People’s Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
Saad Dahlab University -Blida 1-
Electronics Department

Thesis to obtain a master degree in control system engineering by:

Dilmi Mohamed Yacine and Kaddour Bachir

Design of a SCARA type, vision based robot

Supervisor: Boualem kazed

Abstract

College year: 2017-2018

Today and more than ever, the market needs highly trained robotics engineers, to keep up with the required skills we choose to start from the basics and build a low cost arm manipulator from scratch using only the minimum building blocks available on electronics stores.

This work shows only the bare minimum of mathematical knowledge to build robotic arm capable to achieve a simple pick and place task.

Key words: SCARA, Forward kinematics, Inverse kinematics, Pick and place.

Résumé

Aujourd'hui et plus que jamais, le marché a besoin d'ingénieurs robotiques hautement qualifiés. Pour suivre les compétences requises, nous décidons de partir des bases et de créer un bras manipulateur peu coûteux en utilisant uniquement le minimum des composants de base disponibles sur les magasins d'électronique.

Ce travail montre seulement le strict minimum de connaissances mathématiques pour construire un bras robotique capable de réaliser un assemblage simplifié.

Mots clés: SCARA, Cinématique direct, Cinématique inverse, assemblage

---

ملخص

إخترنا البدء من الصفر في تصميم وتصنيع ذراع تحكم آلية منخفضة التكلفة باستخدام الحد الادنى من القطع المصممة خصيصا لأغراض مماثلة والمتاحة تجاريا في محلات الإلكترونيات للحاق ومجاراة المهارات المطلوبة في سوق الوظائف التقنية الذي بات بحاجة ملحة وأكثر من أي وقت مضى لخبراء الروبوتات.

هذا العمل يتضمن الحد الأدنى من المفاهيم الرياضية اللازمة لتصميم ذراع آلية قادرة على أداء مهمات التجميع البسيطة.

كلمات مفتاحية: الحركية الأمامية، الحركية العكسية، ذراع آلية، مهام التجميع

Table of content

Chapter 1: Robotics evolution from science fiction novels to real world applications.

Chapter 2: Simulation of SCARA type arm manipulator.

Chapter 3: building a SCARA type arm manipulator

Figures list

Figure number & Title *Page number*

Chapter 1: Robotics evolution from science fiction novels to real world applications.

Chapter 2: Simulation of SCARA type arm manipulator.

Chapter 3: building a SCARA type arm manipulator

General introduction

While two different groups working on two different projects, a PUMA robotic arm configuration and a mobile robot, our part is to build a 4 degree of freedom SCARA type arm manipulator which will be fixed on the mobile part and collaborate with the other robotic arm on the same workspace to achieve certain tasks without collisions, a big project like this designed to use later for pedagogic purposes.

In this work we will work only on the kinematics problems of the SCARA type arm, to do so we need some mathematical tools like trigonometry and linear algebra to make sure that there is a solution for any given task that's why we need to compute all possible position could occupied by end effector tool (Forward kinematics) and to calculate how much and in which direction we need to rotate the motors to pose the gripper on the desired position to pick up or place a target (inverse kinematic).

We chose Arduino platform as a microcontroller because of its built-in functions that save us time and efforts to deliver a prototype as fast as possible. There is a lot of building blocs, ready to use solutions available on the market like gripper, servos brackets, linear actuator but we decided to build and design every joint ourselves from scratch using only the common tools such us Drill, Saw and Screws which are the right tools to deal with the wood, a cheap material easy enough to deal with.

Chapter 1: Robotics evolution from science fiction novels to real world applications.

- ❏ A brief history of arm manipulators.
- ❏ A technical view on robotic arms.
- ❏ A promising fields of applications.

1.1 A journey from Asimov’s three laws of robotics to the first arm manipulator and beyond

Karel capek was the first to introduce the word “robot” to the world by his R.U.R play (stands for Rossum’s Universal Robots a major industrial power that makes artificial people incapable of thinking, called “roboti”’ (derived from the Slavic word robota, it means forced labour). They are much more replicants (a bioengineered humanoid designed to afford hard work) rather than machinery which is close to the current definition of the term. [*1*]

While karel capek just introduce the word “robot” to the english language, Isaac Asimov a biochemistry teacher from boston university applied the suffix “-ics” and created the term “robotics” as we know today in his short story “Liar!” which is the earliest recorded use of the word “robotics” according to the oxford dictionary, actually asimov was not initially aware of this, he assumed the word already existed. Of Course his contribution to the field bigger than just coin a term, he inspired by his sci-fi novels many filmmakers to direct a movies shaped the meaning of robotics in popular culture today. [*2*]
Asimov mostly known by his three laws of robotics a three principles became a standard when we ask ethical questions about future robots, like decision making algorithms in self driving cars. [*3*]

After nearly two decades from asimov robot short stories getting published, the first digitally operated and programmable robot invented by George Devol and call it “unimate” in 1954, The key innovation was the “programmability” of the machine: it could be retooled and reprogrammed at relatively low cost so as to enable it to perform a wide variety of task. It was sold to General Motors in 1960, one year later it was installed to lift pieces of hot metal from die casting machines which is a dangerous task for workers who might be poisoned by toxic fumes or lose a limb if they were not careful. [*4*]

General Motors in the late 60s, became the most automated car maker in the world with production speed never achieved before when rebuilt its Ohio plant using Unimate spot welding robots which produce cars per hour faster more than two times the rate of any automotive plant in existence at the time! [*4*] [*5*]

At the same year (1969) and for the sack of make a robot capable of more sophisticated applications Victor Scheinman at Stanford University invented the Stanford arm, an all-electric, 6-axis articulated arm, both Unimation company and General motors bought the designs, developed them and later marketed it as the Programmable Universal Machine for Assembly also known as PUMA. [*6*]

After 4 years ABB Robotics introduced IRB 6, the world's first microcomputer controlled electric industrial robot side by side with KUKA robotics with its First industrial robot “FAMULUS” the first articulated robots to have six electromechanically driven axes. [7] [8]

In 1978, in the laboratory of Professor Hiroshi Makino, at Yamanashi University, a revolutionary arm manipulator prototype was designed to deal with some challenging pick and place tasks, a robot named SCARA (acronym stands for Selective Compliance Assembly Robot Arm) it excels in many types of assembly operations with high precision, speed, and smooth motion. [9]

Robotic arms until this day still make a progress in price / performance ratio where even a hobby level arm manipulator today could achieve a stunning tasks.

1.2 All about arm manipulators

Before going any further we need to define exactly what an arm manipulator is, according to the International Federation of Robotics:

“a Manipulating industrial robot is an automatically controlled, reprogrammable, multipurpose manipulator programmable in three or more axes, which may be either fixed in place or mobile for use in industrial automation applications” [10]

1.2.1 Robotic arm anatomy [17]

It is clearly from the definition that any arm manipulator must consists of the following parts shown on the diagram below:

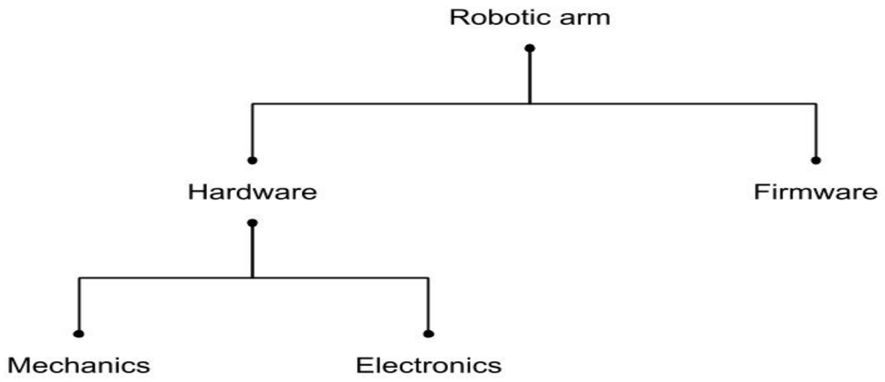

Figure 1: Anatomy diagram of robotic arm

Mechanics:

a frame designed to achieve a desired task with consideration of the physics conditions that surround it, a mobile arm manipulator designed to travel across a hard road might be use caterpillar tracks instead of wheels and attached with the right tool on its end effector to perform the assigned task (drilling for exemple).

Electrics:

Even if the arm operate completely on hydraulics and/or pneumatics actuator it still need electrical components such as battery, sensors and microcontroller which power, measure and control the mechanism.

Software:

An algorithm decides how and when the arm manipulator do something, it may have the correct mechanical design receiving the correct amount of power but it would not go anywhere without a program telling it to move. if the program poorly written, it will affect the performance very bad. robots must interact with their environment without human intervention.

1.2.2 Robotic arm parameters [*11*]

Number of axes: two enough to reach any point in a plan, three are required to reach any point in space. To fully control the orientation of the end effector three more axes (yaw, pitch, and roll) are required.

Degrees of freedom: number of possible independent relative motions between the pieces of the mechanism.

Working envelope: a space representing all positions which may be occupied by the robot.

Kinematics: describe the motion of the arm without reference to the forces which cause it.

Payload: (Carrying capacity) how much weight a robot can lift.

Speed: how fast the robot can position the end of his arm. either by angular or linear speed of each axis or as the speed of end effector when all axes are moving.

Acceleration: a robot may not be able to reach its specified maximum speed for a complex path or movements requiring frequent changes of direction over a short distance that’s why acceleration is a limited factor.

Accuracy: is the difference between the absolute position of the robot and the desired position.. Accuracy affected by speed, workspace and payload. It can be improved with external sensing like: camera, ultrasound or Infrared.

Motion control: sophisticated applications such as spray painting requires continuously controlled motion to follow a path in space, with highly precision speed and orientation while simple pick and place tasks have a limited number of pre-taught positions that does not require such as tough continuous control.

Power source: some robots use electric motors (faster), others use hydraulic actuators (stronger).

Drive: while some robots connect electric motors to the joint directly others connect motors to the joints via gears.

Compliance: a measure of the amount in angle or distance movement that a robot axis will move when a certain force is applied to it. Compliance can be responsible for overshoot when carrying high payloads.

1.2.3 Basic manipulator geometries

A joint is a connection between two rigid bodies which allows movement with one or more degrees of freedom, almost all arm manipulators whatever their mechanical design based on two types of joints either revolute and / or prismatic.

Revolute joints is a one degree of freedom kinematic pair provide single-axis rotation function while a prismatic joint provides single-axis linear sliding movement.

Figure 2: Prismatic joint Revolute joint

By combining a series of revolute and / or prismatic joints we will get 4 robotic arm geometries:

Rectangulaire Arm Geometry:

Use cartesian coordinates to move linearly in each direction (forward and backward, left and right, up and down) using only prismatic joints along x, y, z axes to generate a rectangulaire work envelope.

Cylindrical Arm Geometry:

Moved linearly in two directions (vertical and horizontal motion) and rotate in one other, its given the designation of R2P (revolute and two prismatic joints). This type have a cylindrical work envelope.

Spherical Arm Geometry:

With two revolute joints rotating in its base and shoulder and one prismatic joint sliding at the reach of the arm this robot is given the designation 2RP and has a spherical work envelope.

Articulated Arm Geometry:

Have rotation in three directions and three axis at least using revolut coordinates, that's why it given the designation of 3R, left and right movements are provided by the rotation of the base, horizontal movement by the shoulder and vertical movement by the elbow, the work envelope of this robotic arm is almost entirely spherical.

Figure 3:Graphical representation of manipulator geometries and their work envelope

1.2.4 Robotic arm geometries comparison

Arm Geometry	Advantages	Disadvantages
Rectangulaire	❑ Have the simplest geometry and control system. ❑ Have the largest work envelope in case of Gantry robot type.	❑ Workspace smaller than robot volume in case of cartesian robot type. ❑ Unable to reach areas under objects.
Cylindrical	❑ Allows for quick movement with high repeatability. ❑ A smaller use of floor space. ❑ A larger payload capacity due to the structural rigidity.	❑ Presmatic guides difficult to seal from dust and liquides. ❑ Back of the robot can overlap work volume.
Spherical	❑ Covers a large volume from a central support. ❑ Can bend down to pick objects up off the floor.	❑ Their high-cost, large use of floor space and lack of flexibility make it hard to justify in most industrial applications compared to articulated robots .
Articulated	❑ Covers a large workspace relative to its volume. ❑ Minimal floor space use and high positioning mobility of end effector.	❑ higher hardware cost and requires skilled technicians.

1.3. Arm manipulators invade the world

We already have robotic manipulators replaced human workforce in industrial zones to perform sophisticated and heavy tasks with high precision and speed, others receiving payloads coming to international space station, taking simpels and run tests on martian soil, finishing dangerous tasks in hostile environment (like a contamination sector of nuclear reactor) and so on.

Robotic arms still make a progress and go deeper in every aspect of our lives and keep moving from research labs to the market, the following examples are a samples of what the near future holds:

FarmBot, it's time to own your food.

FarmBot is an open source farming robot with rectangulaire geometry (Gantry type), it sows seed in any pattern and density you want through a web based interface (so no coding is required), waters them efficiently the exact amount that each plant needs based on its type, age, soil conditions and the local weather.

Using the onboard camera and advanced computer vision software, FarmBot intelligently monitors your garden, detect weeds as soon as they emerge and buries them under the soil. With the soil sensor it can show you how your garden changes over time enabling smarter, more efficient farming with each passing season.

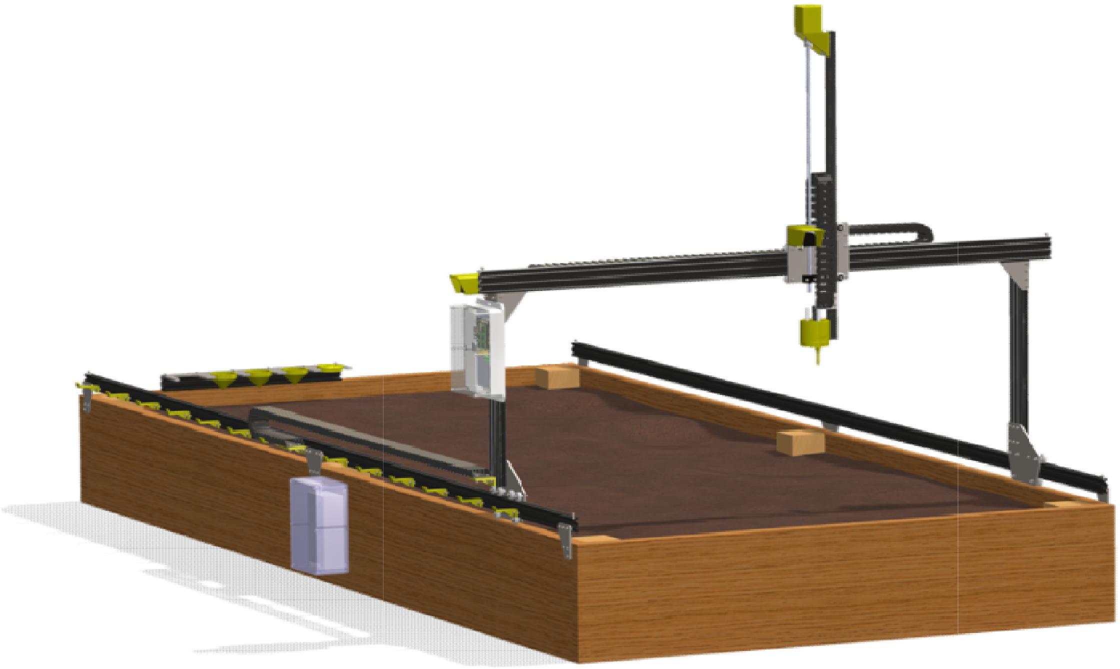

Figure 4: A CAD model for FarmBot Genesis V0.9

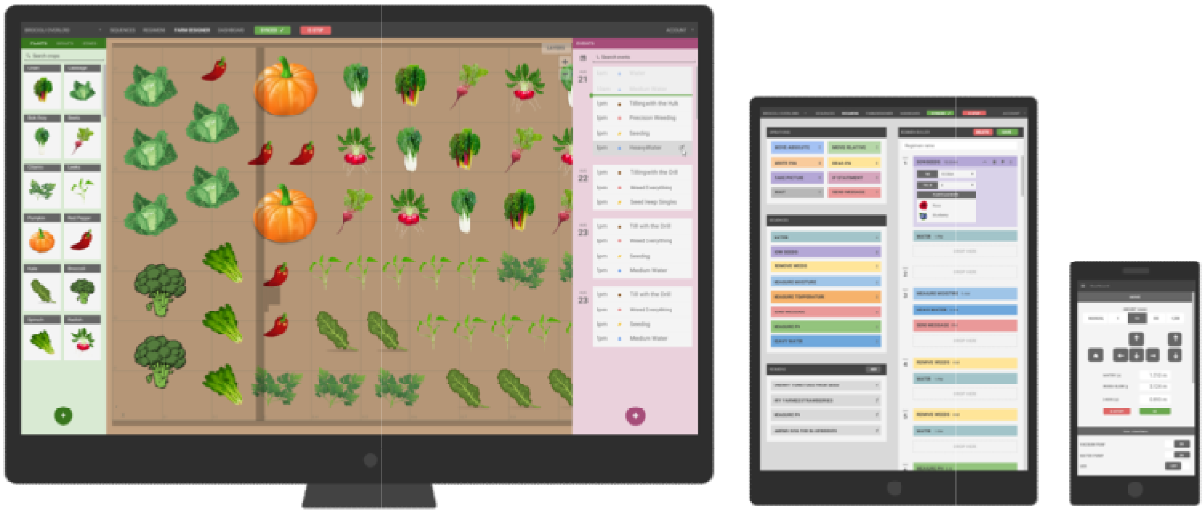

Figure 5: FarmBot Genesis Web App on different devices.

Moley Robotics, a five-star chef in your kitchen [12]

With two articulated robotic hands that can pick up and interact with most kitchen equipment and gesture recognition algorithms capturing and learning actions from a Master Chef while he cooks a meal, Moley Robotics the company behind this robot prototype aims in the near future to unable the customer to choose from more than 2000 recipes, while now the user operates the installation via a built-in touchscreen or smartphone application with cooking ingredients prepared in advance and put in preset locations.

Figure 6: Moley Robotic kitchen cocked a meal

The company hire a masterchef winner Tim Anderson to help teaching their robot how to cock by using motion capture gloves and integrated 3D camera to reproduce the whole sequence of actions later to cook an identical meal from scratch. The consumer version is slated to launch in 2018.

The prototype made its debut at the Hannover Messe industrial robotics trade fair in Hannover, Germany in April 2015, and won both CES Shanghai and UAE AI & Robotics Award.

Cobots, where robots and humans working together side by side [13]

collaborative robots are safe enough to to physically interact with humans in a shared workspace, designed to minimize installation space requirements, it could operate autonomously or with limited guidance to unlock a new automation potential in industry.

Figure 7: YuMi, world's first truly collaborative dual-arm robot

This type of robots are generally intended to assist not replace the production workers, They are often relatively lightweight and easy to program (usually by training them manually for their new tasks using your hands). There are two basic ways to make cobots safe:

1. If it makes contact with a human co-worker, it immediately stops so that the worker feels no more than a gentle nudge. Rounded surfaces help make that nudge even gentler. This approach limits the maximum load that the robot can handle as well as the speed. A robot moving a 25 kg part at high speed is going to hurt no matter how quickly it can stop upon making contact.
2. A sensor-based approach allows the robot to slow down, work around the person or stop as the situation demands to maintain safety. When the person moves away, the robot can automatically resume normal operation

Chapter 2: Simulation of SCARA type arm manipulator.

- ☐ Robotic Arm configuration.
- ☐ Forward kinematic.
- ☐ Inverse kinematic.

2.1 Robotic arm design

We will design a 4 degree of freedom SCARA type arm manipulator where the first joint is prismatic while the rest are revolute (the first two used to reach any point in x,y plan and the third one for end effector orientation).

Figure 8: Our robotic arm configuration

RVC Toolbox [14]

We will use Peter Corke Robotic, Vision and control Toolbox which provides many functions that are useful for the study and simulation of classical arm-type robotics, such things as kinematics, dynamics, and trajectory generation.

Denavit-Hartenberg convention

D-H parameters define the motion of actuators connected by rigid links, this is useful for efficient calculation of forward and inverse kinematics, the process begin by:
Defining the z axis along the axis of rotation for revolut joints or the axis of translation for prematic joints, since this is the first joint the x axis is free choice while the y axis complet the right handed coordinate frame.

When we add another joint we set the coordinate frame as before, to determine the transformation between them we use DH-parameters which are derived from the common normal between there z axis, the common normal is a line perpendicular on two non-intersecting joint axes, the new x axis points along the common normal and has its origin at the intersection of the new z axis. [15]

Notice that the origin is not with in the physical actuator because the D-H parameters are only concerned with motion of the links not the physical placement of components, using this protocol for laying out the reference frames only 4 parameters are needed: [16]

1. □: offset along previous z to the common normal.
2. □: rotates about the previous z axis to line the x axis.
3. □: the length of the common normal itself.
4. □: rotates about the new x axis to bring z into alignment with the axis motion.

D-H convention implementation

Since each joint connects two links, a robot manipulator with n joints will have n+1 links. We number the joints from 1 to n, and we number the links from 0 to n, starting from the base. By this convention, joint i connects link i − 1 to link i. When joint i is actuated, link i moves.Therefore link 0 (the first link) is fixed, and does not move when the joints are actuated.

Figure 9: Attaching D-H reference frames to a SCARA type robot

Link	□	d	a	□
1	0	$□_1$	0	0
2	$□_1$	0	$□_1$	0
3	$□_2$-90	0	0	-90
4	0	$□_2$	0	0

5	□$_3$	□$_3$	0	0

D-H table of (**PRRR)** arm manipulator configuration

Matlab simulation

We added an inactive prismatic joint between second and third revolute joint just for align the last one horizontally. After we assign a random translation and rotation angles to our prismatic and revolute joints, we got our arm manipulator position its end effector on certain x,y,z coordinate.

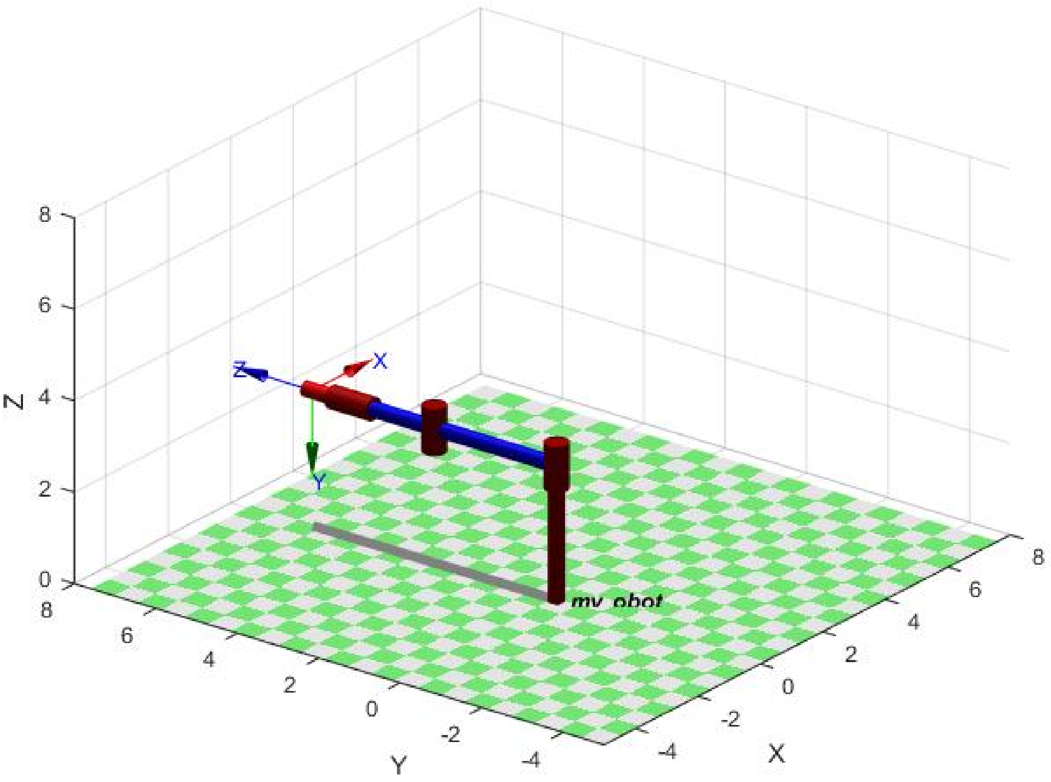

Figure 10: An abstract design of a “PRRR” arm manipulator configuration on matlab

2.2 SCARA Forward kinematic

2.2.1 SCARA work envelope

In order to visualize all possible positions could occupied by arm manipulator end effector on 2 dimension plan, we need an equation that gives us the coordinate (x,y) of the arm tip for any given theta 1 and 2 (rotation angles of our revolute joints). To obtain the solution we choose to figure it out geometrically:

Figure 11: Geometrical solution of end effector coordinates given any theta 1 and 2

So for any $\square_1$ & $\square_2$ we have:

$$\square = \square_1 \times \square\square\square(\square_1) + (\square_2 + \square_3) \times \square\square\square(\square_1 + \square_2)$$

$$\square = \square_1 \times \square\square\square(\square_1) + (\square_2 + \square_3) \times \square\square\square(\square_1 + \square_2)$$

We attach 180 degree range servo motors to the revolute joints, we assume that:

1. $\square_1 = (\square_2 + \square_3) = 3$.
2. $\frac{-\square}{2} \leq \square_1 \leq \frac{\square}{2}$ while $0 \leq \square_2 \leq \square$.
3. We will fixe $\square_1$ at $\frac{-\square}{2}$ while varying $\square_2$ from 0 to $\square$ by 0.1 on each iteration, after that we will again fix $\square_1$ at the previous angle plus 0.01 and varying $\square_2$ as before and so on, we will store all the (x,y) coordinates of arm tip and plot it using Matlab.

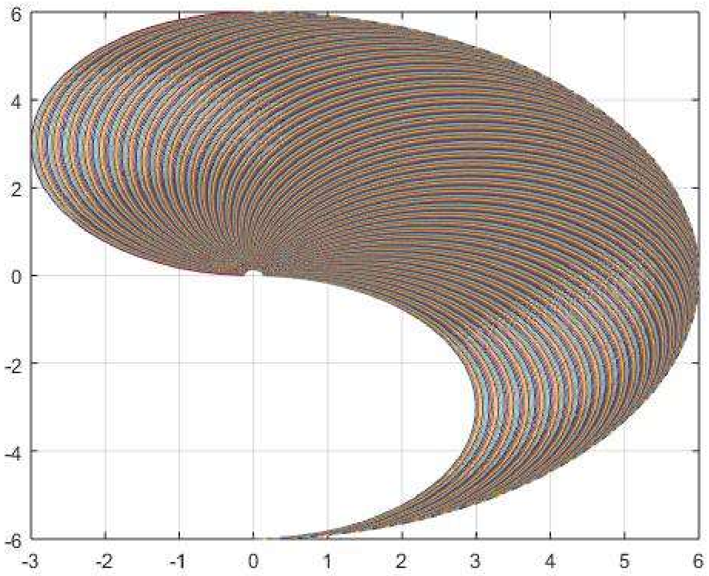

Figure 12: Illustration showing all possible position of arm manipulator end effector

2.2.2 Pose presentation

To write the kinematic equations of our SCARA type robot we chose Denavit-Hartenberg notation which offers a unified mathematical description of translational and rotational displacement, where a homogeneous matrix is used to represent both the position and orientation of each fram by a coordinate transformation.

$$
{}^{n-1}T_n = \mathrm{Trans}_{z_{n-1}}(d_n) \cdot \mathrm{Rot}_{z_{n-1}}(\theta_n) \\ \cdot \mathrm{Trans}_{x_n}(r_n) \cdot \mathrm{Rot}_{x_n}(\alpha_n)
$$

$$
{}^{n-1}T_n = \left[\begin{array}{ccc|c} \cos\theta_n & -\sin\theta_n\cos\alpha_n & \sin\theta_n\sin\alpha_n & r_n\cos\theta_n \\ \sin\theta_n & \cos\theta_n\cos\alpha_n & -\cos\theta_n\sin\alpha_n & r_n\sin\theta_n \\ 0 & \sin\alpha_n & \cos\alpha_n & d_n \\ \hline 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{ccc|c} & R & & T \\ \hline 0 & 0 & 0 & 1 \end{array}\right]
$$

R is the 3×3 submatrix describing rotation and T is the 3×1 submatrix describing translation.

Using Matlab symbolic Toolbox we will compute the homogeneous matrix that gives us the position and orientation of arm manipulator end effector for any translation d and rotations (θ_1, θ_2, θ_3):

$$^{0}T_{1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{1}T_{2} = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 3\cdot\cos(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 3\cdot\sin(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{2}T_{3} = \begin{bmatrix} \cos(\theta_2-90) & 0 & -\sin(\theta_2-90) & 0 \\ \sin(\theta_2-90) & 0 & \cos(\theta_2-90) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{3}T_{4} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{4}T_{5} = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & 0 \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The presentation of end effector position and orientation for any given □, □$_1$, □$_2$ and □$_3$:

$$^{0}□_{4} = {}^{0}□_{1} \cdot {}^{1}□_{2} \cdot {}^{2}□_{3} \cdot {}^{3}□_{4} \cdot {}^{4}□_{5}$$

$$^{0}T_{4} = \begin{bmatrix} c(\theta_1+\theta_2-90)\cdot c(\theta_3) & -c(\theta_1+\theta_2-90)\cdot s(\theta_3) & -s(\theta_1+\theta_2-90) & 3\cdot c(\theta_1)-3\cdot s(\theta_1+\theta_2-90) \\ s(\theta_1+\theta_2-90)\cdot c(\theta_3) & -s(\theta_1+\theta_2-90)\cdot s(\theta_3) & c(\theta_1+\theta_2-90) & 3\cdot c(\theta_1+\theta_2-90)+3\cdot s(\theta_1) \\ -s(\theta_3) & -c(\theta_3) & 0 & d1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.3 A simple “pick and place” Algorithm

2.3.1 inverse kinematic of SCARA type robot

Most of the time you need to set the arm manipulator end-effector on certain position and orientation, you can’t do this right unless you know exactly how much and at which direction you need to move the joints, that’s why the inverse kinematic is much more useful than forward kinematic.

We will take geometrical approach to find a mathematical model that gives the two angels we need to position the arm tip on certain coordinate.

Figure 13: Finding the joints angles

By drawing an imaginary line called *c* opposite to $\square$, we make a right triangle *xyc*, using pythagorean theorem $\square = \sqrt{\square^2 + \square^2}$*(1).*
We have a simple relationship between $\square_2$ and $\square$, where $\square_2 = \square - \square$ *(2)* so we need to find $\square$. In order to do that, we need to use the cosine rule:

$$\square^2 = \square^2 + \square^2 - 2\cdot\square\cdot\square\cdot\square\square\square(\square) \Rightarrow \square\square\square(\square) = \frac{\square^2 + \square^2 - \square^2}{2\cdot\square\cdot\square} \Rightarrow \square\square\square(\square) = \frac{\square^2 + \square^2 - \square^2 - \square^2}{2\cdot\square\cdot\square}$$

Now we have $\square\square\square(\square)$: $\square = \square\square\square^{-1}(\frac{\square^2+\square^2-\square^2-\square^2}{2\cdot\square\cdot\square})$

$$\square_2 = \square - \square \quad \Rightarrow \quad \square_2 = \square - \square\square\square^{-1}(\frac{\square^2+\square^2-\square^2-\square^2}{2\cdot\square\cdot\square})$$

$\square_1$ is the only angle was left, we can establish a relationship between $\square_1$ and $\square$, where: $\square_1 = \square - \square$; $\square = \square\square\square^{-1}(\frac{\square}{\square})$ while $\square = \square\square\square^{-1}(\frac{\square\cdot\square\square\square(\square_2)}{\square+\square\cdot\square\square\square(\square_2)})$

So:

$$\square_1 = \square\square\square^{-1}\left(\frac{\square}{\square}\right) - \square\square\square^{-1}\left(\frac{\square\cdot\square\square\square(\square_2)}{\square+\square\cdot\square\square\square(\square_2)}\right)$$

2.3.2 A modified inverse tangent

Inverse tangent function suffers from these two shortcomings:

1. It can not distinguish between any pair of points opposite from each other on the unit circle. Like it seems in the following figure:

Figure 14:inverse tangent cannot distinguish between opposite points on the unit circle

2. As a result of the cosine of $\pm\frac{\square}{2}$ is 0 ,the ratio of sine over cosine is undefined. Thus, the inverse tangent function fails at these points. To avoid this problem the angle must be in the range of $\frac{-\square}{2}$to$\frac{\square}{2}$, non- exclusive.

As long as the traditional inverse tangent function facing trouble dealing with this two problems we need an alternative solution. An adaptation of the inverse tangent that takes two inputs, y and x, and determines the resulting angle theta, based on the ratio, as well as the signs of y and x.This function is *atan2(y,x)*.

2.3.3 Matlab simulation

Until now we have:

1. The D-H parameters that we need to design the arm manipulator on Matlab.
2. A work envelope representing all possible position the end effector could occupied.
3. A mathematical model gives the joints how much and which direction to move in order to place the arm tip on the desired position.
4. A function (*atan2(y,x)*) returns the correct angels in the entire range of 0 to*2*□for revolut joints rotation.

So, we already have all what we need to simulate an arm manipulator performing a simple pick and place task.

Figure 16: A SCARA type robot picks a red target and place it on the black dot

After we have confirmed visually that the arm manipulator puts its end effector precisely on targets coordinates, we will prove it mathematically by using inverse kinematic to get the rotation angels of the two revolute joints $\square_1$and $\square_2$and then put them on homogeneous transformation matrix. If the last submetrix 3x1 is identical with both $\square_1$and $\square_2$ then the proof is correct.

Picking the red targets from $\square_1 = (5\ 1\ 2)', \square_2 = (3\ 4\ 4)'$ and places them on $\square_3 = (1.5\ 3\ 2)'$, we suppose that $\square = \square = 3$.

We have:

$$\square_2 = \square\square\square^{-1}\left(\frac{\square^2 + \square^2 - \square^2 - \square^2}{2 \cdot \square \cdot \square}\right)$$

$$\square_1 = \square\square\square\square 2(\square, \square) - \square\square\square\square 2(\square \cdot \square\square\square(\square_2), \square + \square \cdot \square\square\square(\square_2))$$

$$ {}^{0}T_{4} = \begin{bmatrix} c(\theta_1+\theta_2-90)\cdot c(\theta_3) & -c(\theta_1+\theta_2-90)\cdot s(\theta_3) & -s(\theta_1+\theta_2-90) & 3\cdot c(\theta_1)-3\cdot s(\theta_1+\theta_2-90) \\ s(\theta_1+\theta_2-90)\cdot c(\theta_3) & -s(\theta_1+\theta_2-90)\cdot s(\theta_3) & c(\theta_1+\theta_2-90) & 3\cdot c(\theta_1+\theta_2-90)+3\cdot s(\theta_1) \\ -s(\theta_3) & -c(\theta_3) & 0 & d1 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

□$_1$ = (5 1 2)′:

□ $_1$ = −20.4962□ $_2$ = 63.6122 67.1146

$$ {}^{0}T_{4} = \begin{bmatrix} 0 & 0.6835 & 0.73 & 5 \\ 0 & -0.73 & 0.6835 & 1 \\ 1 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

□ $_2$ = (3 4 4)′:

□ $_1$ = 19.5728□ $_2$ = 67.1146

$$ {}^{0}T_{4} = \begin{bmatrix} 0 & 0.9983 & 0.0578 & 3 \\ 0 & -0.0578 & 0.9983 & 4 \\ 1 & 0 & 0 & 4 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

□ $_3$ = (1.5 3 2)′:

□ $_1$ = 7.4228□ $_2$ = 112.0243

$$ {}^{0}T_{4} = \begin{bmatrix} 0 & 0.8708 & -0.4916 & 1.5 \\ 0 & 0.4916 & 0.8708 & 3 \\ 1 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

2.3.4 a circular trajectory

To make the arm tip draw a circle on piece of paper, we need two parameters, a diameter and center coordinate (we choose them freely as long as they are within the work envelope), we will use those as inputs for an equation gives us the coordinate (x,y) of each point from the circle as outputs .

Figure 17: Mathematical model of circular trajectory

Matlab simulation

We create a function that compute all possible coordinates of each point from the circle and store them on their appropriate workspace variable, then we pass those coordinates to inverse kinematic function to place the arm tip on them.

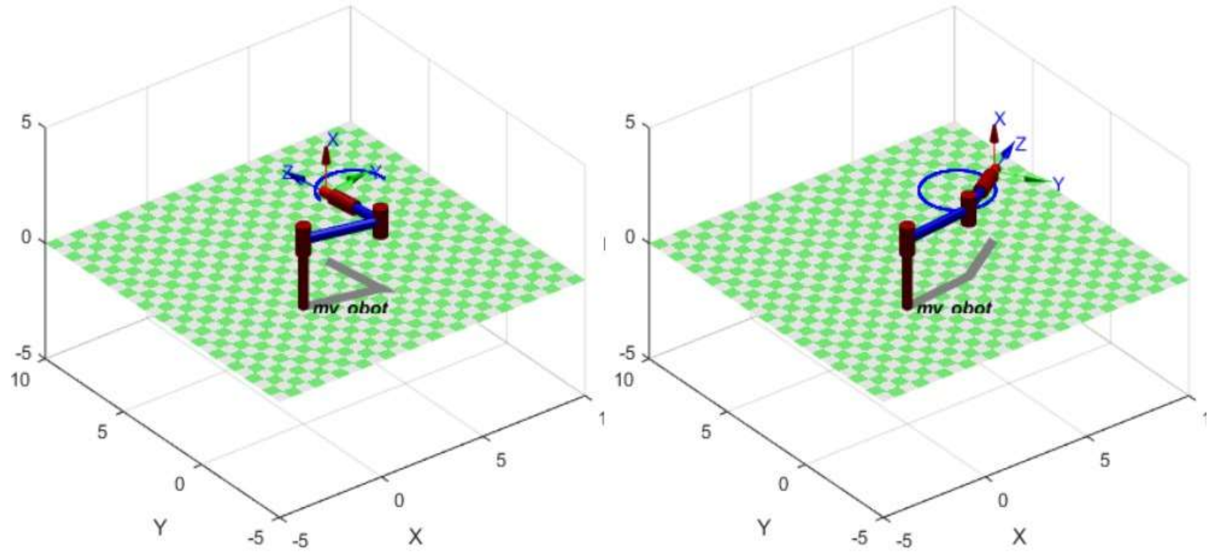

Figure 18: Simulation of SCARA type robot drawing a circle

Chapter 3: building a SCARA type arm manipulator

- ☐ Robot structure.
- ☐ A simple pick and place algorithm.

3.1 Arm manipulator components

In order to interact with the outside world without human intervention or with limited guidance any robot must have at least one sensor to detect events and make decisions upon it and it can’t achieve any task without an actuator which converts the power source to a mechanical motion. Finally it needs any type of a controller to decide what's to do next, when and how to implement it.

3.1.1 Actuators

We just saw early on chapter 2 that our arm manipulator configuration consists of:

A prismatic joint:

We will use a linear actuator to make it, a mechanism that converts rotational motion of a DC motor to linear motion that lifts the arm on the z axis like it seems in the following figure:

Figure 19: The working principle of linear actuator

Because of it’s open loop controller, stepper motor doesn't need any sensor for position feedback, this means we could spare some space for something else. DC motor with encoder also fit well for this specific application, its feedback measures allow us to reach our desired altitude with precision.

For some reasons we get only the dc motor, by running a dozen of tests we found a linear relationship between how much we keep the motor on and where the arm will be, If we suppose that the gripper holds nothing and we run the motor on clockwise direction for 1 second the arm will rise by 2.5 cm else (run the motor counter clockwise) the arm will go down by 3 cm. This is not a efficient way to get things done because we need to run same tests for every different weight we need to pick up.

Figure 20: Translational displacement of arm tip for 1 second

Now, that we know how much should we run the motor to get the desired altitude we need to know in which direction should we rotate it?

To do so, we always set the arm on initial altitude we called d_initial let's say it equals to 10, now we want to go to the desired altitude, then we subtract d-initial from the desired altitude if the difference was positive then we will rotate in clockwise direction else the opposite.

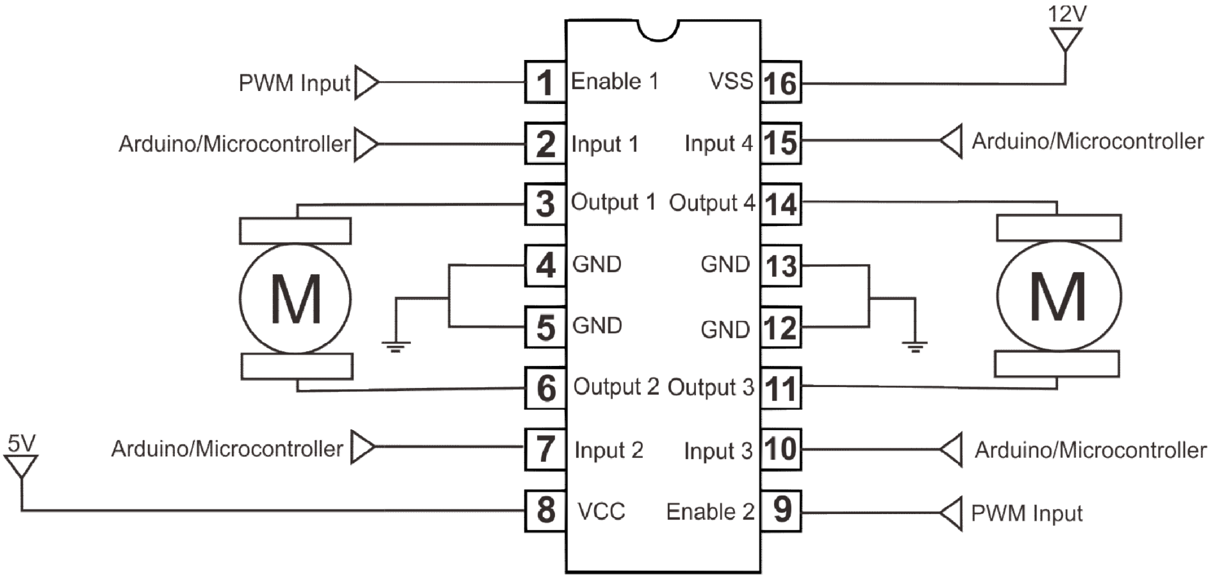

Figure 21: L293D ship to control the rotation speed and direction of the DC motor

Two revolute joints:

Flexibility is the main advantage if you choose the wood to make your mechanical parts, it is so easy to handle and to work with, so you can cut the shape you want with the dimension you need.

Figure 22: A side view of the two revolute joints with servo motors

3.1.2 Sensors [17]

To achieve any task we need to position arm manipulator end effector tool on the desired position, to make it right we need sensors to detect targets coordinates.

Pixy, Robot vision made easy [18]
Using computer vision algorithms almost all targets recognition problems became relatively easy, Pixy is a fast vision sensor with powerful processor that could detect 7 color signature and track hundred of objects simultaneously.

Figure 23: The second version of Pixy

The key feature here is the ability of pixy to delivers only the kind of data you care about like targets coordinates in this case. Pixy support all popular controller platforms.

A none vision solution to detect objects coordinate

Computer vision solution is quite expensive, to keep the cost under the budget we came up with low cost simple solution, proximity distance sensor attached to a servo motor which keep rotating until the sensor detect an object, at this point we have two viable information, an angle and a distance which are also known as polar coordinate.

Figure 24: Illustration explain the working principle

Unfortunately this application suffer from many drawbacks like it is unable to identify the nature of the target or track it, it can't detect anything located 10 cm away from him in case we used sharp distance sensors.

3.1.3 Controller

its ready to use GPIO and built-in functions that spare us the trouble of reading a datasheet and setting every bit we need to run the desired peripherals make the Arduino platform a perfect choice to deliver a prototype as fast as possible.

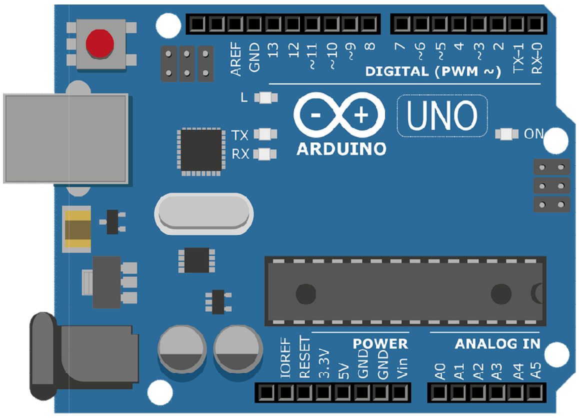

Figure 25: Arduino Uno

[redacted]

Figure 26: a real picture of our arm manipulator

3.2 Visual diagram for pick and place algorithm

Figure 27: A detailed diagram explains how the arm picks up targets and sets them

General conclusion:

Practical results almost identical with theoretical results because of vibrations, frictions and noun ideal actuators which always prevents the actual output one step behind the desired output.

While matlab is great way to study robotics kinematics and dynamics, model and simulate all other robotic subsystems like power circuit design, it was never enough to build an actual arm manipulator, we need to improve our embedded electronics skills and knowledge and get better on using manufacturing tools both manually or using special machines (CNC, laser cutter, 3D printers,,,) .

There is always limited time, effort and budget you could spent so make sure that your feasibility study worth the troubles.because sometimes you found yourself obliged to take risks or sacrifices to catch up with project delivery date.

References:

[1] Zunt, Dominik. "Who did actually invent the word "robot" and what does it mean?". The Karel Čapek website. Archived from the original on 2013-01-23. Retrieved 2017-02-05.
https://web.archive.org/web/20130123023343/http://capek.misto.cz/english/robot.html

[2] Asimov, Isaac (1996) [1995]. "The Robot Chronicles". *Gold*. London: Voyager. pp. 224–225. ISBN 0-00-648202-3.

[3] Owen Holland. "The Grey Walter Online Archive". Archived from the original on 9 October 2008. Retrieved 25 September 2008.
https://web.archive.org/web/20081009055230/http://www.ias.uwe.ac.uk/Robots/gwonline/gwonline.html

[4] Pearce, Jeremy. "George C. Devol, Inventor of Robot Arm, Dies at 99", *The New York Times*, August 15, 2011. Retrieved February 7, 2012. "In 1961.
https://www.nytimes.com/2011/08/16/business/george-devol-developer-of-robot-arm-dies-at-99.html

[5] "National Inventors Hall of Fame 2011 Inductee". Invent Now. Retrieved 2011-03-18.
http://invent.org/inductee-detail/?IID=426

[6] "Computer History Museum – Timeline of Computer History". Retrieved 30 August 2007.
http://www.computerhistory.org/timeline/?category=rai

[7] "The Robot Hall of Fame : AIBO". Archived from the original on 5 September 2018. Retrieved 31 August 2007.
https://web.archive.org/web/20070906164457/http://www.robothalloffame.org/06inductees/scara.html

[8] https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-2:v1:en Archived 2018-09-5 at the Wayback Machine.

[9] KUKA-Roboter.de: 1973 The First KUKA Robot Archived 2009-02-20 at the Wayback Machine. English, 28th of March 2010
http://www.kuka-robotics.com/en/company/group/milestones/1973.htm

[10] "History of Industrial Robots". Archived from the original on 2012-12-24. Retrieved 2012-10-27.
https://web.archive.org/web/20121224213437/http://www.ifr.org/uploads/media/History_of_Industrial_Robots_online_brochure_by_IFR_2012.pdf

[11] https://en.wikibooks.org/wiki/Robotics/Types_of_Robots/Arms, Archived 2018-09-5.

[12] https://www.cnet.com/news/robotic-kitchen-will-be-your-own-personal-chef/ ,Archived 2018-07-5.

[**13]**https://www.weforum.org/agenda/2016/05/introducing-the-cobot-humans-and-robots-working-together/,Archived 2018-07-5.

[14] Peter corke**,**"Robotics Toolbox for Matlab " book.

[15] Paul, Richard (1981). *Robot manipulators: mathematics, programming, and control : the computer control of robot manipulators*. Cambridge, MA: MIT Press. ISBN 978-0-262-16082-7.

[16] Spong, Mark W.; Vidyasagar, M. (1989). *Robot Dynamics and Control*. New York: John Wiley & Sons. ISBN 9780471503521.

[17] Kazed Boualem ”robotics” lecture notes, Saad Dahlab University.

[18] https://pixycam.com/,Archived 2018-07-5, Archived 2018-07-5.