# DESIGN AND DEVELOPMENT OF LEAD-THROUGH PROGRAMMING METHOD USING LOW COST INCREMENTAL ENCODER FEEDBACK

**SAMEH MOHSEN OMER KANZAL**

**A report submitted as a partial fulfillment of the requirements for the degree of Mechatronics Engineering**

**Faculty of Electrical Engineering**

**UNIVERSITI TEKNIKAL MALAYSIA MELAKA**

**June 2015**

“ I hereby declare that I have read through this report entitled “DESIGN AND DEVELOPMENT OF LEAD-THROUGH PROGRAMMING METHOD USING LOW COST INCREMENTAL ENCODER FEEDBACK” and found that it has complied the partial fulfillment for awarding the degree of Bachelor of Electrical Engineering (Mechatronics)

Signature: .......................................................

Supervisor’s Name: DR. MUHAMMAD FAHMI BIN MISKON

Date: $24^{th}$/6/2015

I declare that this report entitled "DESIGN AND DEVELOPMENT OF LEAD-THROUGH PROGRAMMING METHOD USING LOW COST INCREMENTAL ENCODER FEEDBACK" is the result of my own research except as cited in the references. The report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature:………………………

Name: Sameh Mohsen Omer Kanzal

Date: 24st/June/2015

# DEDICATION

I would like to express my gratitude to my supervisor: DR. MUHAMMAD FAHMI BIN MISKON for his sincere guidance along my project. I would also like to thank my panels and lecturers for their continuous contributions that made this project possible.

Lastly, I would like to thank the special gifts have been given to me by Allah, my Father, Mother and Sisters, for their prayers and support along the journey of my study. I would never be able to compensate them for whatever they have been doing to me along my bachelor degree journey.

## ABSTRACT

Recently robots are widely used in various fields particularly in industry. Despite this fact, robots still require an undeniable amount of knowledge from the operators or workers who deal with them. As a result, robots cannot be easily programmed if the operator or the worker is not well-experienced in robotics' field. One of the programming methods that has been introduced to make programming task user-friendly is lead-through robot programming. However, the existing lead-through programming methods still require an amount of knowledge that is not available for most of the operators and workers. The main objective of this project is to design a lead through programming method for point-to-point robots' programming using inexpensive incremental encoder feedback, which can record, save and playback the robots' motion while considering the accuracy and precision of the motion. To validate this method, an experiment was conducted in this project, where an operator manually moves a two DOF (degree of freedom) robotic arm on a white board while the encoder feedback was recorded and later the same motion was played back by the robot. Then both recorded and playback trajectories were compared and analyzed. The results show that the played back accuracy is 96.17% for motor 1 and 97.86% for motor 2 with a standard deviation of 0.9593 for motor 1 and 2.33583 for motor 2.

# ABSTRAK

Dewasa ini robot digunakan dalam banyak aktiviti manusia terutamanya di industri. Walaupun robot banyak digunakan, ia masih memerlukan operator dan juruteknik berkemahiran tinggi untuk digunakan. Kesannya, robot sukar di programkan. Salah satu usaha yang memudahkan program robot ialah dengan kaedah lead-through. Bagaimanapun, kaedah ini masih memerlukan tenaga mahir dan kosnya tinggi dengan sensor dan alatan tambahan yang mahal. Justeru, objektif projek ini adalah untuk mereka kaedah program lead-through menggunakan incremental encoder, yan gboleh rekod, simpan, dan main semula pergerakan robot. Untuk tujuan validasi, eksperimen dijalankan dengan seorang operator menggerakkan 2 DOF robot di atas sekeping papan putih dengan bacaan enkoder direkod dan dimain semula. Trajektori yang direkod dan yang dimainkan di bandingkan dan dianalisa. Hasil kajian menunjukan ketepatan motor 1 dan motor 2 ialah 96.17% dan 97.86% dengan standard deviation sebanyak 0.9593 dan 2.33583 untuk motor 1 dan 2.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF APPENDICES

# CHAPTER 1

# INTRODUCTION

## 1.0 Overview

This chapter includes motivation regarding the continuous evolution of robotics existence in a diverse of fields, occurrence of lead-through programming method, Problem statement, objectives of this paper and the scope along with the expected limitations of the project.

## 1.1 Motivation

In the last two decades, robots have gained enough technological concern and public acceptance to shift from revolutionary concept to an evolutionary development that remarkably attracts developers' and operators' attention [1], Figure 1.1 shows the world annual supply of the industrial training and how it has dramatically increased.

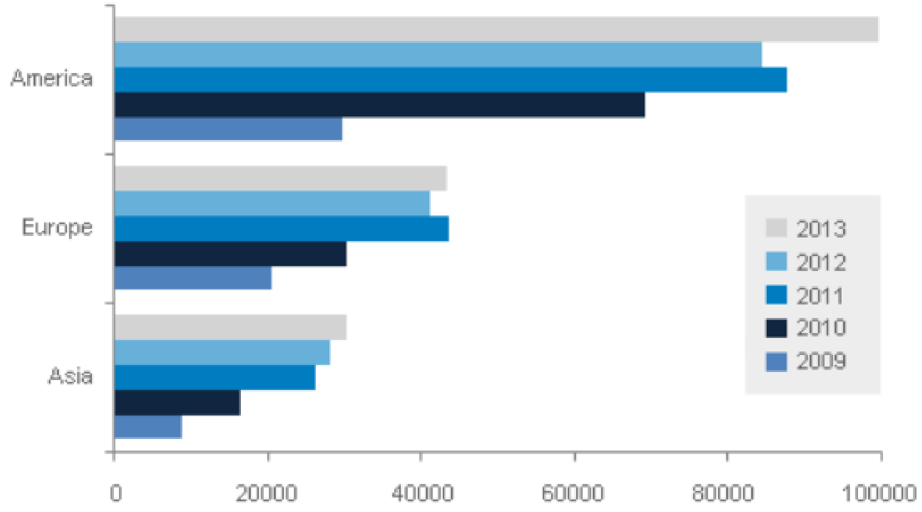


Figure 1.1: World Annual Supply of Industrial Training by Region 2009-2013 [2]

Motion planning problem is the main concept that hooks robots developers' concern, this type of planning is known as a trajectory generation. Recently, a diversity of human-friendly robots and partner robots have been developed for the aim of interaction between human and robots in various fields. These robots require intelligent capabilities to support the human-robot interactions [3].

On the other hand, many people are afraid that robots are replacing the human being jobs. But in fact they are relieving humans from various tedious, routine and even dangerous jobs. One of the widely spreading jobs that are being taken over by robots is welding process, as robots have recently replaced human in such jobs, as they are considered extremely hazardous, in terms of noise, intense generated heat and ultraviolet light form the welding torch [4]. As a consequence of the above mentioned various applications used, where robots are implemented in, robots and their motion planning, termed as a trajectory generation, have been given a remarkable attention and undetached part of human being daily life.

Robots Programming can be complicated and time consuming in terms of their motion planning, thus the process of simplifying robots' motion programming has been a top-priority for robotics' industry since the inception of the first industrial robot [13]. Consequently, making robots affordable to everyone, including those who are not well-experienced with robotic systems' basic knowledge has gained a non-deniable concern. Based on that need, a new trajectory generation method was proposed on 12th of August 1994 by Timothy L. Graf, lead-through teaching method [14]. It relied on the concept that the operator moves the robot and meanwhile it records the motion data and then saves it for a further playback of the same motion applied by the operator. By applying this method to the field of robotic systems trajectory generation industries, the affordability, feasibility and even efficiency will be ensured. Moreover, it gave the robots a sense of human as they detect the motion and then play it back without any effort from the operator during the playing back mode.

## 1.2 Problem Statement

Currently, robots' trajectory generation is planned and designed by engineers and designers. In other words, manipulating robots' trajectories using a joystick or keypad on a teach-pendant is not easy for a limited skills and experiences operator [9]. For example, if an industrial company needs to change the position and orientation information of the robots used in its industry it will have to contact the manufacturer of its robots to adjust this information using software or whatever method that is used to program their robots. Such procedures make it a bit burdensome for SMEs (Small and Medium Enterprises) to handle and deal with, especially when changes are needed more frequently. As a result, the need for an easier trajectory generation concept, which can be handled with a wider range of workers and operators became vital. Despite the fact that a lead-through programming method, using a teach-pendant, is able to give an operator or a worker the ability to handle the generation of a robot's trajectory, the initial position ($Q_0$), final position ($Q_f$) and the time required to achieve the trajectory (t) are still knowledge-demanding variables and require a certain level of robotics knowledge [10]. Moreover, the accuracy of the lead-through programming method became very critical when the robotic arm is required to pick and place an object, as any inaccurate recording information may result in a failure for the robotic arm to grab the object and place it to its exact final position. Consequently, a lead-through programming method that requires only a physical effort from the operator, which eliminates the need for a teach-pendant, can solve the knowledge limitation of

SMEs' operators. i.e. the operator in such a programming method is required to only deal with simple switches and physical movement of the end effector.

## 1.3 Objectives

The objectives of this project is to:

i. Design and develop a lead-through programming method for a robotic arm that can record the initial and final positions, save them and then repeat them as accurately as possible.
ii. Analyze the performance parameters of the lead-through programming systems in terms of error, accuracy, encoders' pulses detection-speed and precision.

## 1.4 Scope

This project develops a trajectory generation using a lead-through programming method for robotic systems used in SMEs (Small and Medium Enterprises). The project aims to produce a prototype of a robotic arm with two DOF (degree of freedom) that is able to record the initial and final position of the end effector as moved by the operator and then play it back when required. The performance of the designed system is discussed in terms of error, accuracy, encoders' pulses detection-speed and precision. For the experiment and analysis, an Arduino DUE controller is used to interface the developed system and control the trajectory of the arm based on the motion of the operator. The results of this project is a robotic arm moved manually to a desired final position ($Q_{rec}$) and then repeat the same motion by itself to the same final position ($Q_{played}$).

# CHAPTER 2

# LITERATURE REVIEW

## 2.0 Overview

This chapter contains theoretical background of the trajectory generation, methods of trajectory generation, lead-through programming method and its problems along with the available proposed solutions and the summary of the solution related to the proposed idea in this project.

## 2.1 Theoretical Background

Figure 2.1 shows a schematic of a robotic system in general.

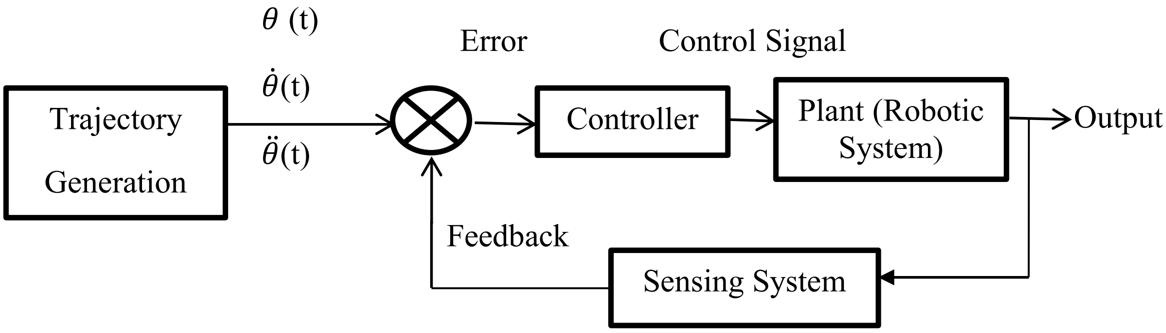


Figure 2.1: Robotic Systems Block Diagram

Robots programming has gained an undeniable concern for the few past years, due to their daily corporative and interactive applications offered to their users and operators.

According to [5], trajectory refers to a time history of position, velocity and acceleration for each degree of freedom. The term trajectory generation is not only generating a path for a tool frame to be located within a tool frame, but also includes the human interface issue with the robot' path specification [5]. For example, if an operator wants to change the location of the robot within a specific space then he may want to be able to specify nothing more than the location and orientation of the end effector and then let the system decide the other information required for that motion, such as duration, velocity and other details.

By assuming the motion of the manipulator is considered as a tool frame, T, and its space is the station frame, S, then the trajectory generation is, in general, changing the position and orientation of the tool frame from an initial value, $T_{initial}$, to an end value, $T_{final}$, relative to the station frame [5].

In some applications, it is vital to specify the motion of the tool frame in more details. For example specifying the sequence of the desired via point (intermediate points between the initial and final position). These via points are considered as a set of intermediate points carry out the position and orientation information of the tool-frame relative to the station-frame [5].

For further elaboration, most of robotic systems have a common block diagram, shown in figure 2.1, which explains and illustrates the system general input and output and then the sub-blocks which include the processes involved in both input and output.

As mentioned above robots programming has been given much attention. Recently robots have been involved in most of nowadays activities, such as industrial, human services, and even rehabilitation systems. For these reasons a pathway for robots is vital to be studied and determined as well as the human interface issue which indicates how the robotic system does receive its pathway from human.

Trajectory generation is a very wide term that includes many problems which need to be studied independently, not to deny that spatial, time and smoothness are the most relevant problems to trajectory generation.

- **Spatial**: the orientation and location of the end effectors, and how accurately they reach their destination.
- **Time**: how long the end effectors take to reach their desired destination.
- **Smoothness**: identifies whether the robotic system vibrates while moving from the initial to the final station-frame. And how smooth its motion is.

For the purpose of solving such relevant issues many studies have been introduced to contribute to this field. Below are some of these studies, introduced in details.

## 2.2 Methods to Generate a Trajectory

As shown in Figure 2.2, in [6] robots programming methods are divided into three main categories, despite the fact that over 90% of the robots are programmed using the first method, teach method, lead method and off-line programming.

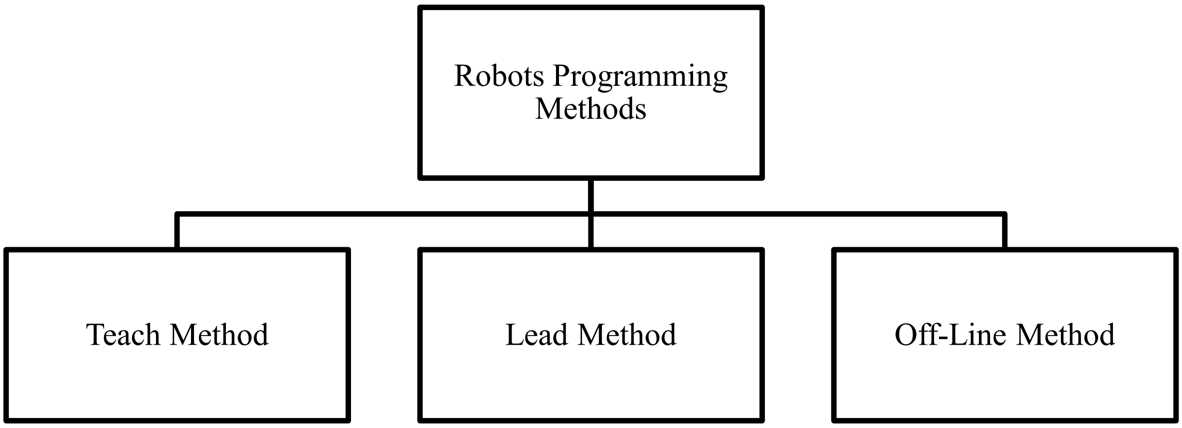


Figure 2.2: Robots Programming Methods [6]

- **Teach Method, On-Line**

The program is generated using either a menu-based system or a text editor. The main characteristic of this method is that the robot is thought how to change its position and/or orientation in a number of different co-ordinate systems to a desired location. This method of programming is simple to be implemented when simple movements are required, but its main disadvantage is that the robot will be out of service during the programming session. Example of teach method is Tiji trajectory generation [7].

- **Off-Line Programming Method**

This method is similar to the teach method in terms of the program build up, except that there are additional tools used to process the CAD (Computer Aided Design) data of the components and generate a sequence of information to be processed. The advantages of this method over the other methods are as follows:

i. Reduce the programming time.
ii. Makes the programming easier.
iii. Enables concurrent engineering and reduces product lead time.
iv. Allows process optimization.

An example of a trajectory generation using off-line programming is in [8].

- **Lead Method**

This method is a physical movement of the robot itself by the operator, during that movement the robot records the movement of its joint and then plays it back. This method is limited to small and medium size robots only, as it is difficult to physically move a large-size robot [8].

## 2.3 Lead-Through Programming Method

Figure 2.3 illustrates the process of recording data where the operator moves the robot manually by either using one of the interface devices, mentioned below in the problems section, or moving it physically. During the robotic system movement, the transducers attached to the system record the movement's data and store it in either RAM or external memory card for further processing. As a final stage, the data recorded will be processed for the trajectory generation.

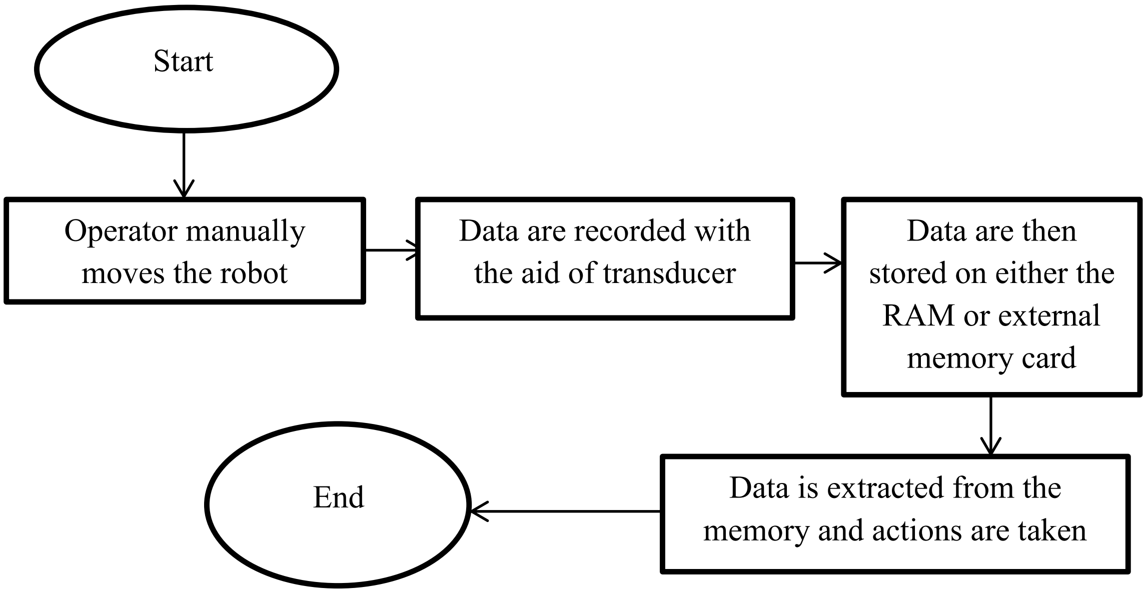


Figure 2.3: Flow Chart of the Data Record Process

### 2.3.1 Lead-Through Programming Problems

There are four known problems with the lead through method which are (1) affordability, (2) intuitiveness and teaching accuracy of the teach-pendant interface as a human machine interface (HMI), (3) feasibility of the on-line programming due to the great number of the teaching points and (4) the confidentiality and intellectuality of the sensor-less systems especially when path precision is taken into consideration.

The first problem can be described in terms of changing the robotic arms' location and orientation. It is desirable to move the robotic arm's tool frame rather than moving the space frame itself, for such changes in locations and orientations maneuvering robots using a keypad of joystick on the pendant is not easy and affordable to all operators, as it requires a non-deniable amount of skills and experiences [9].

The second problem is regarding the teach-pendant which is one of the most common ways for programming a robot as well as a common human machine interface (HMI) [10]. Yet to program a robot using a teach pendant, the operator should set up the robot's jogging conditions, frame and motion mode, only then he can use the joystick of the teach pendant to move the robot [6]. In comparison with the off-line programming methods, programming a robot with a teach pendant does not need a PC, which is an advantage in terms of cost. Yet a teach pendant programming method is not intuitive and has a low teaching accuracy which requires rounds and rounds of trails and errors, hence it is a time consuming and requires a certain level of robotics knowledge to deal with a teach pendant [10].

On the other hand, the third problem is based on a robotic machining perspective, where there are two types of machining processes whose motion are governed by complex work-piece [11]. Cleaning and deburring machines are typically the first type, which have a very complex 3D curved surface path, crucial cycle time requirements and relatively low surface accuracy. Most of the deburring operations are done manually in extremely noisy, dusty and unhealthy environmental conditions, therefore an automation for these operations is highly desirable [11]. On the other hand, milling machines are the second type of machining processes in which robots move in a simpler path with a lower feeding speed (20-30 mm/s) [11]. One of the most difficulties these machines encounter is generation of the robot motion. Despite the fact that teach pendant is the most carried out conventional method to fulfill a robot on-line programming, it is not feasible for machining processes especially for deburring processes as it has a great number of teaching points and high accuracy is needed for positioning purposes [11]. Moreover, offline programming method, which extracts the CAD data of the work-piece, is more accurate and flexible but it is cost-effective for large batch sizes and still requires additional calibration procedures for higher accuracy demands [11].

Finally yet importantly, the fourth problem is involved in robots that have direct contact with objects they manipulate are called robot force control. With the force control, robots gain one more step towards human nature (feeling or touching). Trajectory generation by the lead-through teaching for force control robots is quite time-consuming process if path precision is considered [12].

According to [12], methods of programming robot paths can be categorized as CAD based and non-CAD based method. CAD-based system is a method where the operator specifies the geometrical entities such as the surface or the edge of the geometry from a CAD model, and then the system will automatically simulate and generate the path in the virtual world. Despite the beneficial features of the CAD drawings, in reality they are neither confidential nor intellectual especially in the foundry industry [12]. As robots in certain situations have to be able to effectively capture the geometrical information of the area or the object they are acting on.

### 2.3.2 Available Solutions

Based on the previously listed problems, there are four relevant solutions to these problems respectively. (i) Is the usage of the ISD (industrial steering device) which is known as the jogging mouse, (ii) is a 6 DOF (degree of freedom) wire-based programming device, (iii) is an effective teaching method referred as programming by demonstrating (PbD) and (iv) is the addition of a sensing system to the robotic hand.

The first solution was proposed in [9], a commercially available 6 degrees-of-freedom steering device ISD (Industrial Steering Device) from space control was used on a welding arm. For the purpose of accessibility the mouse jogging device is attachable to various locations on the robot. Both the mouse jog and the robot axes were calibrated so that the operator could not jog the robot if the mouse was re-mounted from one location to another without a calibration done on the device. After the mouse jogging device was mounted and calibrated, as shown in Figure 2.4, a graphical user interface (GUI) was used to aid the process of the lead through teaching method, as shown in Figure 2.5.

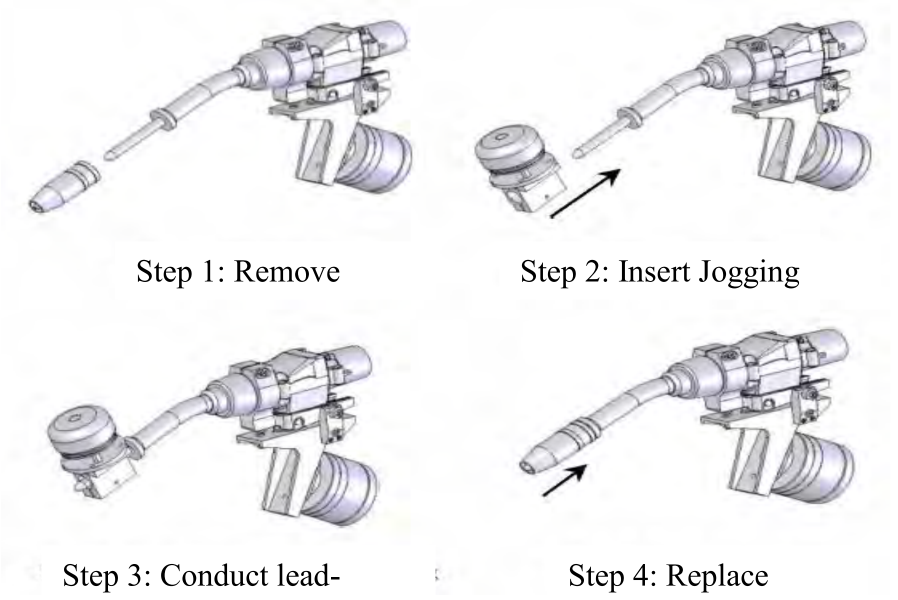


Figure 2.4: Four Major Operational Sequences for the Lead-Through Teaching [9]

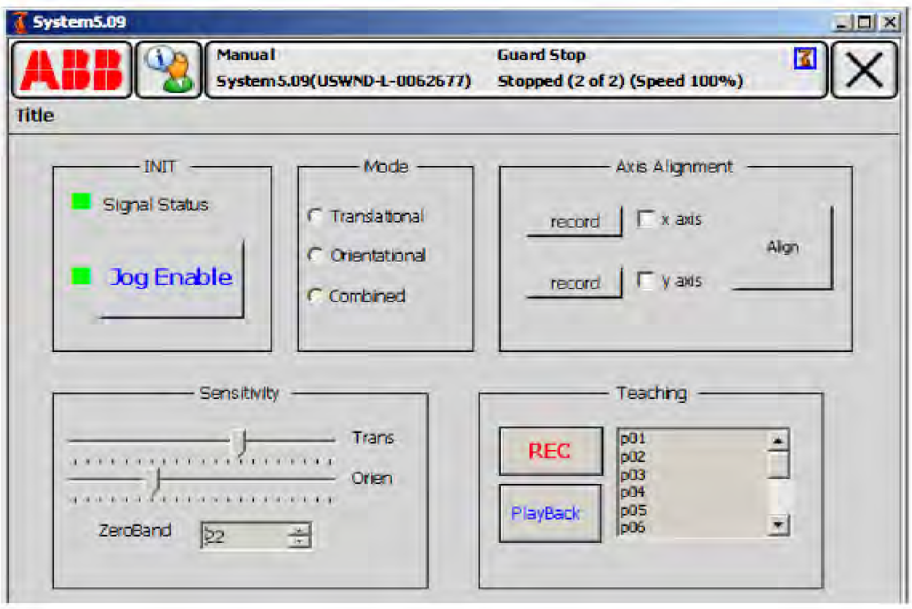


Figure 2.5: Graphical User Interface on Teaching Pendant to Assist Jogging [9]

The second solution was proposed in [10] for the second problem. Even though a 6-DOF mouse is an intuitive technology and demands low physical efforts, it was not a simple solution since the calibration between the 6-DOF mouse and robot coordinate system was required. Consequently, a new device to program a robot was introduced, a