TECHNICAL UNIVERSITY OF KOŠICE

FACULTY OF MANUFACTURING TECHNOLOGIES WITH A SEAT IN PREŠOV

Real-time robot programming system for production applications

Bachelor thesis

2021 Liubomyr Kulka

TECHNICAL UNIVERSITY OF KOŠICE

FACULTY OF MANUFACTURING TECHNOLOGIES WITH A SEAT IN PREŠOV

Real-time robot programming system for production applications

BACHELOR THESIS

Study programme:	Mechanical Engineering
Study specialization:	Automotive Production Technologies
Department:	Department of Automobile and Manufacturing Technologies
Supervisor:	prof. Ing. Sergej Hloch, PhD
Consultant:	

Prešov 2021

Liubomyr Kulka

Abstract in English

Robotics is used in almost all areas of human activity. Robots play an important role in production, in particular in the automotive, aerospace, food industries, medicine and so on. To date, the number and variety of products is very wide, so the production process must be very flexible and have the ability to adapt to new conditions, technological solutions. The trend towards mass production of products to individual orders requires a high degree of flexibility and automation of all processes in the production chain. The level of complexity of robots is constantly increasing, it requires a higher level of skills of employees. Enterprises refuse to cooperate with workers who do not have enough knowledge and experience to work with industrial robots. To set up an industrial robot for stable, accurate and efficient operation, it takes a lot of time, a high level of skill of the worker, knowledge of programming. This problem causes a lot of research in this area and encourages the creation of new methods of programming industrial robots. Research and the creation of new programming methods will help increase human - robots interaction to a more intuitive level.

Keywords

Industrial robot, robot manipulator, Arduino platform, gyroscope, accelerometer, 3D visualization.

Abstract in Slovak

Robotika sa používa takmer vo všetkých oblastiach ľudskej činnosti. Roboty hrajú dôležitú úlohu vo výrobe, najmä v automobilovom, leteckom a kozmickom priemysle, potravinárskom priemysle, medicíne atď. K dnešnému dňu je počet a rozmanitosť výrobkov veľmi široká, takže výrobný proces musí byť veľmi flexibilný a musí byť schopný prispôsobiť sa novým podmienkam, technologickým riešeniam. Trend hromadnej výroby výrobkov k jednotlivým objednávkam si vyžaduje vysoký stupeň flexibility a automatizácie všetkých procesov vo výrobnom reťazci. Úroveň zložitosti robotov sa neustále zvyšuje, vyžaduje si vyššiu úroveň zručností zamestnancov. Podniky odmietajú spolupracovať s pracovníkmi, ktorí nemajú dostatok vedomostí a skúseností na prácu s priemyselnými robotmi. Nastaviť priemyselného robota na stabilnú, presnú a efektívnu prevádzku vyžaduje veľa času, vysokú úroveň zručností pracovníka, znalosť programovania. Tento problém spôsobuje veľa výskumov v tejto oblasti a podporuje vytváranie nových metód programovania priemyselných robotov. Výskum a vytváranie nových programovacích metód pomôže zvýšiť interakciu človeka a priemyslu na intuitívnejšiu úroveň.

Kľúčové slova

Priemyselný robot, robotický manipulátor, platforma Arduino, gyroskop, akcelerometer, 3D vizualizácia.

63684

TECHNICKÁ UNIVERZITA V KOŠICIACH

FAKULTA VÝROBNÝCH TECHNOLÓGIÍ

Katedra automobilových a výrobných technológií

ZADANIE BAKALÁRSKEJ PRÁCE

Študijný odbor: **Strojárstvo**

Študijný program: **Technológie automobilovej výroby**

Názov práce:

Programovací systém robotov v reálnom čase pre výrobné aplikácie

Real time robot programming system for manufacturing application

Študent: **Liubomyr Kulka**

Školiteľ: **prof. Ing. Sergej Hloch, PhD.**

Školiace pracovisko: **Katedra automobilových a výrobných technológií**

Konzultant práce:

Pracovisko konzultanta:

Pokyny na vypracovanie bakalárskej práce:

Úvod
1. Analýza súčasného stavu riešenej problematiky
2. Definovanie nedostatkov a návrh riešenia problému
3. Experimentálna metodika a experimentálne overenie
4. Výsledky a diskusia
5. Nový prístup k riešeným problémom

Záver

Jazyk, v ktorom sa práca vypracuje: anglický

Termín pre odovzdanie práce: 21.05.2021

Dátum zadania bakalárskej práce: 30.10.2020

TECHNICKÁ UNIVERZITA V KOŠICIACH FAKULTA VÝROBNÝCH TECHNOLÓGIÍ -03-

Dr.h.c. prof. Ing. Jozef Zajac, CSc.

dekan fakulty

Declaration

I hereby declare that this thesis is my own work and effort. Where others sources of information have been used, they have been acknowledged.

Prešov, 19. May 2021

....

Signature

Gratitude

I would like to express my sincere gratitude to all those who in some way helped me and supported me in creating this bachelor's thesis. I especially want to thank the supervisor of the bachelor's dissertation prof. Ing. Sergej Hloch, PhD. As well as the control commission, which checks my bachelor's thesis. Thank you very much for your help.

Content

List of Figures

List of Tables

Introduction

Industrial robots plays a very important role in all industries - from food to aerospace. The main advantage of robots is that they can perform accurate and error-free cyclic work for a long time, which significantly reduces the number of defects in production. However, before starting work, employees have an important task - to program the robot so that its work was as fast as possible, efficient and most importantly safe for humans. The process of programming, adjusting and calibrating the robot must take place as soon as possible so that the robot can quickly begin its task, after installation on the production line. Fast programming requires experienced workers who will not make mistakes during setup, because every mistake is a waste of time and money, even now the process of programming robots is quite expensive, and sometimes is necessary to change the program because the product is constantly changing. Today, many methods of interaction with the robot are being developed, based on augmented reality, voice commands, human movements and gestures. An important detail, in addition to the skills of the employee, is the interface of robot-human interaction, it should be easy to learn and multifunctional, as well as be adapted to modify, add new methods and functions. Therefore, the development of such effective methods is given much attention in our time.

1 Overview of the current state

The general principles of robot-human interaction are described in papers [1], [2] and [3], the type of robot determines its ability to interact with humans, some types of robots are designed to automate cyclical production processes, while another type is designed to interact with workers through robot-human interactions (HRI). defined in DIN EN ISO 9241-110.

Today, the question of combining safety and speed of interaction between the robot and the worker is acute, this issue is considered in papers [4] and [5]. The robot, integrated into the human work area, must be equipped with reliable safety measures for safe cooperation and in case of danger completely stop functioning.

Robotics makes extensive use of augmented and mixed reality, shown in papers [6] and [7], it is a convenient tool for controlling industrial robots, intuitive for the inexperienced worker, the control interface is easy to understand and does not require large investments. Article [8] demonstrates a method of programming robots using a telephone based on augmented reality, this development provides a simplified programming process using a simple graphical interface, no physical contact with the robot, flexibility for modifications to the robot and workspace [8].

The proposed method in paper [9] – allows to control the supplemented 3D-model of the robot using an infrared tracker. The movements of the tracker, which a person holds in his hand, are registered by infrared sensors, which are placed around the perimeter. However, the company may have difficulty using this method, as there may be other sources of infrared light that will interfere with the normal operation of the tracker, so this complex must be isolated.

The use of augmented and virtual reality as an environment for the development, testing and implementation of new production systems, as well as for its further optimization and to create an environment for staff training is described in documents [10] and [11]. Virtual reality allows to model the production unit, namely man and collaborative robot, and identify system deficiencies at the design stage, in the future it can save a lot of resources.

A new direction in programming and control of robots is control by means of voice commands. The interface of human interaction and robot can be based on different sensors. In papers [12] and [13] the possibility of interaction of the robot and the person by means of voice commands is considered. Voice recognition is a resource-intensive process, so it requires significant power, so in paper [12], the option of voice processing using cloud technologies is considered. Article [13] demonstrates an experiment in which an inexperienced worker interacts with the graphical interface of robot programming, using, mouse and keyboard, as well as interaction using voice commands based on web frameworks, which are installed in almost all modern web browsers, which allows to recognize pronunciation with high accuracy and speed. Research has shown that interaction based on voice commands is very intuitive and reduces robot programming time by 46%. However, the issue of background noise in large enterprises remains open, which can affect the efficiency and safety of robot-human interaction based on voice commands.

In papers [14] and [15] the method of control of an industrial robot by means of human gestures is described and shown. Human movements and gestures are recognized based on the gesture library the interpreted data is transmitted to the robot microcontroller. A set of RGB-d and monochromatic IR cameras are used to collect data. The gestures of the hands and body are able to control the movement of the robot in six directions. This technology is useful where accurate positioning of the robot is not required. An important aspect is the accuracy of recognizing the movements themselves with the help of artificial intelligence.

Mobile platforms with a robotic arm are beginning to be used more and more often in enterprises, paper [16] considers the technology of a mobile robotic platform, and the synchronous operation of a robot and a human. This robot is designed to support the employee - the operator and the mutual performance of the task, these devices can be made to transfer multiple parts from one place to another. With the help of laser scanners, the platform can create a map of the environment for orientation, as well as bypass obstacles that may stand in the way of the platform.

1.1 The main contribution

The main contribution of this work is a description of the implementation of the method of control of production robots, collaborative robots, as well as other types of manipulators using a gyroscope and accelerometer. We will consider in detail the hardware that will be required for project implementation, as well as consider the firmware code of the microcontroller, the connection of all components of the circuit. In addition, the contribution of this article is to show in which areas this method can be used, where it may be slightly more effective than others, and where its use is undesirable. Show and describe a simplified graphical interface with a 3D model of the manipulator, which can serve as a virtual environment to create a trajectory of the robot. The purpose of this work is not to point out the shortcomings of other methods, but only to expand the existing tools of human-robot interaction, which, in turn, can lead to faster production and deeper integration of the worker into the working space of the robot.

2 Development of inertial measuring unit

The IMU inertial measuring device is an electronic device that measures and outputs the values of the body's velocity, its orientation, and the gravitational forces with which the body moves, and sometimes the magnetic field that surrounds it. IMUs are commonly used in maneuvering aircraft, including unmanned aerial vehicles, and spacecraft, satellites and landing modules. Recent developments allow the manufacture of GPS modules with IMU support [17]. The IMU allows the GPS receiver to operate in conditions where GPS signals are not available, such as in tunnels, in the middle of buildings, or in electronic interference [18].

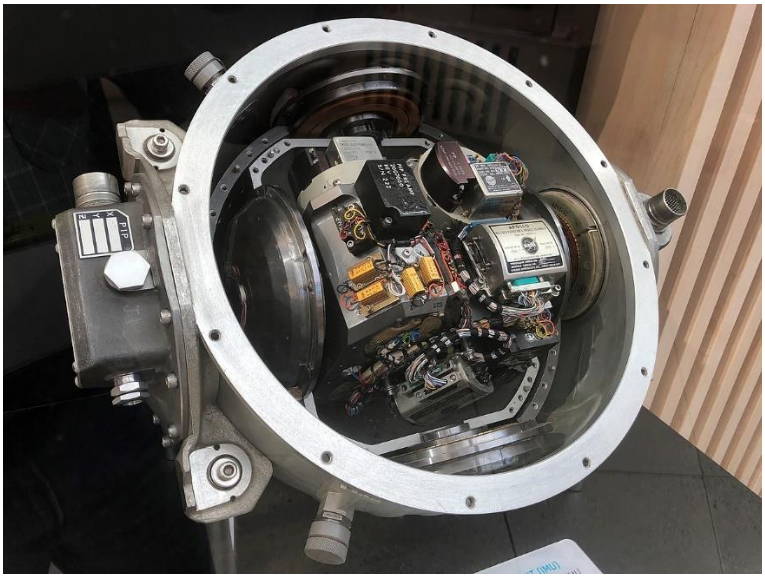


Fig. 1 Inertial measuring device of the Apollo spacecraft [19]

Recent advances in microelectromechanical systems (MEMS) have made it possible to create small and light inertial navigation systems. These advances have expanded the range of possible applications to include areas such as the capture of human and animal movements [20]. IMU are a major component of inertial navigation systems used in surface vehicles and guided missiles, among others. As such, the data collected from the IMU sensors allows the computer to track the position of the device using the method of counting coordinates. An inertial measuring device for measuring the current acceleration speed uses one or more accelerometers, and shows changes in rotation parameters such as pitch, roll and yaw using one or more gyroscopes. Some devices also have a magnetometer to be able to calibrate the drift orientation.

Inertial navigation systems were originally developed for missiles [20]. American rocket pioneer Robert Goddard experimented with elementary gyroscopic systems. These systems have become more widely used with the advent of spacecraft, guided missiles and commercial airliners. Early German World War II V2 [21] guidance systems combined two gyroscopes and a side accelerometer with a simple analog computer to adjust the azimuth for a rocket in flight. Analog computer signals were used to power four graphite rudders in the exhaust of a missile to control flight [20]. The Gn & C (guidance, navigation, and control) system for V2 has provided many innovations as an integrated platform with short-circuit guidance [20].

2.1 Laser gyroscopes

Laser gyroscope is an optical device for measuring angular velocity, usually used in inertial navigation systems. Laser gyroscopes use the Sagnac effect - the appearance of a phase shift of counterpropagating light waves in a rotating ring interferometer. Unlike a mechanical gyroscope, this device does not seek to maintain the initial direction, but measures the angle of rotation of the device in the plane of the resonator contour. By counting the number (or phase for small angles) and the direction of the antinodes of the standing wave passing through the photodetector platforms, which is stationary in the inertial reference frame, it is possible to get the value of the angle by which the turn is made, and by differentiating in time to obtain the angular velocity. The advantages of this gyroscope are digital output signal, short readiness time, no moving parts[22] [23].

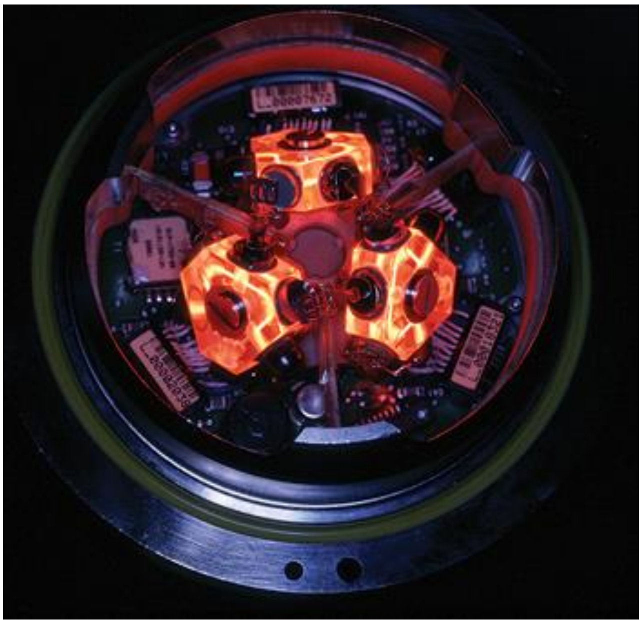


Fig. 2 Laser gyroscopes used in modern aircraft for navigation [23]

3 Preparation for project implementation

This section will discuss the hardware and software for the implementation of the method of robot control. Microcontrollers, development environments, communication protocols.

3.1 Hardware part

An Arduino platform was chosen to implement the method of controlling an industrial robot using gyroscope and accelerometer indicators. This platform is easy to learn and does not require deep knowledge in programming and electronics.

Arduino is an opensource platform that is used to create projects of varying complexity including a large number of sensors and modules, stepper motors and so on. This project will use the Arduino NANO board [24]. An important advantage of the Arduino platform is that its programming does not require a special device - a programmer, there is an opportunity flash the microcontroller with one click of the mouse and the USB cable, which connects directly to the microcontroller. Arduino-based device designs can work independently or interact with software on a computer (e.g .: Flash, Processing, MaxMSP).

Processor characteristics ATmega328:

Number of processor pins	32
Flash memory	32 KB
EEPROM memory	1 KB
SRAM memory	2 KB
ADC channels	6
PWM channels	6
Serial interface	x1
I2C interface	Yes

Tab. 1 Processor characteristics [24]

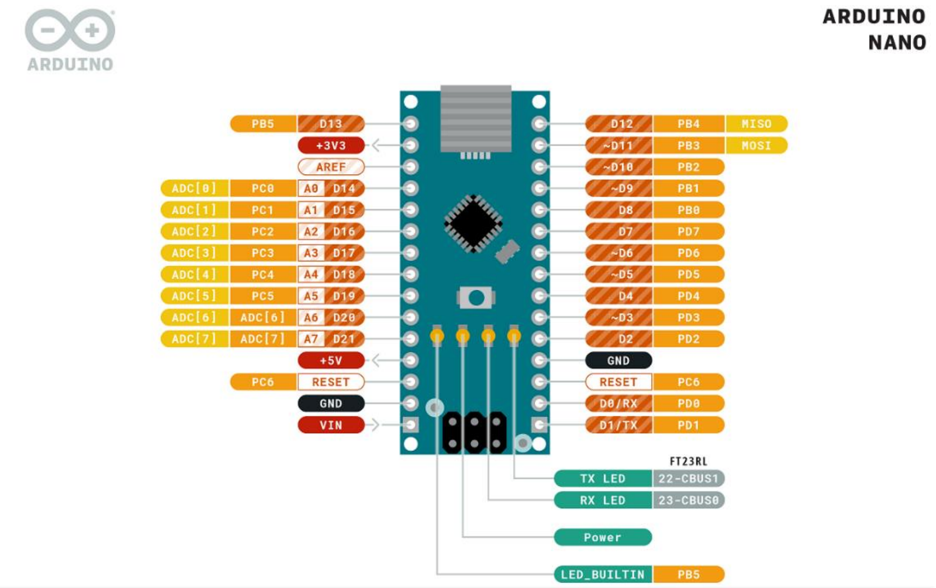


Fig. 3 Pinout of the Arduino NANO microcontroller [24]

3.2 Power supply board

The type of power source of the microcontroller is an important aspect in the development, the method of power supply directly affects the quality of the processor, as well as the sensors whose signals it receives.

The supply of voltage to the terminals Vin and GND is a universal way, through these pins the power supply is supplied to the onboard voltage stabilizer AMS1117-5.0. This is a linear stabilizer that has its pros and cons. This allows to power Arduino and Arduino projects from a voltage of 7-12 volts. The stabilizer is designed so that it gives a good equal voltage with minimal vibration, but converts all excess voltage into heat.

At 12 Volts on Vin it is possible to receive from pin 5 V no more than 500 mA without risk of an overheat of the stabilizer. Power supply on Vin output is possible only if the Arduino project does not use powerful energy consumers [25].

3.3 Acceleration and tilt sensor

An inertial measuring device (IMU) is an electronic device that measures the angular velocity and acceleration acting on a sensor (gravity) using a combination of an accelerometer and a gyroscope, sometimes also magnetometers.

A GY-85 sensor was chosen for this project [26]. GY-85 contains three microcontrollers that measure acceleration, orientation using a gyroscope and a magnetometer. Communication with the sensor using the I2C data transmission protocol. The X axis and the Y axis are horizontal, the Z axis is vertical. The rotation around the axes is named after the main axes of the aircraft.

There are three axes of rotation "pitch", "roll", "yaw" that correspond to the axes X Y and Z.

The accelerometer used on the GY-85 is the ADXL345. The chip makes it possible to measure acceleration in three planes with a resolution of 13 bits, which allows you to get a very high measurement accuracy and can measure in the range of ± 2 g, ± 4 g, ± 8 g, ± 16 g [27]. ADXL345 also has some additional functions, such as touch detection.

Despite the fact that it has a fairly good calibration, it is possible to calculate new values for calibration of sensors and save them in the chip. They will automatically apply to all future measurements. The most common way to calibrate the accelerometer is to perform a series of measurements (preferably from 10 to 100) in standby mode and calculate the average value.

Typically, the chip returns the digitized values of the sensor with a resolution of 10 bits. To continue working with this data, the values must be converted to a common unit of acceleration.

All digitized data collected from the three sensors is processed by a separate DMP processor (any Arduino board can also be used).

Fig. 4 Sensor GY – 85 [29]

The accelerometer and gyroscope do not give the actual value of acceleration or angular velocity, because their signal passes through an analog-to-digital converter, which gives values such as 65 536. Suppose the accelerometer now operates in the measuring range $FS = \pm 2g$, the full range of possible values will be $2 * FS = 4g$. The corresponding voltage values are digitized by a 16-bit ADC, which can break the entire interval by a maximum of $2^{16} = 65536\, units$. The minimum increment that can be detected is precisely the unit of measurand $LSB = 2 * 65\,536$. It is necessary to remember, that the account is kept from null, so in fact the maximum measured value will be $2 * FS\_true \; = (2^{16} - 1) * LSB$. That is, the more bits in the ADC or DAC, the smaller the discrepancy.

The ITG3205 gyroscope has a digital output and determines the angular velocity along its three axes [28]. The module has 3 built-in 16-bit analog-to-digital converters - one converter on each axis, so that the data rate can reach 400 kHz.

The data is transmitted via the I2C or SPI interface and is represented as a 16-bit number. It has a high resolution, so the accuracy of the change in inclination is determined with an accuracy of 1 degree or less.

The HMC5883L magnetometer (or electronic compass) has a digital interface. The analog-to-digital converter has 12 bits. The measurement error is only 1-2 °.

The GY-85 should be placed at least 10 cm away from metal objects to avoid interference.

3.4 Sensor connection

The sensor GY - 85 is connected by four pins, 2 to power 5 volts and minus ground, and two for communication via the I2C data transmission protocol. The Arduino microcontroller and the sensor will be mounted on the breadboard and will be connected by wires [29].

GY - 85	Arduino
VCC_IN	+5V
GND	GND
SCL	A5
SDA	A4

Tab. 2 Connection: Arduino - GY – 85 [29]

3.5 Program part

Consider the work environment in which the work will be conducted - Arduino IDE [30]. The IDE stands for Integrated Development Environment, there is a notebook in which will be written code, a preprocessor and a compiler that check and compile the code, and tools that load the code in the chosen way. IDE written in java. In this environment, there are tools for easy work with COM ports of the computer, there is also a plotter for graphs, this is a very useful tool. The software will be written in C Programming, the standard programming language for microcontrollers.

3.6 Communication protocol

The I2C serial communication protocol (also called I2C - Inter-Integrated Circuits) uses two bidirectional communication lines called SDA (Serial Data) and SCL (Serial Clock) bus for data transmission [31]. There are also two power lines. The SDA and SCL buses are pulled to the power bus through resistors.

There is at least one master connected to the network, which initializes the data transfer and generates synchronization signals. The network also has slave devices that transmit data at the request of the master. Each known device has a unique address to which the master addresses.

The address of the device is specified in the passport (datasheet). Up to 127 devices, including several master devices, can be connected to one I2C bus. It is possible connect devices to the bus during operation, it supports "hot plug".

Data transmission is initiated by the master device, for this it begins to generate 9 clock pulses and sends them to the SCL line [32]. At the same time on the SDA data line, it sends the address of the device with which the task is to establish communication, clocked with the first 7 clock pulses (limit in the address range: 2^7 = 128 minus the zero address). The next bit of the parcel is the operation code (read or write), and the other bit is the acknowledgment bit (ACK) that the slave device has received the request. If the confirmation bit is not received, the exchange ends. Or the master device continues to send repeat requests. To facilitate communication with I2C devices for Arduino, a standard Wire library was written [33].

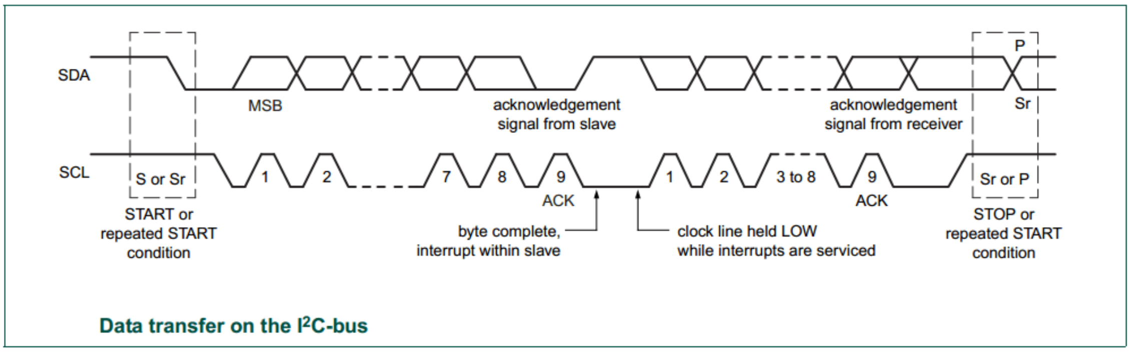


Fig. 5 The principle of operation of the I2C protocol [32]

4 The principle of operation of the MEMS accelerometer and gyroscope

4.1 MEMS technology

MEMS are micromechanical systems consisting of micro components ranging in size from 0.001 mm to 0.1 mm. These components are made of silicon, polymers, or metals [36]. Also, these devices usually work in conjunction with processors (microcontroller) to improve system performance [34].

MEMS (microelectromechanical systems) technology is based on production technology developed for silicon integrated circuits. Micromechanical structures are made by etching on a silicon substrate with the formation of sensor elements or micromechanical drives that can move. At present, accelerometers and gyroscopes based on MEMS technology have been used for more than 15 years in the car security system, including airbags, rollover detection and car alarm systems.

MEMS accelerometers are also used to detect movement in various devices, such as joysticks for video games, in mobile phones. MEMS micromirror optical starters are used in projectors, high-definition TVs.

4.2 MEMS – accelerometer

The operation of the accelerometer can be based on the capacitor principle. The moving part of this system is a small load, which is displaced depending on the acceleration of the device. The main element of the accelerometer based on MEMS technology is the design of a movable platform consisting of two independent plates, one of which is fixed relative to the base, and the other is attached to the elastic elements. When the sensor is accelerated, the springs expand and compress. This acceleration changes the charge capacity between the plates by changing the distance between them, based on this, the actual acceleration is calculated. Based on these data, it is possible to obtain the movement of weight and at the same time the desired acceleration.

MEMS structures are usually formed of monocrystalline silicon or polysilicon, which is applied at very high temperatures to the surface of the monocrystalline silicon wafer. The dimensions of MEMS structures are measured in micrometers and also require very high accuracy of silicon photolithography and etching technology. Sensors can be manufactured to measure acceleration from one g to hundreds of g with a bandwidth of up to 20 kHz. With this flexible technology, it is possible to create designs with very different mechanical characteristics. One mechanical parameter that can be controlled and changed is the spring stiffness. The weight of the sensor element and the damping of the structure can also be changed according to the project. [35][36]

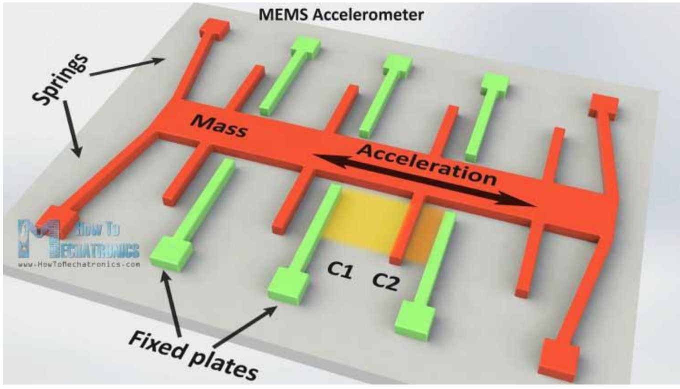


Fig. 6 Accelerometer MEMS technology [36]

4.3 MEMS – gyroscope

The gyroscope measures the angular velocity using the Coriolis effect. Like an accelerometer, this offset will change the capacity, which will be measured, processed, and correspond to a certain angular velocity [37].

The gyroscope consists of three independent uniaxial vibration sensors of angular velocity (MEMS gyroscopes), which respond to rotation around the X - axes, Y - axes, Z - axes. Two suspended masses oscillate on opposite axes. With the advent of angular velocity, the Coriolis effect causes a change in the direction of vibration, which is recorded by the charge capacity sensor. The differential capacity of the component of the proportional displacement angle is measured. The resulting signal is amplified, demodulated and filtered, resulting in a voltage proportional to the angular velocity. This signal is digitized using a built-in 16-bit ADC. Digitization speeds can vary programmatically from 3.9 to 8000 samples per second, and user-programmed low-pass filters (LPF) provide a wide range of possible cutoff frequencies. LPF is required, including to remove vibrations from the motors (usually above 20-25 Hz) [37].

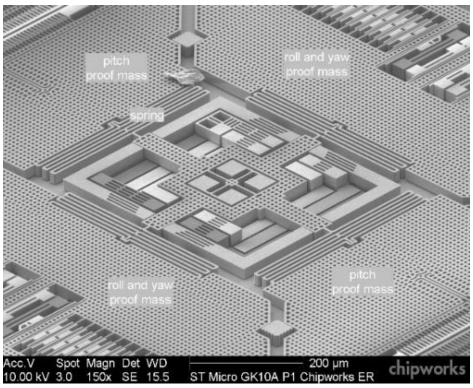


Fig. 7 View of gyroscope MEMS technology [38]

5 The principle of operation of the sensor GY – 85

5.1 Microprocessor registers

One of the most important skills required when working with microprocessors is the ability to interact with registers. This is an integral part of programming microcontrollers and embedded systems.

In general, a register is a special type of memory inside a microcontroller that is used to control the processor and peripherals. Each register is a memory cell with a length of 2^n bits, where each bit can be represented as a microswitch with which to control a parameter of the microcontroller.

Each of the registers has its own serial number - address. The address of the register is indicated by a 32-bit number specified in the hexadecimal number system. By writing to the register address a certain combination of ones and zeros, which are usually represented in hexadecimal form, it is possible to configure and control a node in the processor. In general, it should be noted that there are two types of registers: general purpose registers and special registers. The first is inside the processor core, and the second is part of RAM. For each processor there are special manuals (datasheet), which contain the so-called register map, from which the user can learn the purpose, control, and location of a register [39].

Therefore, usually the structure of the register is described in the form of a small table with pointers:

1. The name of the register and the description of its purpose
2. The address of the register or its offset relative to the base address
3. Default value after reset
4. Type of register access (read, write, read / write)
5. Values and descriptions of the parameters of the recorded bits

5.2 Sensor registers GY – 85

In total, the sensor has 57 registers, of which it is possible to access 32, which are used to configure and control the microprocessor.

The information sheet of the sensor indicates the so-called register map, for example, consider the register 0x31, which consists of 8 bits.

The DATA_FORMAT registry controls the submission of data to register 0x32 through Register 0x37. All data except data for the ± 16 g range [39].

1. Bits D0 and D1 change the range of sensor acceleration measurements ± 2g; ± 4g; ± 8g; ± 16g.
2. Bit D2 determines the bit transfer parameter, from the youngest to the oldest, or vice versa, if the bit is set to 1.
3. Bit D3 determines the measurement mode. When this bit is set to 1, the device has a full measurement resolution, the output resolution increases with increasing range g. If the bit is set to 0, the device is in 10-bit measurement mode.
4. Bit D5 sets the interrupt, 0 activates the interrupt, 1 sets the interrupt inactive.
5. Bit D6 adjusts the SPI data bus.
6. Bit D7 performs self-tuning of the sensor, excluding measurement offset.

6 Start of implementation

This section will describe the beginning of the project, the connection of all necessary electronic components. Basic program code for the microcontroller, visualization of the received data.

6.1 Description of the general principle of robot control

The robot will be controlled by an accelerometer and gyroscope. The output data from the sensor will be linked directly to the main tool of the industrial robot, such as a welding machine, a gripper, and so on, which means that it will not be possible to control each joint of the robot. The control will be carried out exclusively by rotating the sensor in three axes.

The angle of inclination will be determined directly from the gyroscope, as well as from the accelerometer using trigonometric functions. This is necessary in order to reduce the measurement error of the angle, as well as to cancellation the so-called zero drift. The data from the gyroscope is unstable and constantly drifting. Next will be described a way to solve this problem.

There are 3 modes of robot control.

1. The movement of the main working tool on the X and Y axes
2. The movement of the main working tool on the Z axis
3. Rotation of the main working tool on the X, Y and Z axis.

The organization of control in this way is due to the peculiarity of the accelerometer. Since the angle of inclination is determined by the accelerometer using the vector of gravity, it is impossible to determine the rotation around the Z axis, as this axis is always parallel to the vector of gravity. And also it is caused by the fact that it is impossible to provide movement of the working tool in three axes only by means of 2 axes of rotation.

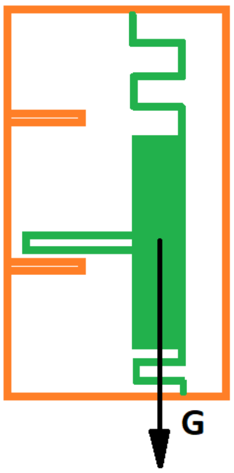


Fig. 8 Influence of free fall acceleration on the accelerometer sensor

6.2 Connection of all system components

In order for the structure to be reliable and for the connection between the components to be stable, a breadboard as well as connecting wires will be used. In this case, the Arduino microcontroller and the GY - 85 sensor will be securely fastened and will not shift from the starting position, because any movement will affect the measurement of the sensor. On the prototyping board are fixed tactile switches to adjust the control method, LED indicators and resistors.

Component name	Number
Arduino NANO	1
Sensor GY - 85	1
Tact Switch	2
Resistor 10 KiloOhm	2
Resistor 220 Ohm	2
LED indicators	2

Tab. 3 The number of required elements for connection

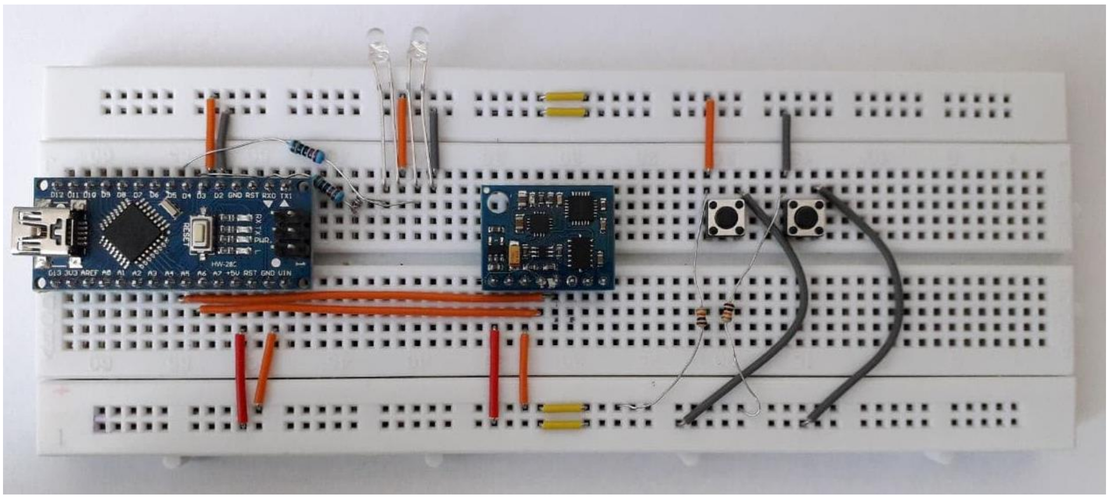


Fig. 9 The microcontroller, sensor and other components are fixed on the breadboard.

6.3 General structure of the program

Basically, all programs written on the Arduino platform consist of two basic parts, setup and a basic loop. The settings indicate the basic settings of the program, the initialization of communication with the serial port of the computer. In the part of the cycle, the main program is executed.

```
void setup()
{
   //TODO
}

void loop()
{
   //TODO
}
```

6.4 Software structure

First, all the necessary variables for the program will be declared. Then the adjustment unit will be performed, all used foams will be configured, the port for data transfer to the computer will be initialized, and also the sensor will be calibrated. The calibration algorithm will be described in a separate function that will rotate the offsets for the accelerometer and gyroscope.

In the main cycle of the program, the initial data will first be obtained from the sensor., then with the help of trigonometric functions and integration, they will be converted into real indicators of the tilt of the sensor. Then the data will be processed using a special filter that will increase the accuracy of the measurement and will be sent to the serial port of the computer. The program will also constantly monitor the status of the buttons to transmit the desired data.

6.5 Basic settings in the program code

In this section the writing of the software for the project will be gradually considered, the functions, variables will be described. To begin, is necessary to announce all the libraries and directives that will be used in this project.

```
#include "GY_85.h"
#include "math.h"
#include <Wire.h>
#define TIME_STEP  50
#define CALIB_ITER 100
#define CALIB_Z 15000
#define FK 0.3
```

The GY - 85 library contains all the necessary functions for working with the accelerometer and gyroscope, it also contains the functions of automatic calibration of the device.

The standard math.h library contains all the necessary mathematical, trigonometric operations, as well as constants such as the number pi, rounding operations, obtaining random numbers, and a set of macrofunctions. The Wire.h library is used to communicate the microcontroller and the device via the I2C protocol. There are also several directives that determine the frequency of sensor polling, the number of offsets calibration iterations, and the complementary filter coefficient.

```
GY_85 GY85;      //create the object
```

In this line the object of the sensor GY - 85 is initialized.

```
void setup()
{
    Wire.begin();
    delay(10);
    pinMode(4, OUTPUT);
    pinMode(5, OUTPUT);
    pinMode(3, INPUT);
    pinMode(2, INPUT);
    delay(10);
    Serial.begin(9600);
    delay(10);
    GY85.init();
    delay(10);
    calibration();
    /*
    Serial.print (calibration_acc_x_val);
    Serial.print (" ");
    Serial.print (calibration_acc_y_val);
    Serial.print (" ");
    Serial.print (calibration_gyr_x_val);
    Serial.print (" ");
    Serial.println (calibration_gyr_y_val);
    */
}
```

This configuration function contains commands to initialize the wired library, initialize the serial connection to the computer via the COM port, the argument of this function contains the value 9600 - bit rate per second, initialization of the sensor. Each command is divided by a delay function of 10 milliseconds.

The pinMode() function adjusts the foam of the microcontroller. The tactile switches are connected to 2 and 3 pins, the INPUT argument means that 5 volts can be applied to the pin, this will mean that the button is pressed. Two LEDs are connected to digital pins 4 and 5, the OUTPUT argument means that the pin works as a voltage output and can supply 5 volts to light up the LED.

The calibration() function starts the algorithm for setting new offsets for more accurate measurement. It is possible to display the values of new offsets using the function Serial.print ().

```
for(int i = 0; i < CALIB_ITER; i++)
{
 calibration_values[i] = GY85.accelerometer_x(GY85.readFromAccelerometer());
}
for(int i = 0; i < CALIB_ITER - 1; i++)
{
calibration_values[i + 1] = calibration_values[i] + calibration_values[i + 1];
}
calibration_acc_x_val = calibration_values[CALIB_ITER - 1] / CALIB_ITER;
```

The calibration algorithm works as follows:

1. Receiving data from the sensor CALIB_ITER times;
2. Summing all the data;
3. Dividing the sum of all measurements by their number.

6.6 The main cycle of the program

In this part of the code, the sensor transmits data via the I2C communication protocol. The data is stored in variables with the appropriate names ax, ay, az and gx, gy, gz. Offsets are deducted from the received data. Due to the fact that in the normal position the Z axis is constantly parallel to the gravitational vector, the graph shows that the Z axis has a constant acceleration of 1g, Therefore, is required to subtract the number 255 from the variable az because the sensor provides data in 10-bit resolution.

```
float ax = GY85.accelerometer_x(GY85.readFromAccelerometer()) -
calibration_acc_x_val;
float ay = GY85.accelerometer_y(GY85.readFromAccelerometer()) -
calibration_acc_y_val;
float az = GY85.accelerometer_z(GY85.readFromAccelerometer()) - 255;

float gx = GY85.gyro_x(GY85.readGyro()) - calibration_gyr_x_val;
float gy = GY85.gyro_y(GY85.readGyro()) - calibration_gyr_y_val;
float gz = GY85.gyro_z(GY85.readGyro()) - offset_Z;
```

Having obtained the angular velocity data, it is necessary to find the angle of rotation by integration. Integration calculates the angle at which the sensor tilts over a short period of time, so it is necessary to constantly sum up the value of the previous measurement.

$$\varphi(t) = \int_{time\_s}^{time\_f} \omega(t) \quad (1)$$

In order to integrate the function as accurately as possible, it is necessary to know the difference between the current and previous measurement, the difference between these two values will give the exact measurement period. To do this, use millis() functions that indicate how much time has passed since the microcontroller was turned on. The values of these functions are stored in time_f and time_s variable. The difference between these two variables shows how long the program ran from the last integration to the beginning of the new one.

```
time_f = millis();
Angel_X = Angel_X + gx * (time_f - time_s) * 0.001;
Angel_Y = Angel_Y + gy * (time_f - time_s) * 0.001;
Angel_Z = Angel_Z + gz * (time_f - time_s) * 0.001;
time_s = millis();
```

6.7 Calculation of the angle using an accelerometer

Because the accelerometer has one negative effect, zero drift. This means that after starting the microcontroller and starting the sensor, the angle gradually increases and after 1 minute the measurement error will be 1-2 degrees, even if the sensor is constantly at rest and in a horizontal position. Therefore, to compensate for this effect, a complementary filter will be used, as well as accelerometer data, which will also be converted into an angle.

The next step is to calculate the angle with an accelerometer. Because the accelerometer, in addition to acceleration during movement, shows the acceleration of gravity. This fact can be used to obtain the angle of inclination. Because in the normal position the projection of gravity on the X axis is zero, and at an angle of 90 degrees the projection is 1 G. Using trigonometric operations, we can find the angle.

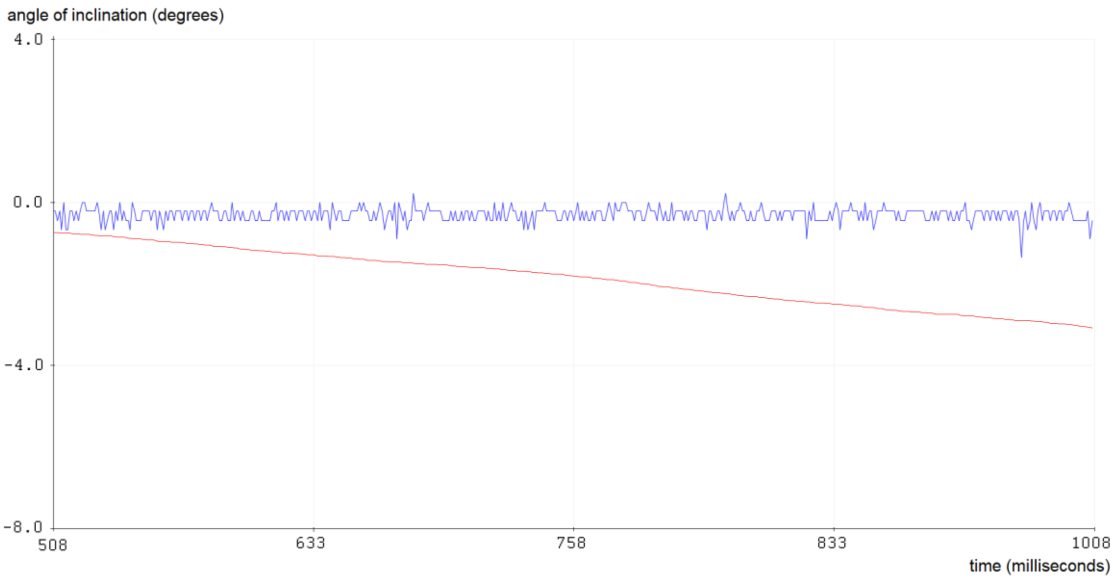
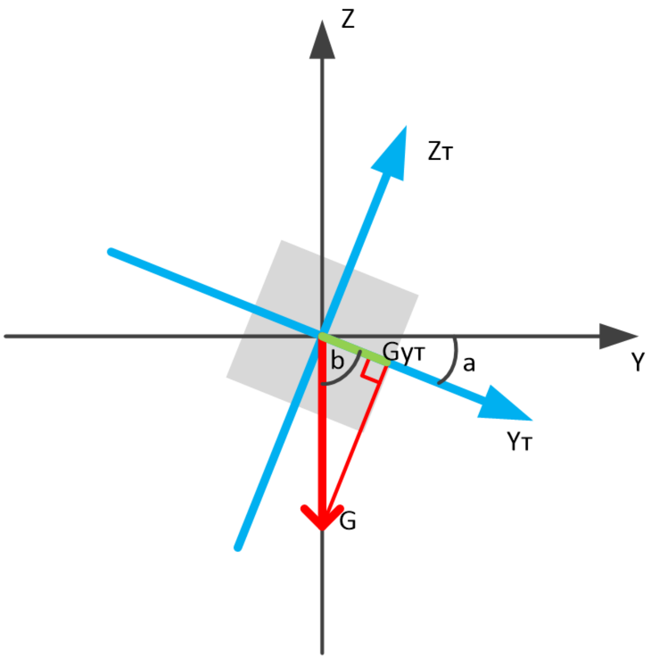


Fig. 10 Gradual change in the graph of the inclination of the accelerometer along the X axis, relative to the graph of acceleration.



This figure shows the principle of calculating the angle of inclination.

1. a - The angle of the body;
2. b - Deviation angle from the Z axis
3. G - Gravity vector
4. Gyt - Projection of gravity on the Y axis.

Fig. 11 Projection of gravity on the Y axis

```
angle_ax = 90 - acos(ay) * 180 / PI;
angle_ay = 90 - acos(ax) * 180 / PI;
```

These formulas calculate the angles of inclination using the projections of the vector of gravity on the X and Y axes.

6.8 Complementary filter

```
Angel_X_A = Angel_X_A * (1 - FK) + angle_ax * FK;
```

```
    Angel_Y_A = Angel_Y_A *(1 - FK) + angle_ay * FK;
```

This formula summarizes the instantaneous value of the accelerometer and the integral value of the gyroscope. The additional filter uses accelerometer data, the measurements of which are stable and do not drift, to equalize the value of the gyroscope, compensating for the drift of its value, as well as to smooth out discrete integration errors. It is also possible to adjust the force of correction by the coefficient of the filter K.

The coefficient K affects the operation of the filter. Basically, this value is adjusted manually depending on the environment in which the sensor is located. Also, the coefficient is adjusted depending on the accuracy of the sensor, the drift of its values. Too high a value of K will make the sensor too sensitive, so the measurement will be affected by slight vibrations. Too small a value of K may not be enough to fully compensate for the sensor drift.

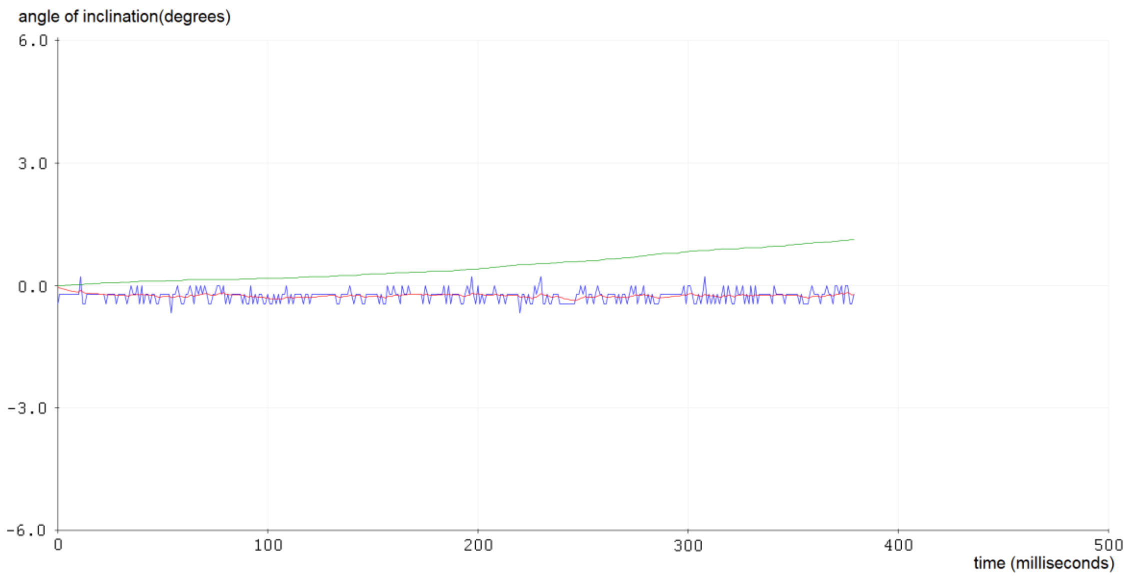


Fig. 12 Tilt angle using a complementary filter (Red)

This image shows 3 graphs. The blue graph shows the acceleration. The green graph shows the angle without the use of a complementary filter, so it is constantly increasing. The red graph shows the actual tilt angle using a complementary filter and since the sensor is in a horizontal position the graph is stable.

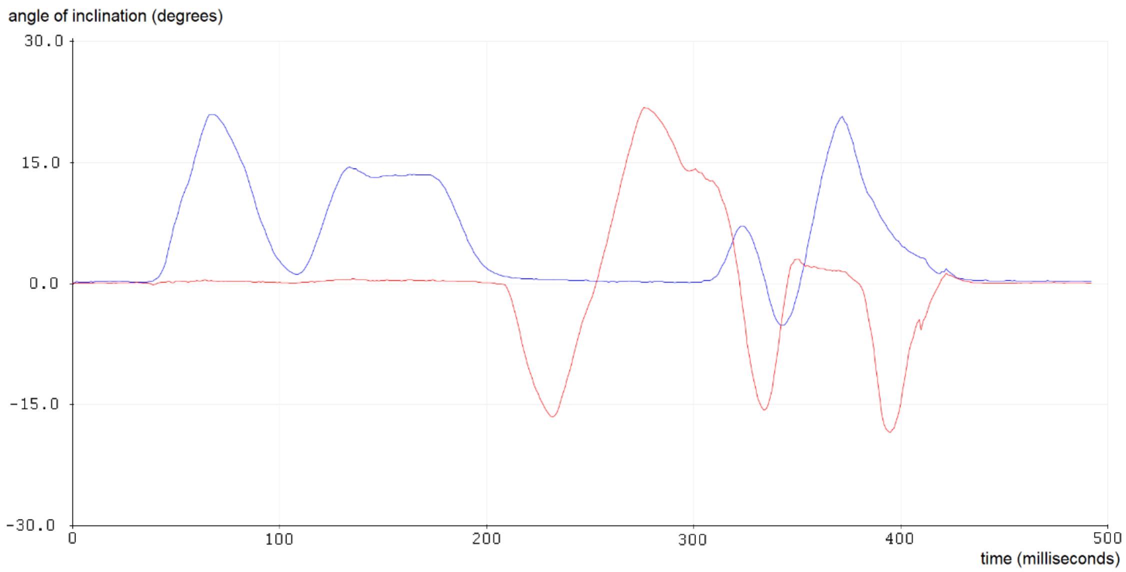


Fig. 13 Graph of the angle of inclination along the X and Y axes using a complementary filter

6.9 Button control

To control the operating modes will be used 2 tactile buttons as well as two LED indicators. A total of 3 control modes will be available Each tactile button can be in two states: true and false, so it is possible to make changes by pressing the keys. If both buttons are set to false, data transfer to the computer will be suspended, so this can be considered as turning off the sensor.

```
if(digitalRead(3) == true && flag1 == true)
{
  flag1 = false;
  button1_state = !button1_state;
}
if(digitalRead(2) == true && flag2 == true)
{
  flag2 = false;
  button2_state = !button2_state;
}
```

7 3d visualization

Visualization of the robot control process is an important aspect, as it allows constantly monitor the position of the robot, as well as facilitates interaction and makes it more intuitive. Software with visualization of 3D robot control was developed for this project. A simple interface has also been created that displays a variety of information, such as input data, coordinates of the main working tool, operating time, sensitivity coefficients and blind zone coefficients. This section will discuss the program code as well as the interface.

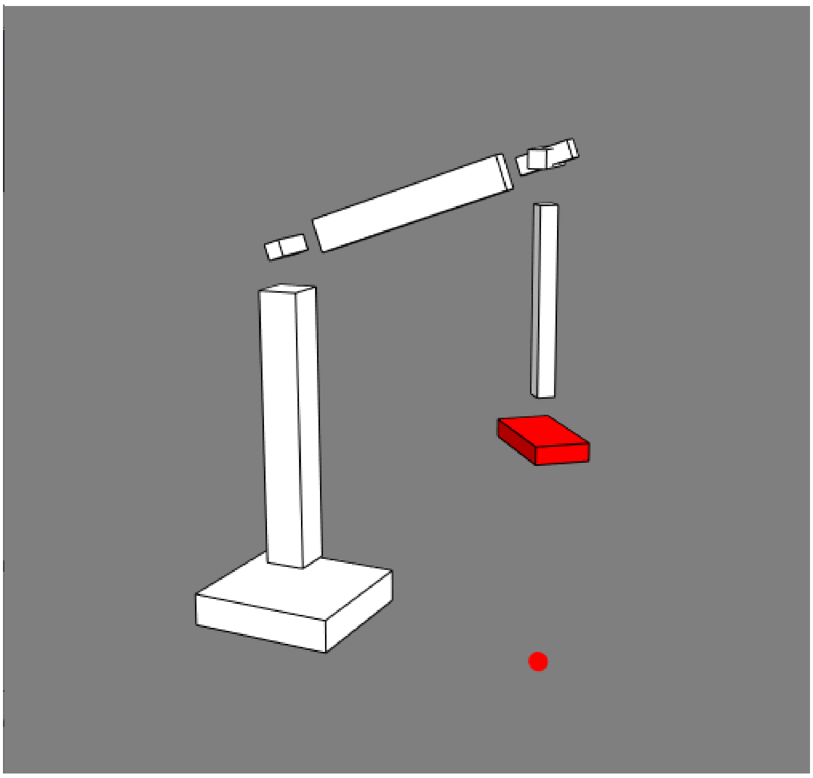


Fig. 14 A simplified 3D model of the robot manipulator

The Processing program was chosen to visualize the robot control process. It is an open source software based on Java [40]. This is a simple and fast set of tools for people who want to program images, animations and interfaces. Used by students, artists, designers, researchers and amateurs to study, prototype and manufacture. It is designed to learn the basics of computer programming in a visual context (meaning that each * .pde file in the Processing visual shell is a separate image or animation) and a professional production tool. This open source software is easy to learn, has a wide range of features, with this software there is an opportunity create programs in 2D and 3D, also created a large number of libraries that further expand the capabilities of this program. The official site of processing has a course of very useful lessons [41] that will allow quickly learn a beginner and start creating own projects.

The Arduino platform sends data from the sensor to the serial data port, in turn the visualization program constantly listens to the port and as soon as the port buffer receives the data packet, they are immediately written to the array for storage.

The basis of the program is a simplified model of an industrial robot with six stages of motion. The 3D model of this robot consists of primitive 3D figures - parallelepipeds. It is possible also resize an individual robot component. The 3D model is based on inverse kinematics. Inverse kinematics is an animation technique that searches for a set of joint configurations that will provide the smoothest, fastest, and most accurate movement to a given point. Inverse kinematics is one of the methods for planning the motion of multicomponent dynamical systems. Inverse kinematics is widely used in robotics, three-dimensional computer animation. It is mainly used in situations where it is necessary to precisely position the flexible connections of one object relative to other objects in the environment. The inverse kinematics algorithm is the opposite of the direct kinematics algorithm.

7.1 Getting data

The connection between the Arduino platform and the visualization program is made via a serial data port. After receiving data on the exact angle of inclination based on the gyroscope and accelerometer, as well as the control mode, the Arduino platform sends all the data to the serial port in strict order, separating each digit with a "/". To connect a processing-based program to a serial port, it is necessary initialize and connect to it. This is done in the next part of the code.

```
String port = Serial.list()[0];
serial = new Serial(this, "COM4", 9600);
```

Connects to port COM 4 at a speed of 9600 bits per second.

The next step is to obtain data from the serial port buffer:

```
if (serial.available() > 0)
    {
      received = serial.readStringUntil('\n');
      String[] list = split(received, '/');

      if(received != null)
      {
        a[0] = Float.parseFloat(list[0]);
        a[1] = Float.parseFloat(list[1]);
        a[2] = Float.parseFloat(list[2]);
        a[3] = Float.parseFloat(list[3]);
      }
    }
```

The function constantly checks whether the data appeared in the buffer, if so, they are written to variable. Then comes the parsing stage. Since the obtained data is represented by a string, they must be divided and converted into numeric values of type float. This is done using the split() function. This function writes to a variable a list of values that are separated from each other by the sign "/". Sorted data is converted to numeric variables using Float.parseFloat ();

7.2 The principle of control of the 3D model

The software uses the input from the sensor as a change of coordinates per unit time, on which the 3D model must move. Therefore, due to the different angle of inclination it is possible to adjust the speed. This makes controlling the robot very convenient and smooth.

7.3 Graphical interface

Then the function button(); is called which creates a simple graphical interface to display all important information, such as the coordinates of the work tool, sensor input, and so on.

```
text("Input DATA:", 940, 250);
text("X:", 965, 285);
text(a[1], 985, 285);

text("Y:", 965, 310);
text(a[2], 985, 310);

text("Z:", 965, 335);
text(a[3], 985, 335);

text("Camera position:", 1090, 450);
text("X rotate:", 1115, 475);
text(mouseX / 2, 1210, 475);

text("Y rotate:", 1115, 500);
text(mouseY / 2, 1210, 500);
```

For example, the display of input data will be considered. Each element is a block in which there is text, also in arguments of function coordinates X and Y for exact positioning of the block are specified. Function text (a [1], 985, 285); takes as its first argument a variable that contains the value of the angle of inclination along the X axis, so it is constantly updated.

The camera position is also displayed. Since it is adjusted with the mouse, the argument for displaying data takes the coordinates of the mouse on the X and Y axes.

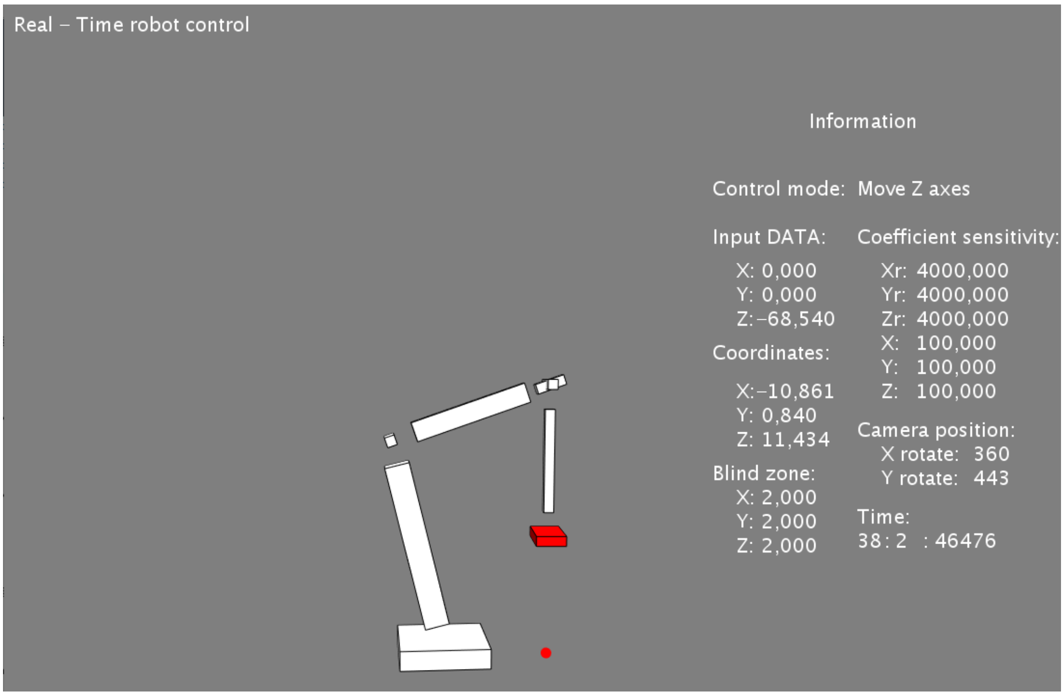


Fig. 15 A graphical interface that displays information

This block of information consists of several blocks that indicate such data.

1. **Control mode:** shows which type of control is currently enabled.
2. **Input data:** Data sent by the GY-85 sensor.
3. **Coordinates:** Coordinates of the main working tool, the position of the tool are projected on a horizontal surface in the form of a red dot.
4. **Blind zone:** The 3D model does not move if the sensor is not tilted more than 2 degrees
5. Sensitivity factor: The factor adjusts the sensitivity of the 3D model to the movement of the sensor, if the factor is too small, the 3D model will respond to minor movements and vibrations.
6. Camera position: The position of the camera relative to the 3D robot model, the position can be adjusted with the mouse or keyboard.
7. Time: Shows the time that has elapsed since the software was started.

7.4 Control settings

Special coefficients have also been added for easy control of the 3D model

1. The coefficient of sensitivity.
2. Blind zone coefficient.

These coefficients allow to change the sensitivity of the control, because the device on which the sensor will be attached will be in the hand of a person, so there will always be vibrations and minor movements that will affect the position of the 3D model. The blind zone setting allows to select the maximum tilt angle at which the robot will not respond to vibrations or slight tilts.

```
  if(a[1] >= -blindZoneX && a[1] <= blindZoneX)
  {
    a[1] = 0;
  }
  else if(a[1] > blindZoneX)
  {
    a[1] -= blindZoneX;
  }
  else if(a[1] < blindZoneX)
  {
    a[1] += blindZoneX;
  }
```

```
if(a[0] == 1)
{
  hpx += a[2] / sensCoefX;
  hpy += -a[1] / sensCoefY;
}
```

The sensitivity coefficients reduce the input data by several hundred times, so it is possible control the 3D model more smoothly.

For easier orientation, a horizontal grid has been added, on which the center of the main working tool is designed, this will make it easier to determine the position of the robot. It is also possible to control the camera rotation, for a better view, and it can be used if the real robot is fixed, for example, at an angle, on a vertical surface, or attached to the ceiling.

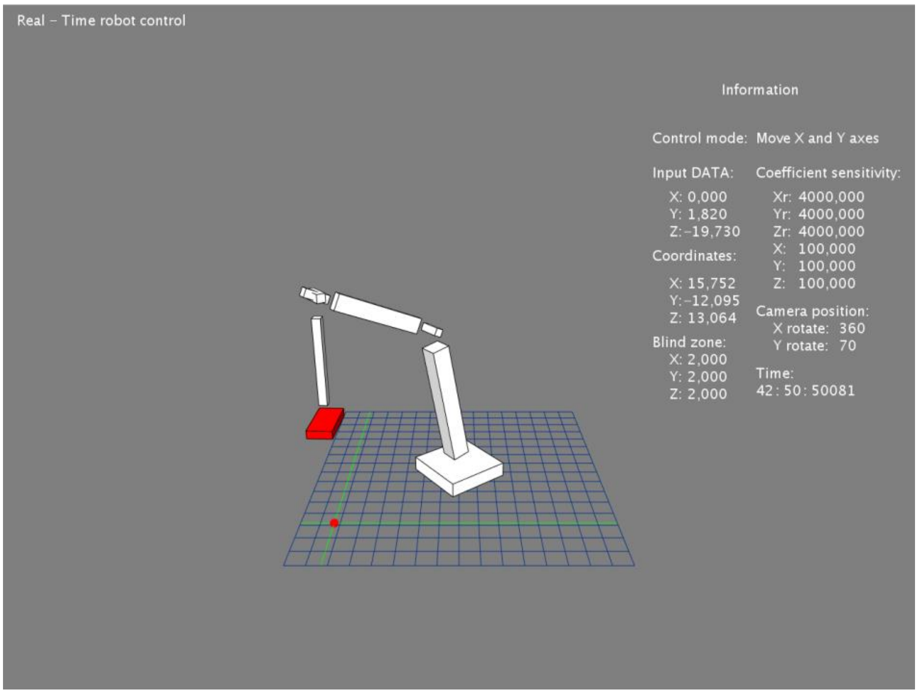


Fig. 16 Horizontal grid on which the coordinates of the main working tool are projected

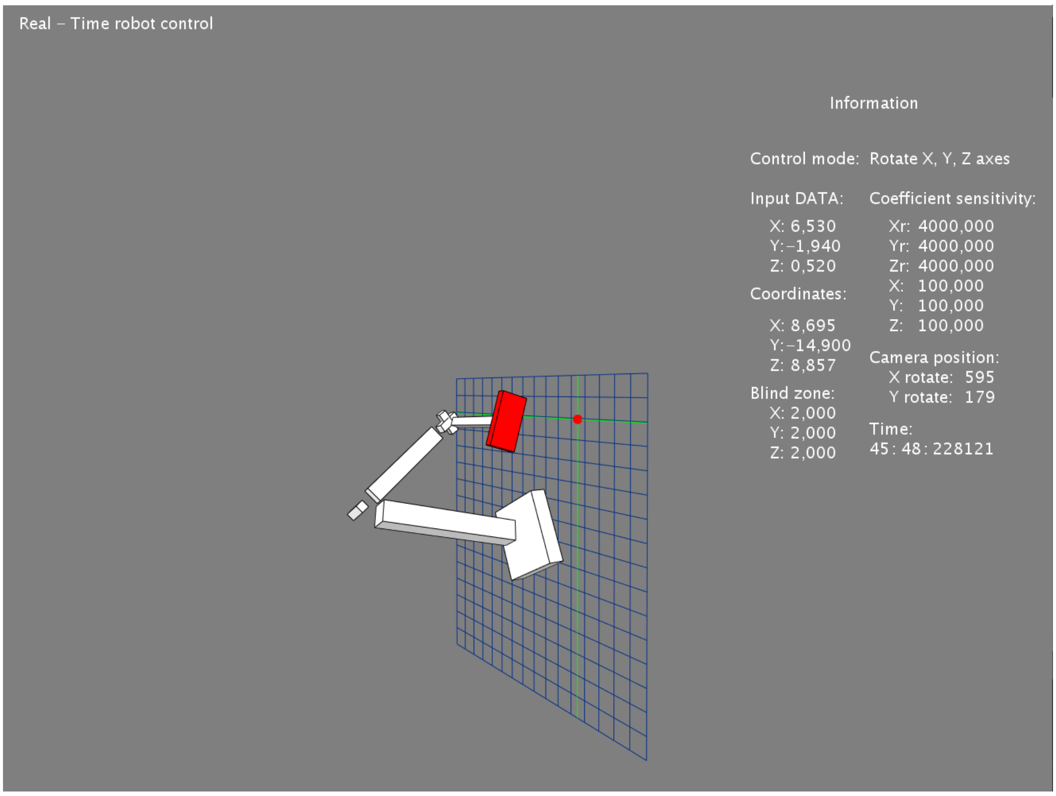


Fig. 17 Demonstrated the ability to change the position of the entire system to simulate different production situations

8 Application of technology

Accelerometer and gyroscope control technology (inertial measurement system) is relatively easy to develop, does not require special conditions of use and does not require large investments. This technology allows to control the object at some distance from it, or at close range, but not to interact with the object itself. Thus, this technology can be used in many areas. For example, it is possible to consider the case of moving large loads using a manipulator mounted on the truck. To operate the manipulator, the operator must be close to the truck, in some cases careless handling of the manipulator can lead to industrial injuries. The second aspect is that the operator does not have a complete overview and can control the situation only from one moment.

Fig. 18 The manipulator is mounted on a truck [42]

In some cases, IMU management technology may be more efficient. For example, if the load is placed in a hard-to-reach place, in this case, so that the operator can monitor the situation, he needs to have an overview from all sides. If there is a possibility of an emergency during operation, the operator must stay away from the manipulator.

This control system can be useful in the mining industry, large bulldozers have blind areas, and sometimes when transporting large loads it is necessary to keep the situation under control, for example, to control the movement of the bucket from the side, not in the cabin. Thus, smooth and slow movements can control the main tool of the bulldozer.

Fig. 19 Bulldozer with bucket [43]

Today, manufacturing companies must quickly adapt to changing demand for a product. Production lines must be flexible and ready to changes in the production process. Currently, the industry is actively using collaborative robots, which are adapted to work with people. Modification of the production line can be carried out at the expense of collaborative robots, they are easy to transport, and reprogramming can be carried out using technology based on IMU. This type of programming is not suitable for precise manipulations, such as welding, but is suitable for example for moving boxes with goods. Moving the semi-finished product along the production line. Maintenance of CNC machines.

This type of collaborative robot programming is much faster and easier, it is suitable for inexperienced workers who have no experience in programming industrial robots, they will be able to quickly master this method, so it will be possible to quickly make changes in the production process.

However, great attention should be paid to the safety of control mechanisms with this technology. Since the main device will be attached to the operator's arm, it is necessary to ensure the safety of the working space around the person, as any shock can lead to unpredictable movements of the manipulator, which can lead to a dangerous, emergency situation.

For safety, the main control device can be equipped with buttons and switches that will control the operation of the device. For example, the device will be activated only after pressing and holding one button. Pressing another button will stop the operation of the device, or be limited to only one axis of movement, or rotation. The security system can also be organized at the software level. If the angular velocity of one or more axes is greater than the set value, the device is suspended, and if the angle of the sensor is greater than the maximum, the manipulator stops moving, it will allow to work safely indoors with a low ceiling, not will develop a high speed of movement of the manipulator and so on.

Thus, this control technology can be used in many industries, taking into account all the safety measures that must be followed for efficient and safe operation.

9 Advantages and disadvantages of technology

Gyroscope and accelerometer control technology can be effective in a variety of situations, such as loading heavy or bulky loads or performing operations that do not require high accuracy. However, this technology, like all others, has its advantages and disadvantages.

The advantages include the fact that the hardware and software do not require large investments. The main elements of the hardware are the accelerometer and gyroscope, depending on their cost, the quality of the sensors will be different. The advantages also include the fact that the device can operate wirelessly, the robot can be controlled remotely. The employee will be at the command post and control the robot by monitoring it through monitors. In this way, there is an opportunity participate in a process that requires employee intervention, but is dangerous to humans. This technology is easy to learn for new employees who have not previously interacted with the robots - manipulators, management is simple and intuitive.

The disadvantages include the measurement error of the gyroscope, as it has a drift of zero. This means that the angle of inclination is constantly increasing, but in fact the gyroscope is in a stable and stationary position. Zero drift depends on the quality of the gyroscope, more expensive models of sensors have less measurement error. To compensate for the error, necessary to use data from the accelerometer and use a complementary filter to align the values. The disadvantage of the technology is also that during operation it is necessary to adhere to strict safety precautions. The device can be attached to the worker's arm, so the space around it must be safe, because the slightest shock can change the trajectory of the manipulator, which can lead to injuries. This technology does not guarantee high accuracy of positioning of the manipulator as all commands are set only by tilting of a hand.

Fig. 20 An industrial robot performs the operation of assembling boxes [44]

Conclusion

Robot programming is usually a complex process that requires a lot of time, investment and experienced professionals. The technology proposed in this article is easy to learn for new employees, well suited for small and medium enterprises, for technologically simple works, such as moving boxes, light semi-finished products. The proposed technology uses only two sensors of the cheap segment, and demonstrated good results of positioning accuracy of the 3D model of the robot, so depending on the size of the investment, the accuracy of the device can be increased many times, which in turn will increase accuracy and production efficiency. The great advantage of the proposed method is the extensibility and the ability to modify both the device and the software. The control can be flexibly modified and adapted to different technological conditions. Based on an opensource database is possible to create different options for controlling a 3D model or a real robot. For example, by controlling with sharp inclines to a certain position and back to the starting position, the manipulator will make single movements at a given distance or rotate at a specific angle. It is also possible to perform a pattern movement in which the employee performs only one operation, and with the help of software it is possible to create a pattern of this operation with the appropriate coordinates. Another feature of the technology is the possibility of application on various types of manipulators, ranging from industrial collaborative robots and ending with manipulators that are installed on trucks. The constant development and creation of new technologies will help in the future to develop new more intuitive methods of interaction of workers with robotics.

Bibliography

[1]. Sebastian Blankemeyer, Rolf Wiemann, Lukas Posniak, Christoph Pregizer. “Intuitive Robot Programming Using Augmented Reality” [online]. [cit. 2021-03-19] Available on the Internet: < https://www.researchgate.net/publication/327192365_Intuitive_Robot_Programming_Using_Augmented_Reality>

[2]. Sven Stumma,*, Johannes Braumannb,c, Sigrid Brell-Cokcana,c. “Human-Machine Interaction for Intuitive Programming of Assembly Tasks in Construction”

[3]. Julia Berga,*, Albrecht Lottermosera, Christoph Richtera, Gunther Reinharta. “Human-Robot-Interaction for mobile industrial robot teams”

[4]. Carole S. Franklin a,⇑, Elena G. Dominguez b, Jeff D. Fryman c, Mark L. Lewandowski d. “Collaborative robotics: New era of human–robot cooperation in the workplace”

[5]. Rafael A. Rojas ∗, Manuel A. Ruiz Garcia, Luca Gualtieri, Erwin Rauch. “Combining safety and speed in collaborative assembly systems – An approach to time optimal trajectories for collaborative robots”

[6]. Renjie Zhang, Xinyu Liu, Jiazhou Shuai, Lianyu Zheng. “Collaborative robot and mixed reality assisted microgravity assembly for large space mechanism”

[7]. Sebastian Blankemeyera,*, Rolf Wiemannb, Lukas Posniakc, Christoph Pregizerc, Annika Raatz. “Intuitive Robot Programming Using Augmented Reality”

[8]. Sonia Mary Chacko, Armando Granado, Vikram Kapila. “An Augmented Reality Framework for Robotic Tool-path Teaching”

[9]. S.K. Ong∗, A.W.W. Yew, N.K. Thanigaivel, A.Y.C. Nee. “Augmented reality-assisted robot programming system for industrial applications”

[10]. Torsten Sebastian Sieversa,*, Bianca Schmittb, Patrick Rückerta, Maren Petersenb, Kirsten Trachta. “Concept of a Mixed-Reality Learning Environment for Collaborative Robotics”

[11]. Jonas Wassermanna,*, Axel Vickb, Jorg Kruger. “Intuitive robot programming through environment perception, augmented reality simulation and automated program verification”

[12]. Christian Deuerlein a , ∗, Moritz Langer a , Julian Seßner b , Peter Heß a , Jörg Franke b. “Human-robot-interaction using cloud-based speech recognition systems”

[13]. Tudor B. Ionescu ∗, Sebastian Schlund. “Programming cobots by voice: A human-centere d, web-base dapproach”

[14]. Panagiota Tsarouchia, Athanasios Athanasatosa, Sotiris Makrisa, Xenofon Chatzigeorgioua, George Chryssolourisa*. “High level robot programming using body and hand gestures”

[15]. Mikhail Ostanin* Dmitry Popov* Alexandr Klimchik*. “Programming by Demonstration Using Two-Step Optimization for Industrial Robot.”

[16]. Aswin K Ramasubramanian, Nikolaos Papakostas ∗. “Operator - mobile robot collaboration for synchronized part movement”

[17]. Zheng Yang, Chenshu Wu, Zimu Zhou, Xinglin Zhang, Xu Wang, Yunhao Liu. "Mobility Increases Localizability", ACM Computing Surveys, 2015. [online]. [cit. 2021-03-04] Available on the Internet: < http://tns.thss.tsinghua.edu.cn/~yangzheng/papers/Yang-Mobility-CSUR.pdf >

[18]. Inertial measurement unit. [online]. [cit. 2021-03-19] Available on the Internet: < https://en.wikipedia.org/wiki/Inertial_measurement_unit >

[19]. Inertial measuring device of the Apollo spacecraft. [online]. [cit. 2021-03-19] Available on the Internet: https://en.wikipedia.org/wiki/Apollo_PGNCS

[20]. Inertial measurement unit. [online]. [cit. 2021-03-19] Available on the Internet: < https://military.wikia.org/wiki/Main_Page>

[21]. V-2 missile. [online]. [cit. 2021-03-19] Available on the Internet: https://www.britannica.com/technology/V-2-missile

[22]. Gyroscope. [online]. [cit. 2021-03-19] Available on the Internet: https://www.britannica.com/technology/gyroscope#ref823350

[23]. Ring Laser Gyroscopes. [online]. [cit. 2021-03-19] Available on the Internet: <https://www.findlight.net/blog/2019/12/22/ring-laser-gyroscopes-for-inertial-navigation-and-transportation-systems/ >

[24]. Arduino Nano Microcontroller – Documentation. [online]. [cit. 2021-03-04] Available on the Internet: < https://store.arduino.cc/arduino-nano >

[25]. Arduino platform power supply. [online]. [cit. 2021-03-04] Available on the Internet: < https://thepihut.com/blogs/raspberry-pi-tutorials/how-do-i-power-my-arduino >

[26]. GY - 85 sensor – documentation. [online]. Available on the Internet: <https://github.com/madc/GY-85>

[27]. GY - 85 sensor – information sheet. [online]. [cit. 2021-03-04] Available on the Internet: < https://www.analog.com/media/en/technical-documentation/data-sheets/adxl335.pdf >

[28]. Gyroscope sensor ITG-3205 - information sheet. [online]. [cit. 2021-03-04] Available on the Internet: < https://dl.btc.pl/kamami_wa/itg3205.pdf >

[29]. Power supply sensor gu – 85. [online]. [cit. 2021-03-05] Available on the Internet: < https://medium.com/jungletronics/gy-85-a-quick-datasheet-study-79019bb36fbf >

[30]. Software development environment. [online]. [cit. 2021-03-05] Available on the Internet: < https://www.arduino.cc/en/guide/environment >

[31]. Description of the data transfer protocol. [online]. [cit. 2021-03-05] Available on the Internet: < https://en.wikipedia.org/wiki/I²C >

[32]. Data transmission, I2C protocol. [online]. [cit. 2021-03-05] Available on the Internet: < https://deepbluembedded.com/i2c-communication-protocol-tutorial-pic/ >

[33]. Library Wire.h for communication with computer port. [online]. [cit. 2021-03-06] Available on the Internet: < https://www.arduino.cc/en/reference/wire >

[34]. MEMS Accelerometer Technology. [online]. [cit. 2021-03-02] Available on the Internet:<https://www.analog.com/en/analog-dialogue/articles/mems-accelerometers-as-acoustic-pickups.html >

[35]. How Accelerometer works. [online]. [cit. 2021-03-02] Available on the Internet: <https://lastminuteengineers.com/adxl335-accelerometer-arduino-tutorial/>

[36]. MEMS Accelerometer Gyroscope Magnetometer. [online]. [cit. 2021-03-02] Available on the Internet: < https://howtomechatronics.com/how-it-works/electrical-engineering/mems-accelerometer-gyrocope-magnetometer-arduino/ >

[37]. Gyroscopes and Accelerometers on a Chip. [online]. [cit. 2021-03-02] Available on the Internet: <https://www.geekmomprojects.com/gyroscopes-and-accelerometers-on-a-chip/>

[38]. L3G4200D gyroscope proof mass elements. [online]. [cit. 2021-03-02] Available on the Internet: < https://www.memsjournal.com/2011/01/motion-sensing-in-the-iphone-4-mems-gyroscope.html >

[39]. Information sheet ADXL335. [online]. [cit. 2021-03-02] Available on the Internet: < https://www.analog.com/media/en/technical-documentation/data-sheets/adxl335.pdf >

[40]. Visualization software. [online]. [cit. 2021-03-02] Available on the Internet: <https://processing.org >

[41]. Tutorials for studying Processing. [online]. [cit. 2021-03-02] Available on the Internet: < https://processing.org/tutorials/ >

[42]. Truck with manipulator. [online]. [cit. 2021-03-27] Available on the Internet: <https://tractors.fandom.com/wiki/Caterpillar_D9T>

[43]. Bulldozer with bucket. [online]. [cit. 2021-03-27] Available on the Internet: <http://www.farmingmods2015.com/scania-730-truck-trailers-v-1-2/scania-730-truck-and-trailers-1/>

[44]. Collaborative robot. [online]. [cit. 2021-03-27] Available on the Internet: <https://www.cobottrends.com/minipal-cobot-palletizier-certified-ur-application-kit/>