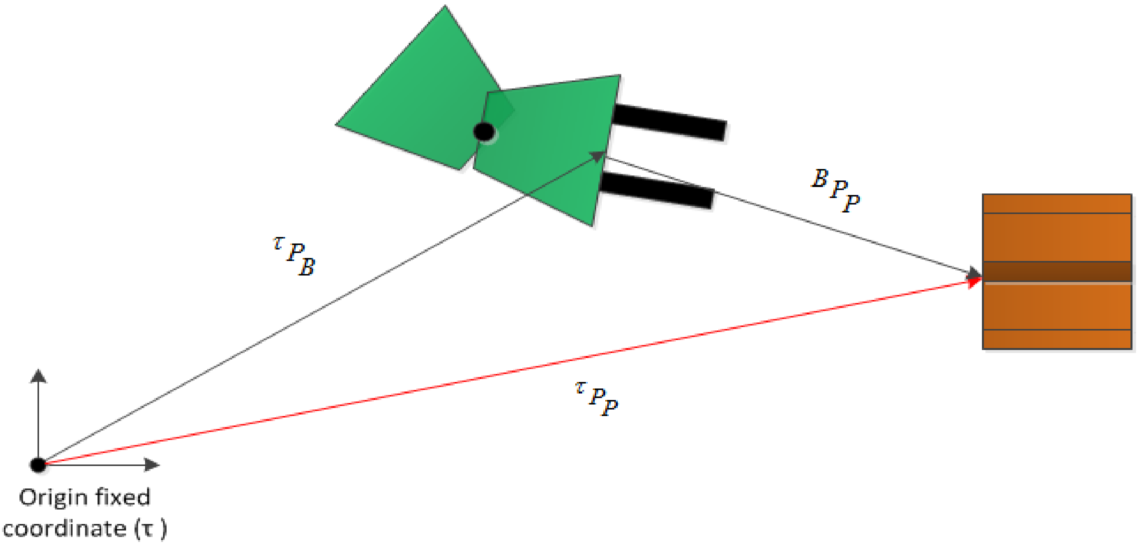


Figure 2- Pallet position with respect to the inertial frame

According to the graphical figure above, the machine is in a distance where the pallet is detected, so the vision sensor calculated the position of pallet to body at each time (${}^{P}P_{B}$); this value will be revised or compensated as the machine goes nearer to pallet. Moreover the position of machine according to the inertial frame (${}^{\tau}P_{B}$), is extracted from the global navigation localization algorithm. In this algorithm the dead-reckoning method and Kalman-Filtering method and the GPS data have been used to position localization the machine; refer to [18]. Now by a simple vector calculation the position of the pallet can be defined in the inertial coordinate frame. Now the *conversion equation* will be:

$${}^{\tau}P_{P} \;=\; {}^{\tau}P_{B} \;+\; {}^{\tau}R_{B} \,*\, {}^{B}P_{P}$$

According that when the camera observes the markers, this algorithms would tell us the position and rotation matrix of body with respect to the marker frames or in other words ${}^{P}P_{B}$, because this vision algorithm defines the pallet as the reference point and calculated the camera position according to that external reference. but in equation above, we need a conversion to find ${}^{B}P_{P}$. The purpose of this conversion can be found in our hypothesis; defining all frames with respect to original inertial coordinate frame τ.

According to conversion equation above, the position of pallet to origin frame (${}^{\tau}P_{P}$) is found, also the current position of machine (${}^{\tau}P_{B}$) is known. Then a new smooth path toward the pallet should be denoted in other words a Go-To-Goal path to update the path following. As explained in previous chapter, commands are sent by 6 control points and 5 velocities, and all these numerical values will go into path following algorithm to control the error signals all by a Bezier curve. To do so:

```
   P6x = P_P_NED_in(1);
   P6y = P_P_NED_in(2);
   P6 = [P6x; P6y];
   P1x = global_cord_in.Pos_B_REF(1);
   P1y = global_cord_in.Pos_B_REF(2);
    P1rot = global_cord_in.Rot_B_REF(1:2,1:2);
   P1 = [P1x; P1y]; % The first position a marker detected
    P2 = P1 + (P1rot * [0.5 0]');% P2 for heading
    P5 = (P1+P6)/2; % to make a straight line
    P3 = (P2+P5)/2;
    P4 = P5;

   Control_Points= [P1; P2; P3; P4; P5; P6];
v0=0.3; v1=0.3; v2=0.3; v3=0.3; v4=0.3;
drive_cmd_message = [Control_Points; v0; v1; v2; v3; v4; 1];
```

In code above, the control points are all defined in a way, starts from the current saved position $P1$ to the last pallet point $P6$. The first point is where the camera has detected object for the first time, that is obtained from the global coordinate navigation algorithm, and the last point is the pallet position obtained from vision data extracted from the previous equation. The second point is chosen $50cm$ away from the first point toward the current heading of the machine. This control point is chosen in order to set the new heading of the machine smoothly, without any immediate heading distortion. The rest points are chosen as the middle points of each two consecutive points in order to that machine does not need the heading to be aligned. The smoothness of this path will be shown in the next chapter.

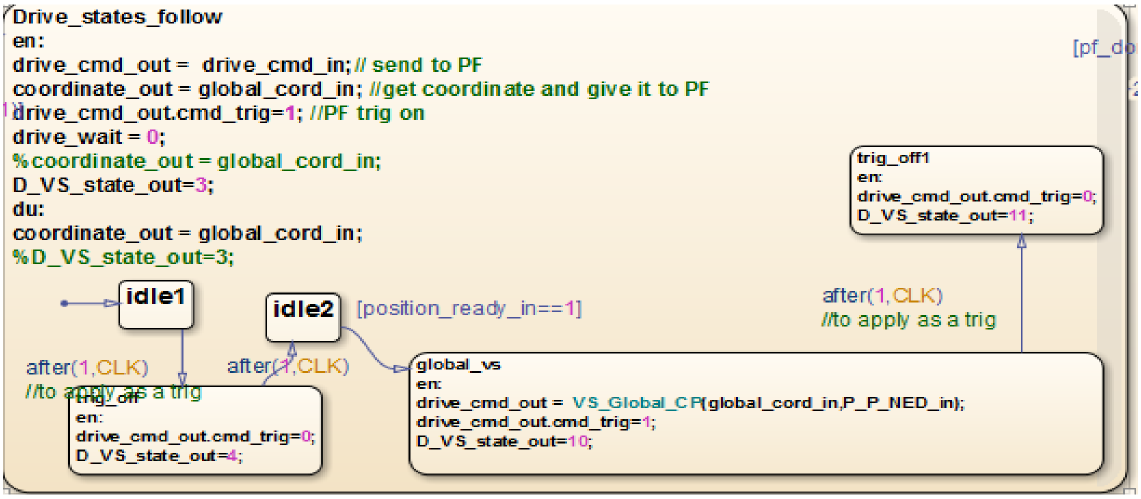
Another crucial note is the “flag” point. As it was mentioned before (section 3.1.1) any command type was combination of control points and the velocities between each two consecutive points. The general structure of command vector is:

Command= [message ID, message Type, Position points, velocity values, Flag]

- Message ID represents the number of commands which is sent or received. Actually messageID has the role of counting the sent/received commands. It is also useful to check that which command is running in the system at current moment.
- Message Type represents the type of sent/received command (section 2.9)
- The control position points and velocities (section 2.8.1)
- Flag represents that if we are in visual servoing mode or not. It acts as a sign to be sure that the visual servoing type is running at the moment. While the machine has simple path following; just like step one; flag number is zero, and As the visual seroving mode is on, the flag number will be one.

According to the explanation above, while detecting the pallet marker there is a jump from Global path following (GPF) to the Global visual servoing (GVS) state. But the key question is that, when does the camera see the object or in other words when this switch from GPF to GVS state occurs?!

To answer this question, a stateflow structure should be presented. Actually, using stateflow in the architecture of a control system can enable us to define multi functionalities of the system in one compact format and switch between them easily. The employed algorithm declares, when the difference between two consecutive approximated position of pallet to origin (${}^{\tau}P_P$); extracted by camera from conversion equation above; is less than a tolerance value (±20cm), the chosen value of pallet position will be good enough to switch into the GVS state. The critical point here is that, there might be uncertainty in pallet position. This can be ignored, since we start moving toward pallet from far distance and as we approach the pallet, the accuracy of data will be increased, as the errors can be compensated in the next step. After this step there will be better information about the pallet position and the heading of machine will be aligned in the next step since there is now enough space to have maneuver. Figure 13, shows the stateflow structure at this step.



```
Drive_states_follow
en:
drive_cmd_out =  drive_cmd_in;// send to PF
coordinate_out = global_cord_in; //get coordinate and give it to PF
drive_cmd_out.cmd_trig=1; //PF trig on
drive_wait = 0;
%coordinate_out = global_cord_in;
D_VS_state_out=3;
du:
coordinate_out = global_cord_in;
%D_VS_state_out=3;
```

```
trig_off1
en:
drive_cmd_out.cmd_trig=0;
D_VS_state_out=11;
```

```
trig_off
en:
drive_cmd_out.cmd_trig=0;
D_VS_state_out=4;
```

```
global_vs
en:
drive_cmd_out = VS_Global_CP(global_cord_in,P_P_NED_in);
drive_cmd_out.cmd_trig=1;
D_VS_state_out=10;
```

Figure 13- Algorithm of switching between GPF &GVS

According to this figure, while the machine is still in the first step; global path following; the machine receives the drive command as explained in chapter 3.1.1 and just as the new defined condition; *position_ready_in*; is activated the machine state will switch to the GVS state. At this state the new M-function *VS_Glob_CP* , the control points for generation of new smooth path are defined. One important thing is that, while we need a new command; here Drive command; there should be a trigger activation, it means that each command in any type, should be accompanied by a trigger activation. While the trigger is activated the command is sent otherwise command is passive. Typically the trigger is allocated for command in order to be sent only one time.

According to figure 13, the switching between GPF and GVS happens while the *position_ready_in* is activated. This condition is the representative of an algorithm that has the role of the output of the vision camera. In function below, this condition is defined.

```
persistent P_P_NED_old1 position_ready_old
if (isempty(P_P_NED_old1))
    position_ready = 0;
    position_ready_old = 0;
    P_P_NED_old1 = P_B_NED;
end
if position_ready_old == 1
```

```
        position_ready = 1;
end

if norm(P_P_B)<7

    P_P_NED = P_B_NED + R_B_NED*P_P_B;
    if norm(P_P_NED-P_P_NED_old1) < position_tolerance
        position_ready = 1;
    end
end
P_P_NED_old1 = P_P_NED;
position_ready_old = position_ready;

if alvar_ready > 0
    position_ready = 0;
    position_ready_old = 0;
    P_P_NED_old1 = zeros(3,1);
end
end
```

According to this algorithm two consecutive position of the pallet with respect to the body frame are compared to each other. If these two position differences are less than a defined variance ($20cm$), it can be considered that the position of pallet is found. So as these two values are less than the tolerance, the new position will be considered as the pallet position and also as the 6th control point. Generally, at this step the approximate position of pallet would satisfy the approach, and the exact location is not vital.

5. TEST AND RESULTS :

5.1 overview

This chapter would describe the evaluation of the implementation consequent results for a smooth and non-conflict pallet picking. All the algorithm and methods conducted in the architecture and in the process of implementation are employed for controlling the system behavior in a way that its reaction to the inputs, is feasible and controllable. So while implementing all the ideas and structures, the validation of those claims should be examined. The validation process in a scientific research can be reached in the combination of simulation results with the final real results in the environment, at first the simulation results should be verified then the real experiments should be done to investigate the effect on the real structure. In 5.2 the required input data to the system is defined and show that how the system can be ready for interacting with the simulation or the real machine and how to transfer data. In 5.3 the results extracted from the simulation system are shown in figures and tables and the accuracy of the ideas are explained and proven.

5.2 System inputs

The simulation structures are all planned and scheduled in the high level control system in a Simulink/MATLAB file. All these implementations should be compiled and then transferred and uploaded on the system computers in order to be able to run the simulation models and monitor their behavior. As mentioned before the initial position of the target pallet is given to the system and the machine behavior response can be observed according to numerical values, and the simulated area of system. To have this communication between operator and the system there should be a middleman, that the UDP connection can handle that.

UDP (user datagram protocol) is the main communication platform that enables computer applications to send and receive messages to/from other hosts on an internet protocol (IP) network with no pre-communication to operate special data paths. UDP uses a simple transmission structure with minimum protocols, and so exposures every reliability of the fundamental network protocol to user’s program[14],[16].

According to the UDP explanation the initial approximate data command can be sent manually to the machine to start the path following. In order to just test our algorithm, we just send a command for the last Bezier with flag equals 1, a explained in chapter 4.2.2 the last Bezier curve has a flag sign equal 1 otherwise it is zero. In figure 14 a series of command based on Bezier control points are given a input to the system, and by a constant trigger only one command will be sent to machine.

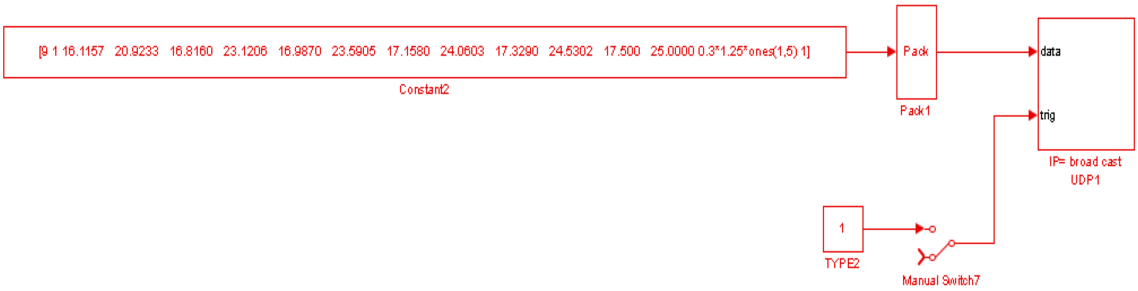

[9 1 16.1157 20.9233 16.8160 23.1206 16.9870 23.5905 17.1580 24.0603 17.3290 24.5302 17.500 25.0000 0.3*1.25*ones(1,5) 1]
Constant2
Pack
Pack1
data
trig
IP= broad cast
UDP1
1
TYPE2
Manual Switch7

Figure 14- manual Control points as input data

The same UDP connections can be used for receiving data from the simulation system, jut by allocating specific IP address to each end/receive connections. The necessary information from the system can be monitored and plotted, for example position of the machine with respect to the inertial frame (${}^{\tau}P_P$), the error signals, state of the machine, and all the other required data can be monitored during the process.

5.3 Simulation result

In this step we check the accuracy of the pallet picking process by calculating the appropriate error signals to check that how accurate the exact pallet position is estimated. In order to do the test the pallet has been located on a fixed arbitrary point according to the inertial global coordinate frame. The pallet approximate position is given to the machine by UDP connection. The machines starts moving as explained in chapter 3, and the required data is stored during the whole navigation process. The main focus on the data tests is while switching algorithms are executed and the pallet is grasped, to see the accuracy of them and observe how correct they work in different situations. Also several tests have been done, to make it harder for robot and test the robustness of system. In all experiments the exact position and orientation of pallet is $\mathrm{N} = 17, \mathrm{E} = 25, \mathrm{Yaw} = 70(\mathrm{degree})$; remember that the inertial global frame is based on NED coordinate frame.

5.3.1 First simulation test:

In the first test, the approximate location of pallet is considered 0.5m far away from the real pallet, thus the last Bezier control point is set to: $\mathrm{N} = 17.5, \mathrm{E} = 25$. First, the switching between states is plotted then the accuracy of picking is checked.

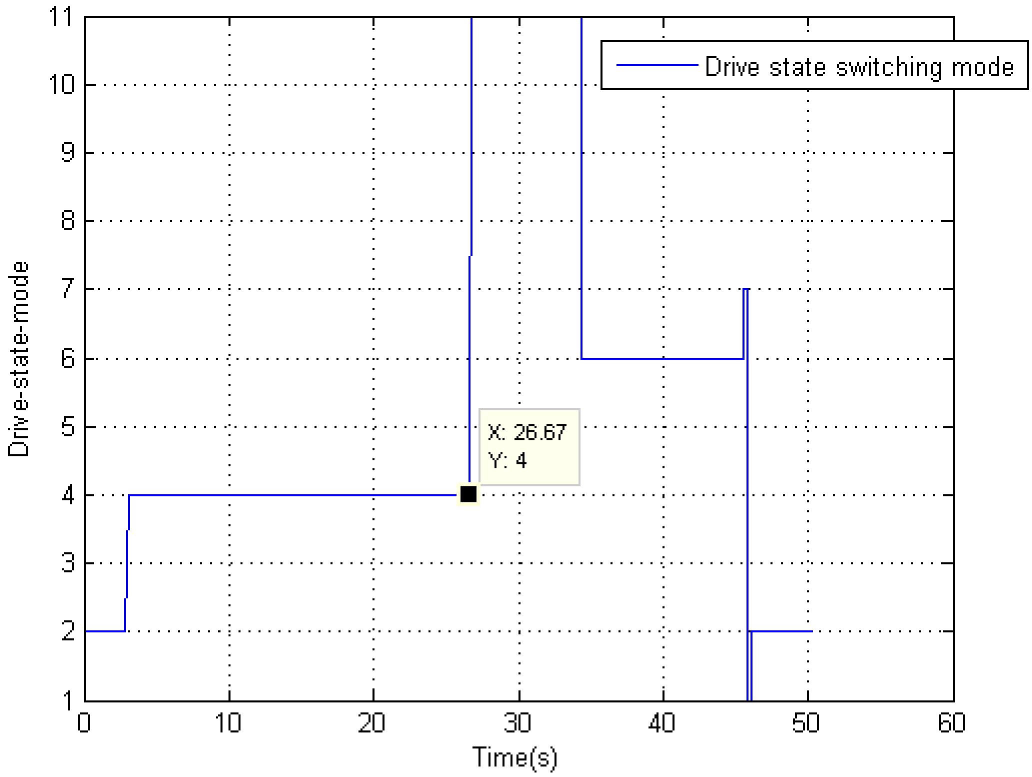


Figure 15- Drive state switching between different states (N=17.5,E=25)

As it can be seen in Figure 15, since the machine starts moving (Drive_state equals 4) and when it reaches to the target (Drive_state equals 2), the states have been switched. For defining all the driving states there have been defined arbitrary integer values for each driving state.

As it can be interpreted, the machine starts Normal driving or simple path following (3.1.1) while the machine is far from the pallet and the pallet markers are not recognized yet, the drive-state will be 4, just as what we have defined in our MATLAB *stateflow.* As the machine gets closer, the drive-state will switch to GVS mode and the switching occurs just by that the state number will be 11. while the machine gets closer to the pallet the local mode is switched, at this time the machine state is 6 . And finally as the machine picks the target the *idle* state occurs by state number 2. In figure above, for the first 3 seconds drive state is 2, it means that machine is in *idle* state and no command is sent, and machine is stationary, just like the final state of machine.

Another important parameter in a correct pallet picking is that the position of the body with respect to the pallet frame can be aligned in order to avoid any collision with pallet body. The important thing is that this data is extracted from the vision sensor camera, so the data can be evaluated since the pallet marker is detected.

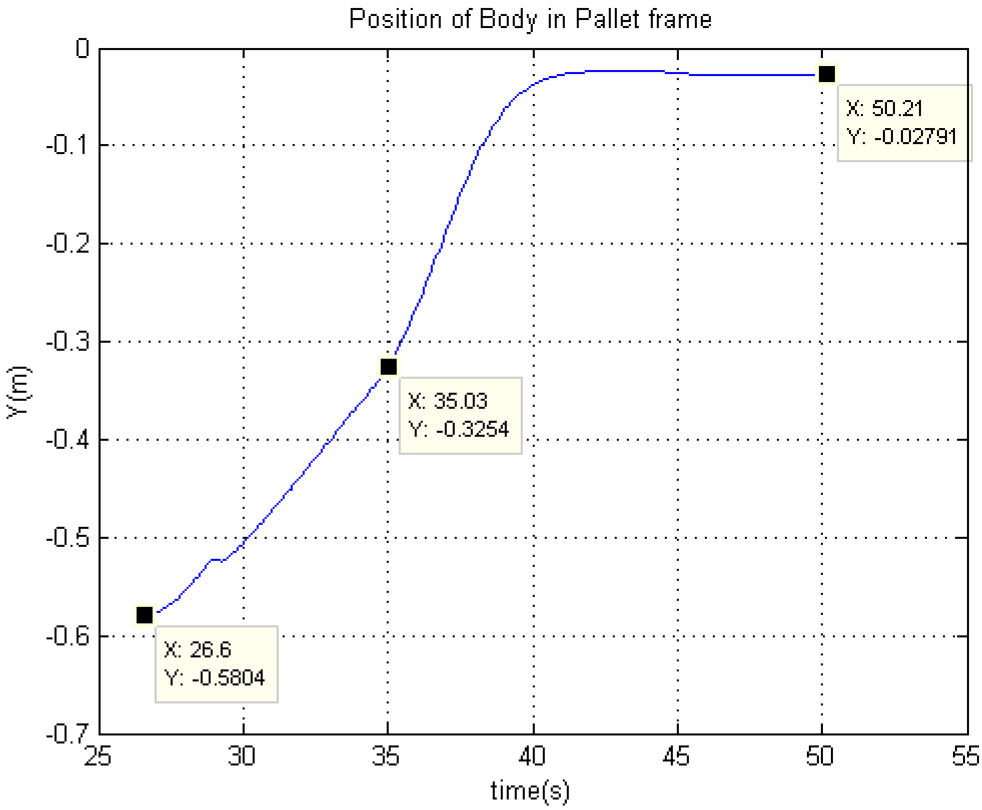


Figure 16- position of body in pallet frame in time domain after first detection

As it can be seen above, the position of machine body in pallet frame is plotted in time domain. By comparing this figure with figures (15), it is clear that this detection time corresponds in both cases which is at $t = 26.6s$, since the plot starts at this time, and previous state is ignored; as the object is not seen yet. Moreover, the final position is around zero; the error is less than 3cm which can be ignored; and this shows that the body frame is obeying the defined sequences and procedures. Also from the figure above, the switching state from GVS to LVS occurs at $t = 35s$ that corresponds to the figure (15).

As the machine enters the GVS step, just as explained in chapter 4.2.2, a series of control points should be defined for go-to-goal smooth path. As extracted from the numerical values of simulation, these control points will be:

P1		P2		P3		P4		P5		P6	
15.28	**18.8**	15.44	19.15	15.71	20.36	15.99	21.57	15.99	21.57	16.7	24.45

Table 1

The above control points are the the points for GVS mode, and to understand the role of the extracted points, the whole path following has been plotted in the next figure to prove this.

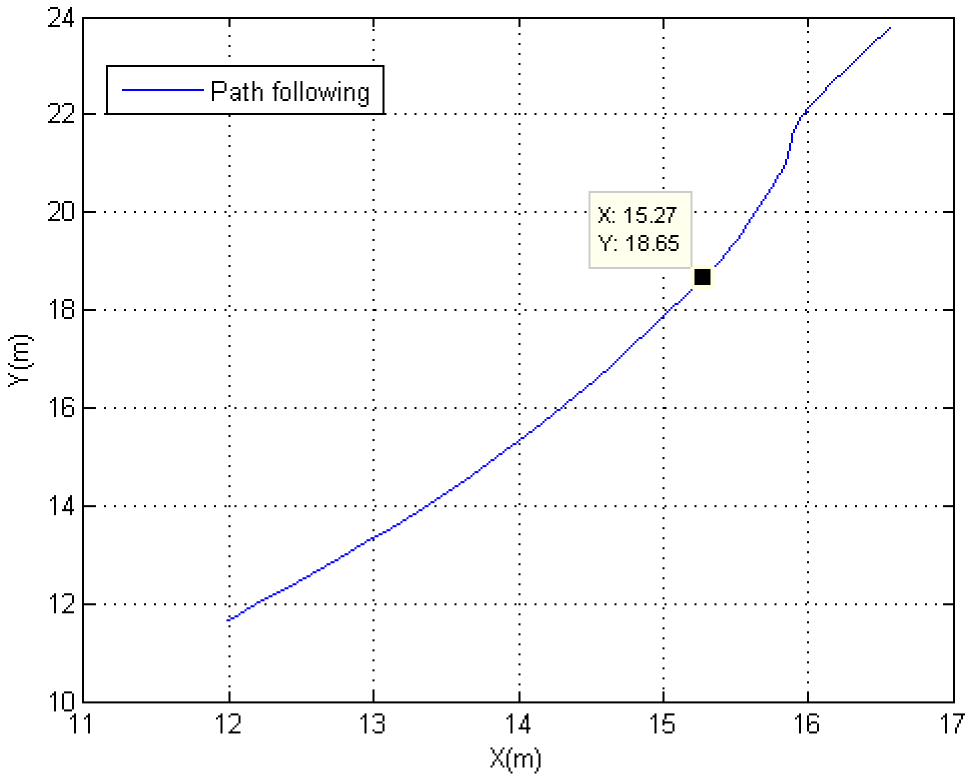


Figure 17- Path following of machine

The figure above represents the whole path machine passes toward the target from start point. From the start point (it can be any point), machine starts in global path following, then as it reaches to $x = 15.27$, $y = 18.65$ the state switches to GVS. The new smooth path in GVS ,explained in chapter 4, is executed and this new path after this point has different but totally smooth form from the previous path in GPF until switches to LVS mode. So it can be seen that selected control points have smooth behavior in changing the heading of machine. Finally the machine reaches to the desired pallet point, the key point in figure above is that, the final point is not the same as the pallet exact position which is $\mathrm{x} = 17$, $\mathrm{y} = 25$, but it should be noted the purpose of this project is that ${}^{\tau}T_p = {}^{\tau}T_F$ just as explained in 2.4. This figure plots the body frame coordinate in global frame, so the little difference with exact pallet position is compensated by the fork frame.

5.3.2 Second simulation test

In this simulation test , we try to make the experiment as hard as possible for the machine in order that, it will be harder to recognize the pallet and machine needs more maneuver to catch the target. To do so, the initial approximated position of pallet has been defined 2 meters far away from the real pallet. As an example the $x = 19m$, the other parameters are just like before. So, $N = 19m, E = 25m, yaw = 70degree$. All the previous tests are now repeated for the system and new figures are plotted.

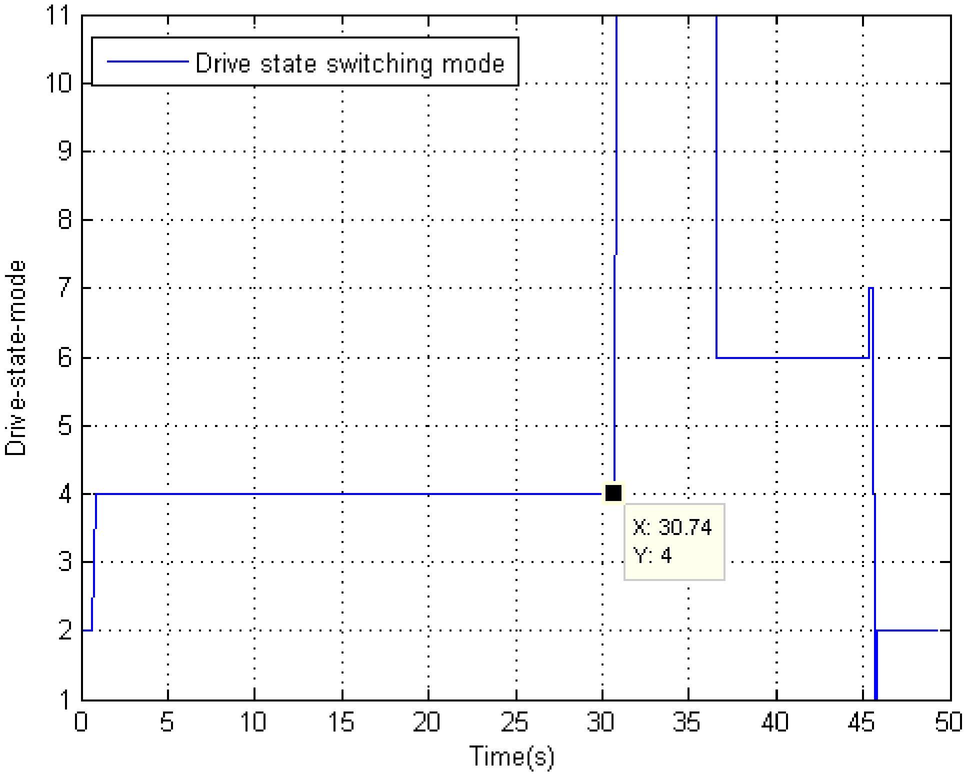


Figure 18-Drive state switching between different states (N=19,E=25)

According to figure above, by making the situation hard for the system, machine can still catch the desired pallet target. The key point in this figure with respect to the first test in figure 15 is that, the pallet detection time is delayed than the first test, so switching from GPF lasts longer. As seen above first marker detection occurs at 30.74s, that the mode switches; while in first test, when the approximate location was near the real position the detection time was at 26.67s. This detection time value difference can be even more in the real situation, but according to the method we conducted, we can be sure that it will still work in real environment. Also the real experiment is shown in the next part.

Just like the first test the position of body in pallet frame should be plotted, but the good point is that, by even making the situation harder for the machine, we still have prefect results.

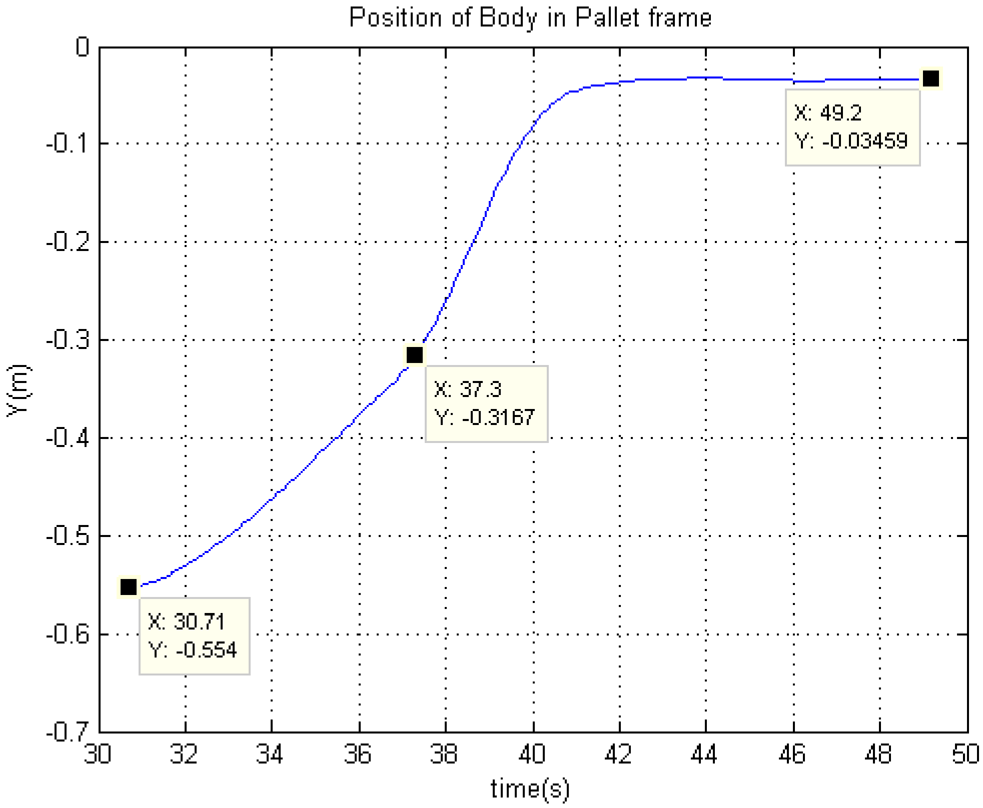


Figure 19- Position of body in pallet frame (N=19)

As it can be seen above, the position of machine body in pallet frame is plotted in time domain. By comparing this figure with figure (18), it is clear that this detection time corresponds in both cases which is at $t = 30.7s$, since the plot starts at this time, and previous time duration is ignored; as the object is not seen yet. Moreover, the final position is around zero; the absolute error is less than 4cm which can be ignored; and this shows that the body frame is obeying the defined sequences and procedures. Also from the figure above, the switching state from GVS to LVS occurs at $t = 37.3s$ that corresponds to the figure 18.

The same as the first test, As the machine enters the GVS step a series of control points should be defined for go-to-goal smooth path. As extracted from the numerical values of simulation, these control points will be

P1		P2		P3		P4		P5		P6	
15.25	**18.62**	15.44	19.17	15.74	20.37	16.04	21.56	16.04	21.56	16.78	24.43

Table 2

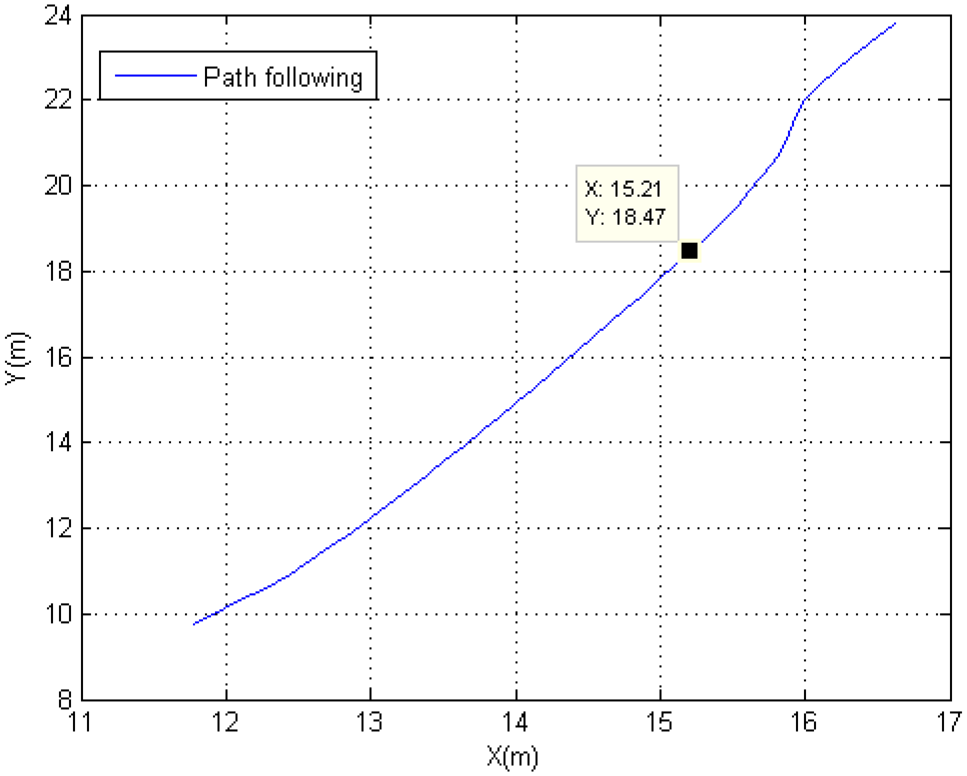


Figure 30- Path following of the machine (N=19)

The obtained control points and the path following figure can prove that the GVS path is executed correctly, and all the defined states can switch consecutively. The switching point from GVS to LVS occurs at the point shown in the table 2.

6. CONCLUSION :

This project covered the problem of controlling system for an autonomous machine in a target picking. The motivation of this project was the mechanization of the industrial areas and replacing the manual machines with the unmanned ones to increase the efficiency and reduce the price.

At first, the general information about the basics of the project and the machines and devices which we are working on, has been explained. Then step by step the necessary theoretical information needed to understand the main concept of the project was defined. Then the major performance of the current system was explained; so that the machine starts from the start point in a global frame coordinate by the approximate knowledge of pallet position, then as soon as the pallet is recognized by camera, machine receives data about the position and orientation of the pallet. Since at this distance the pallet angles respect with body are not reliably estimated, this angle cannot be used for heading alignment. So just using the position data would be enough at this state, which is employed by going toward the pallet in a smooth path. The adding of this step would increase the accuracy and robustness of the final pallet picking goal. Typically, this section could be a good relation between the defined theories and the general information of the whole system performance, which makes the system understandable.

On the implementation chapter, the MATLAB/Simulink software was used to program and create the programmable and executable sequential system, since the current machine has different states that can be run individually and the states can switch between each other as the commands change during operation of the system. As a simple description of the focused areas implementations, at first the generation of the appropriate command signals had to be solved. It means that, the commands should be classified according to their *type*, so a specific architecture was developed for classification and keep the commands execution sequence, and define correct inputs and outputs. Also another task is to use the vision sensor data to perform a path toward object. To do so, using the position data of the camera that displays the pallet position with respect to the machine body is employed. Thus, all the frames, body, camera and pallet are defined in a fixed inertial frame, and the current position of machine and pallet would give the new path control points. Also there should be a logic represents the switching time to the visual servoing state. And this logic is based on the comparison of two consequent position of pallet with robot's motion. Generally, by this implementation the robustness and performance of controlling system would increase.

And in the last chapter, the employed methods performance are tested in two different initial pallet positions to check the capability and robustness of the method. And the output results show the expected results. It is shown that the current method has enough reliability and the results even in difficult situation are good; when pallet initial position is about 2 meters away from the real pallet position.

REFERENCES

[1] Ronald Siegwart, Illah R. Nourbakhsh, “Introduction to mobile robots” Book , Massachusetts Institute of Technology Press, 2004

[2] Mohammad M.Aref, Reza Ghabcheloo, Jouni Mattila “A Macro-Micro Controller for Pallet Picking by an Articulated-frame-steering Hydraulic Mobile Machine”, IEE, International Conference on Robotics & Automation (ICRA) 2014, Hong Kong, China

[3] Finnish center of Excellence in Generic Intelligent Machines (GIM) research, http://gim.aalto.fi

[4] Reza Ghebcheloo, Mika Hyvönen, ”Modeling and motion control of an articulated-frame-steering hydraulic mobile machine”, 17th Mediterranean conference on control & automation, June 2009

[5] F.Bergamsco, A.Albarelli, E.Rodola,A.Torsello, “RUNE-Tag: a high accuracy Fiducial Marker with strong occlusion Resilience”, IEEE conference, Computer vision & Pattern Recognition (CVPR), 2011

[6] G.Flandin, F.Chaumette, E.Marchand, “Eye-in-hand/Eye-to-hand cooperation for visual servoing”, IEEE International Conference on Robotics & Automation (ICRA), April 2000, San Francisco

[7] V.Dworak, M.Huebner, J.Selbeck, “ Precise Navigation of Small Agricultural Robot in Sensitive Area with Smart Plant Camera”, Journal of imaging 2015

[8] W.Xie, Z.Li, X.Tu, C.Perron, “ Switching control of image-based visual servoing with laser pointer in robotics manufacturing systems”, IEEE Transaction on industrial electronics, Vol 56, Feb 2009

[9] A.Cretual, F.Chaummette, P.Bouthemy, “ Complex object tracking by visual servoing based on 2D image motion”. International Conference on Pattern Recognition.

[10] Danica Kragic , “ Visual Servoing for Manipulation: Robustness and Integration Issues”, Doctoral Dissertation, KTH university, Stockholm, 2001

[11] Mohammad M.Aref, R.Ghabcheloo, A.Koulu, M.Hyvönen, K.Huhtala, J.Mattila “Postion-based Visual Servoing for Pallet Picking by an Articulated-frame-teering Hydraulic Mobile Machine”, IEEE, RAM 2013

[12] B.thuilot, P.Martinet, L. Cordesses, J.Gallice “Position based visual servoing: keeping the object in the field of vision”, IEEE , ICRA 2002, Washington DC

[13] Rudy Negenborn, “ Robot Localization and Kalman Filters on finding your position in a noisy world”, Master Thesis, UTRECHT university, 2003

[14] P. D. Groves, Principles of GNSS, inertial, and multi sensor integrated navigation systems, 2013.

[15] S.h. Won, W. W. Melek, F. Golnaraghi et al., “A kalman/particle filter-based position and orientation estimation method using a position sensor/inertial measurement unit hybrid system,” Industrial Electronics, IEEE Transactions on, vol. 57, no. 5, pp. 1787–1798, 2010.

[16] F.A.Estiri, “ 3D object detection and tracking based on point cloud library special application in pallet picking for autonomous machines”, TUT, Master thesis, 2014

[17] P. Corke and S. A. Hutchinson (August 2001), "A new partitioned approach to image-based visual servo control", *IEEE Trans. Robot. Autom.* 17 (4): 507–515

[18] R. Ghabcheloo, M. Hyvönen, J. Uusisalo, O. Karhu, J. Järä, K. Huhtala, “Autonomous Motion Control of a Wheel Loader,” ASME Dynamic Systems and Control Conf. and Bath/ASME Symposium on Fluid Power & Motion Control. October 12-14, 2009. Hollywood, CA, USA.

[19] Fredrik Andersson, “Bezier and B-spline Technology”, Book, 2003