HAN_UNIVERSITY OF APPLIED SCIENCES

HAN Master Major Project

Major Project Report

Using FANUC R-2000iC/210F (6-axis robot) for improved efficiency in FRC parts formation

Student Number: 617931

Name: Karl Wallkum

Track: Master Control Systems Engineering

Company: HAN Smart Production Cell (IPKW)

Supervisors: Nguyen Trung(Company Supervisor)
Wesselingh Ellen(HAN Supervisor)

Date: 04/03/2020

Nguyen Trung

Wesselingh Ellen

STATEMENT OF OWN WORK

For exams (computer-based or otherwise)/assignments/reports/theses

Master: ☐ Automotive Systems ☒ Control Systems Engineering
Date of submission: 04.05.2020
Manner of submission: Paper / Dropbox / Email Paper / Email
Name, student number and phone number: Karl Wallkum
617931
+49 157 7154 5726
karl.wallkum@yahoo.de
Signature(s):

By signing this form, I/we declare that the above-mentioned exam/assignment/report/thesis that I/we have submitted (hereafter referred to as the 'document') was independently created by me/us without any external help and that I/we am/are aware of the rules concerning irregularities/fraud as set out in the degree statute.

Parts of the document that are literally or almost literally cited from external sources (such as internet, books, journals etc.) have been referenced by me/us according to the APA norm, for example, or preferably the Elsevier norm (e.g. footnote) in the cited part of the text (in italics).

10 June 2015

Figure 1: FANUC R-2000iC/210F 6-axis industrial robot arm

1. Summary

Contents

2. Glossary

Numerical Solution Originally the term numerical solution describes a solution given in terms of numbers instead of an explicit closed form expression. In this work, the term ”numerical solution” refers to numerical, iterative methods for solving the inverse kinematics problem by using a sequence of steps leading to incrementally better solutions for the joint angles. Examples for these methdods are the jacobian inversion method, the optimization based method cyclic coordinate descent as shown by Lukas Barinka [6].. 5

3. Acronyms

DCS Dual Check Safety. 2

DDS Data Distribution System. 2, 4

DH Denavit-Hartenberg. 6, 11–13, 24

DOF degrees of freedom. iv, 1, 4, 5, 14, 19, 20, 24

DT Digital Twin. 5, 6

EOAT End Of Arm Tooling. 8–10, 14, 19

ETCS European Train Control System. 6

fig figure. 2

FRC Fibre Reinforced Composites. 2, 5

OS Operating System. 4

PC Personal Computer. 2

PL Production Line. 6

ROS Robot Operating System. 2, 4

SPC Smart Production Cell. 5

4. Introduction

4.1 Abstract

This work aims to integrate a FANUC 210F 6 axis industrial robot arm into an experimental production line. As this production line is set in a research environment, gaining a deeper understanding of all involved systems is desired.

The dynamic behaviour of a physical system is best expressed with an analytical model. In order to control a robot arm, a kinematic model needs to be created. By attaching inertias and external forces, a dynamic model is obtained. With this model, a control algorithm can be derived.

The objective of this thesis is to derive the complete inverse kinematic model of a 6 DOF robotic arm analytically. For an exact numerical simulation of the device most steps are laid out theoretically and difficulties in the practical implementation are described. Additionally for follow up projects this work also contains a quick start guide and a safety manual for the robot in this setting . Finally to contribute to current research, twinning specifications will be defined.

4.2 Preface

Definition of a Robot Robots [34] can be defined as programmable movement automatons [4] that can perform tasks without human supervision and can be taught at least repetitive tasks. Increasingly, also ways to sense their surroundings are added and improve their movements according to the situation [51]. These sensors are placed additionally to the standard feedback-control sensors like pulse encoders.

Origin of Robots The origin of Robots is well documented in "Robotics, Vision and Control: Fundamental Algorithms in MATLAB" by Peter Corke [9]: "The term robot first appeared in a 1920 Czech science fiction play Rossum's Universal Robots" by Karel .apek (pronounced Chapek). The term was coined by his brother Josef, and in the Czech language means serf labor but colloquially means hardwork or drudgery. The robots in the play were artificial people or androids and as in so many robot stories that follow this one, the robots rebel and it ends badly for humanity. Isaac Asimov's robot series, comprising many books and short stories written between 1950 and 1985, explored issues of human and robot interaction and morality. The robots in these stories are equipped with "positronic brains" in which the "Three laws of robotics" are encoded. These stories have influenced subsequent books and movies which in turn have shaped the public perception of what robots are. The mid twentieth century also saw the advent of the field of cybernetics – an uncommon term today but then an exciting science at the frontiers of understanding life and creating intelligent machines. The first patent for what we would now consider a robot was filed in 1954 by George C. Devol and issued in 1961. The device comprised a mechanical arm with a gripper that was mounted on a track and the sequence of motions was encoded as magnetic patterns stored on a rotating drum. The first robotics company, Unimation, was founded by Devol and Joseph Engelberger in 1956 and their first industrial robot shown in Fig. **??** was installed in 1961. [9]

Simple robots I have started working with robots and robotic systems in my bachelor studies. As a starting engineer, I was exploring the possibilities of automated manufacturing with CNC mills and 3D printers. These were very simplistic robotic systems based on a feedforward control with stepper motors for position control. For starting a production process, these devices had to be half automatically calibrated and the position and orientation needed to be taught by pointing the drill/printing head to the markerpoints. As no feedback control is usually available on these devices, loss of step control and other errors are possible and can go unnoticed until the end of production, when the part is inspected.

Current Robots Highly developed automated platforms like the Fanuc 210F have feedback-sensors and -control for point to point as well as tracking motion, computer vision as seen on the deltarobot, digital and analog I/O, builtin- safety like the Dual Check Safety (DCS) motion safety system and high speed fieldbus connectivity like Profient. With a teach pendant, these Robots can be configured and programmed. Alternatively Personal Computer (PC)-based proprietary manufacturer-specific software with a model based test cell can be used to program the robot.

Future robots To compete in the future developements of "Smart Manufacturing", "the Smart Cell project investigates on an automated production cell for the realization of structural composite components which must be produced in large scale or mass produced" (see [43], Smart-Cell project description). Besides research in materials and production, the automation environment system is of interest for this research. To manage the interaction between material but also the machines among each other, a Data Distribution System (DDS) couples the subsystems as a middlware as seen in figure (fig). 4.1. A middleware is "glue code" that helps to interface systems with each other to exchange commands and data to use it as a single robotic system. Robot Operating System (ROS) is a proposed middleware to model, simulate, and control the production line as a homogenious distributed system. Besides the classical publish subscribe pattern to simplify network programming, ROS provides an ecosystem of interchangeable nodes and drivers to provide a wide codebase for complex tasks.

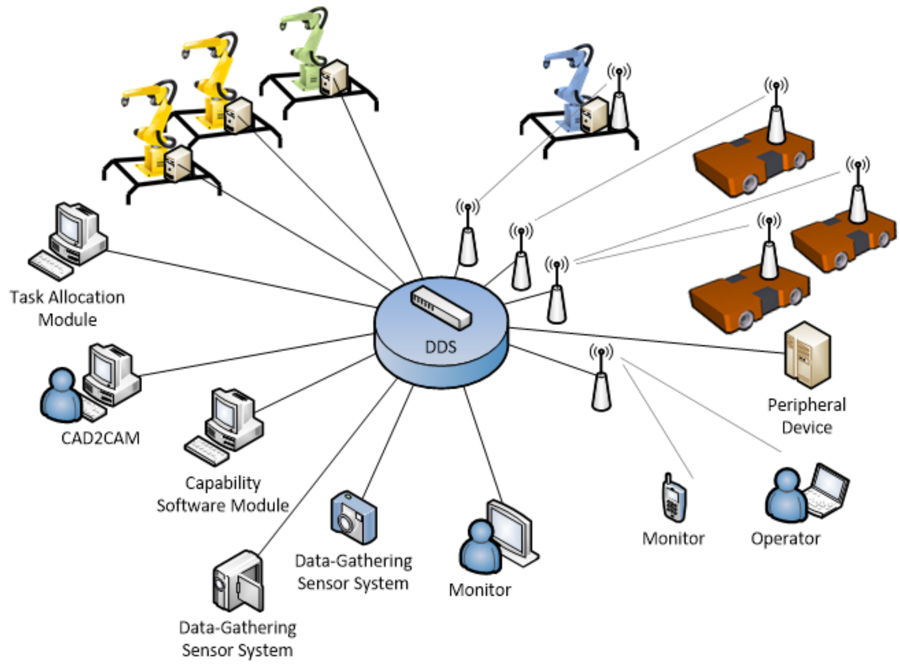

Figure 4.1: DDS as central information exchange platform

4.3 Problem Definition

For Fibre Reinforced Composites (FRC) part production, a robot arm can be used to load the press with raw material, as it allows for more flexibility in the production line than specialized low DOF pick and place systems. As the robot arm has many degrees of freedom, there are different strategies for a control cycle. There are two contratiction main constraints that need to balanced: On one hand, the movement of materials has to be achieved as fast as possible to minimize the cool-down of the molten FRC. On the other hand, the accelerations and forces on it should be minimized while transferring, to make sure no material is lost in the transfer process which would lead to deformations on the final product. This makes it hard to find an ideal, fast control strategy to place the raw material into the press.

5. Literature survey

In the project plan, it was stated, that the master level will be demonstrated by understanding and simulating the dynamics of a 6 axis Robot arm." (see [49], sect. Master Level) This should be done by creating a model of the robot arm. This model can then be used to create a controller. To create the model of the robot arm, a literature review is necessary to lay out the best approach.

Additionally to the modelling and control of the robot arm, digital twinning became a research topic within the project. A review of existing literature on digital twinning will be done as well.

5.1 Field of study

To start the literature review, a set of first keywords was needed. Through an expert interview with my company supervisor Trung Nguyen [29] , who had already supervised other thesis projects in the domain of robotics, a list of keywords to start with was found in a quick discussion. Not all of these keywords were immediately clear, so it was necessary to find definitions for these. With the help of scientific databases and search engines, sources for these definitions could be found.

Keyword	Description (Source with Search Engine)
6 axis robot	serial 6 degree of freedom robots ([48] with HANQuest)
industrial robot arm	some form of jointed structure achieved by the linking of a number of rotary and/or linear motions or joints([53] with Science Direct)
inverse kinematics	Determination of joint variables in terms of the end-effector position ([21] with Springer Link search)
Peter Corke robotics toolbox	Matlab toolbox for the study and simulation of classical arm-type robotics, for example such things as kinematics, dynamics, and trajectory generation ([38] with Google, yahoo, duckduckGo)
motion planning	find a sequence of valid configurations that moves the robot from the source to destination ([26] with Google search)
robot dynamics	relationship between the forces acting on a robot mechanism and the accelerations they produce ([16], with Scholarpedia)
ROS	Robot Operating System - framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behaviour across a wide variety of robotic platforms. (in [36], About ROS, with Google ,yahoo, duckduckGo)

As seen in "Implementation of Robot Systems" [53], the FANUC 210F is an articulated robot arm, also called a jointed arm. It is a 6 axis robot that has six rotational joints, each mounted on the previous link. This type of robot has the ability to reach a point within the working envelope in more than one configuration or position with its final joint,. As there are multiple configurations possible to reach the same position, path planning would become an important topic. This means through inverse kinematics the motion of the joints needs to be determined without considering the local forces that cause them to move. As stated in the project plan, MATLAB is thought to be used. The robotics toolbox by Peter Corke was seen as a good tool for simulating these kinematics in MATLAB. When attaching the dynamics to this model, further simulations could be made to simulate the dynamic bahaviour of the robot arm and create a controller. "ROS is a flexible framework for writing robot software." as seen on the "about ROS" page of the ROS-project. As pointed out by other engineers on the "ROS Answers" page, a project related Forum [44], the role of ROS is not clearly defined. ROS shares characteristics with middleware, frameworks, but also has Operating System (OS)-like features. ROS can take the Role of a DDS, taking a central role in coordinating tasks and information on a distributed system of nodes. ROS can be used as a platform to create a digital twin.

5.2 In depth literature

As stated in the online manual of MathWorks, "Kinematics is the study of motion without considering the cause of the motion, such as forces and torques." [31] Inverse Kinematics, also called backward kinematics is the logical opposite to forward kinematics also called direct kinematics (See figure 5.1).

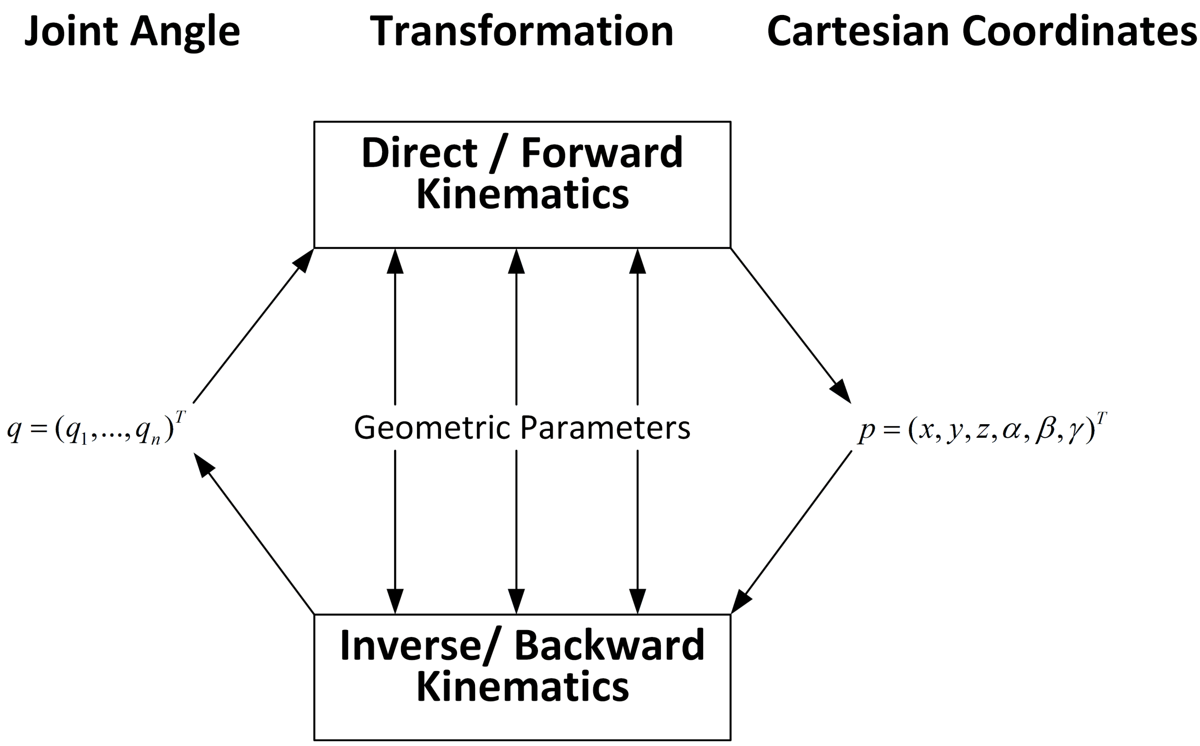

Figure 5.1: Relation between inverse and forward kinematics translated from German Wikipedia site on inverse kinematics [31], for content see Mathworks [35]

The forward kinematic analysis can be obained as seen in "Forward Kinematic Analysis of an Industrial Robot " [8]
In the german wikipedia article on inverse kinematics [19], the thesis "Verallgemeinerte inverse Kinematik für Anwendungen in der Robotersimulation und der virtuellen Realität" [40] is given as one of the main sources. This thesis work gives a good overview on the topic of inverse kinematics. As stated in this thesis work, the Denavit-Hartenberg notation is a robotic convention to map the local coordinate systems within a kinematic chain as found in robot arms.
As seen in "A Mathematical Introduction to Robotic Manipulation" [33], chapter 2.2 (obtained through Semantic Scholar, search phrase: robotic convention), there are several other conventions besides the "Denavit-Hartenberg notation" used in the robotics research field like the the "product of exponentials formulation". As most textbooks prefer a Denavit-Hartenberg fomulation of the kinematics (see [33], ch. 3.1 Manipulation using single robots), this convention will be chosen in this work as well.
Solutions for forward kinematics are simple to obtain but solving inverse kinematics has been one of the main concerns in robot kinematics research. With more DOF, solutions get more complex as non-linear equations with transcendental functions need to be solved. For this set of equations, no general algorithms are available. Often algebraic, geometric and iterative methods for complex manipulators are used to find a solution to the inverse kinematic problem as stated by Tarun Pratap Singh et al. in the abstract of "Forward and Inverse Kinematic Analysis of Robotic Manipulators" [42].

To find a suitable method for solving the inverse kinematic problem, a definition for the solution is needed:

> "A manipulator will be considered solvable if the joint variables can be determined by an algorithm that allows one to determine all sets of joint variables associated with a given position and orientation. [...] The algorithm should find all possible solutions" -Dr.-Ing. John Nassour [28]

With this definition of solvability, all systems with revolute and prismatic joints with 6 DOF in a single series chain are solvable with the current available research. [28]

As the goal is to find a suitable solution strategy for the inverse kinematic problem, it helps to map out the different types of methods. Solution strategies can be split into two classes as stated by Dr.-Ing. John Nassour in his presentation [28]:

Closed-form solutions	**Numerical Solution**
faster because analytical method	slower because of iterative nature
will find all solutions	not always find all solutions
Two approaches:	
algebraic approach **geometric approach**	

As Numerical Solution cannot always deliver all solutions and solve within an unknown number of operations as well as depend on the users decision for accuracy, [28] a closed-form solution will be preferred.

On HAN Quest, with the keywords "6DOF inverse kinematics" the article "Inverse Kinematics Solution and Optimization of 6DOF Handling Robot" [54] can be found. This offers an algebraic method to solve the inverse kinematic problem for 6 axis robots. A similar approach is shown in "Forward and Inverse Kinematic Analysis of Robotic Manipulators" [42]. Another approach can be found in "Forward and Inverse Kinematics Model for Robotic Welding Process Using KR-16KS KUKA Robot" [11]

As an alternative, a geometric modelling can be done as seen in "Workspace analysis and geometric modeling of 6 DOF Fanuc 200IC " [7]. A completely different approach is given in "A inverse kinematic solution of a 6-DOF industrial robot using ANN" by using artificial neural networks [2].
With one of these methods, a solution can be found for the inverse kinematic problem. This solution can then be verified with the robotics toolbox created by Corke [38].
With this solution, a dynamic model of the robot can be created by attaching dynamics to the model as seen in "Control and Safety Mechanisms for a 3 DOF Manipulator with Human Interaction" [50] and "A mathematical introduction to Robotic Manipulation" [33]. A complete example of a dynamic simulation of a 6 DOF robot arm can be found in "Dynamic Multibody Simulation of a 6-DOF Robotic Arm" [25]
A controller for trajectory tracking can then be created with the model as seen in "Experimental Evaluation of Nonlinear Feedback and Feedforward Control Schemes for Manipulators" [22]
This controller could then be plugged into the real robot as stated in the "Control of a FANUC Robotic arm using MATLAB manual" [46] An example of this can be found in "Modelling and analysis of a 6 DOF robotic arm manipulator" [20]

5.3 Digital twinning

In the setup of the experimental production line for FRC at the Smart Production Cell (SPC) Digital Twin (DT) has been chosen as one of the key technologies for “Smart manufacturing”. In the cooperation with Qing, there have been a few discrepancies on the understanding of a DT.

To approach this topic, the term digital twin has to be defined. This Concept was introduced for the first time by Grieves at one of his presentations about Product Lifecycle Management in 2003 at University of Michigan as a virtual representation of what has been produced [18]. As stated by Qinglin Qi [32], a DT is a high fidelity virtual model for physical objects in a digital way to simulate their behaviour. When looking on the Wikipedia page on digital twinning [12], a wide variety of definitions can be found. This shows, that the concept of the DT is not yet clearly defined. All of these definitions have in common though, that DTs are "digital replications of living as well as nonliving entities that enable data to be seamlessly transmitted between the physical and virtual worlds" [13].

To approach the problem form the other side, it might help to look at, what is not a digital twin. A virtual representation of a physical object without any exchange of data is a digital model [52]. If data is fed from the physical system into the model, a digital shadow is created [23]. Only a “bi-directional relation between a physical artefact and the set of its virtual models” [39] fulfils Schleich’s vision of a DT.

As an example, a computer aided design (CAD) model is a representation of a physical entity and it is typically used to describe the shape, dimensions and materials of a construction. This model can be a 2D or a 3D model. Only if the latest sensor data associated with a matching physical device is fed into this CAD model, it can be considered a DS. [52] By simulating different scenarios in a model and representing the current state of the system the DS turns into a digital twin, if decisions are made automatically fed back through an actutator into the physical entity [39]. An implementation of this data transmission can be found in ”Sensor Data Transmission from a Physical Twin to a Digital Twin” [1].

As the DT is mostly used in the context of production lines, it makes sense to look for similar examples in other industries, that have undergone comparable transformations with the beginning of the digital age. Such a comparable system can be found in the transport industry. Production Lines (PLs) have several points in common with train networks. One of the latest internationally standardized systems in rail infrastructure is European Train Control System (ETCS) which will probably become the standard signalling and control component in all European countries. A great overview on the ETCS-standard is given by Thales on their website [14]

6. Setup of the physical Environment

7. Building of a Model

7.1 Forward Kinematics

To model the movements of the end effector of a kinematic chain, a forward kinematic analysis is needed.

7.1.1 Approach

A serial-link manipulator consists of links in a chain connected by joints. For describing the serial-link mechanism geometry the Denavit Hartenberg notation is one possible approach. It gives a description of a manipulator for kinematic solutions, Jacobians, dynamics, motion planning and simulation. This description can be obtained through a five step algorithm:[8]

1. Numbering of joints and links
2. Set up local coordinate reference frames
3. Establishing Denavit-Hartenberg (DH) parameters for each link
4. Calculate homogeneous transformation matrices
5. Compute Forward Kinematic Equations

Numbering of joints and links

A serial link robot with n joints has $n + 1$ links.

numbering of links The numbering scheme for links starts at (0) with the fixed grounded base and then increases sequentially up to (n) for the end effector.

numbering of joints The numbering scheme for joints starts at (1) with the joint connecting the first movable link to the base and then increases sequentially up to n.

relation between links and joints Link (i) is connected to its

- lower link $(i - 1)$ at its proximal [27] end by joint (i)
- upper link $(i + 1)$ at its distal [27] end by joint $(i + 1)$

Joint and link numbering in Fanuc 210F This numbering scheme can be applied to the Fanuc 210F (see 7.1)

Figure 7.1: Links (turquoise) and joints (orange) in the FANUC 210F

Set up local coordinate reference frames

With the Denavit-Hartenberg convention, local coordinate frames can be attached to the far end of the links (i) and their accompanying joints $(i + 1)$. Each link (i) is described relative to the pose of the preceding link.

Assignation of z_i axes With the DH-notation, the z_i axes are assigned to link (i). By using the DH convention, both prismatic and revolute joints can be treated similarly:

revolute: z_i is the axis of revolution of joint $(i + 1)$

prismatic: z_i is the axis of translation of joint $(i + 1)$

This means, that joint z_i turns around axis z_i.

Direction of rotation With the direction of the z_i-axis, the direction of positive rotation around joint $(i+1)$ is also given by the right hand rule (see [37] at 10:35). This means, the direction of positive rotation is counter-clockwise around the z_i-axis. With predictive choice of the direction of positive rotation, the direction of the z-axis can be chosen in order to minimize the DH-parameters (insert source here)

z_i axes in Fanuc 210F As described above, the z_i axes can be attached to the Fanuc 210F (see figure 7.2).

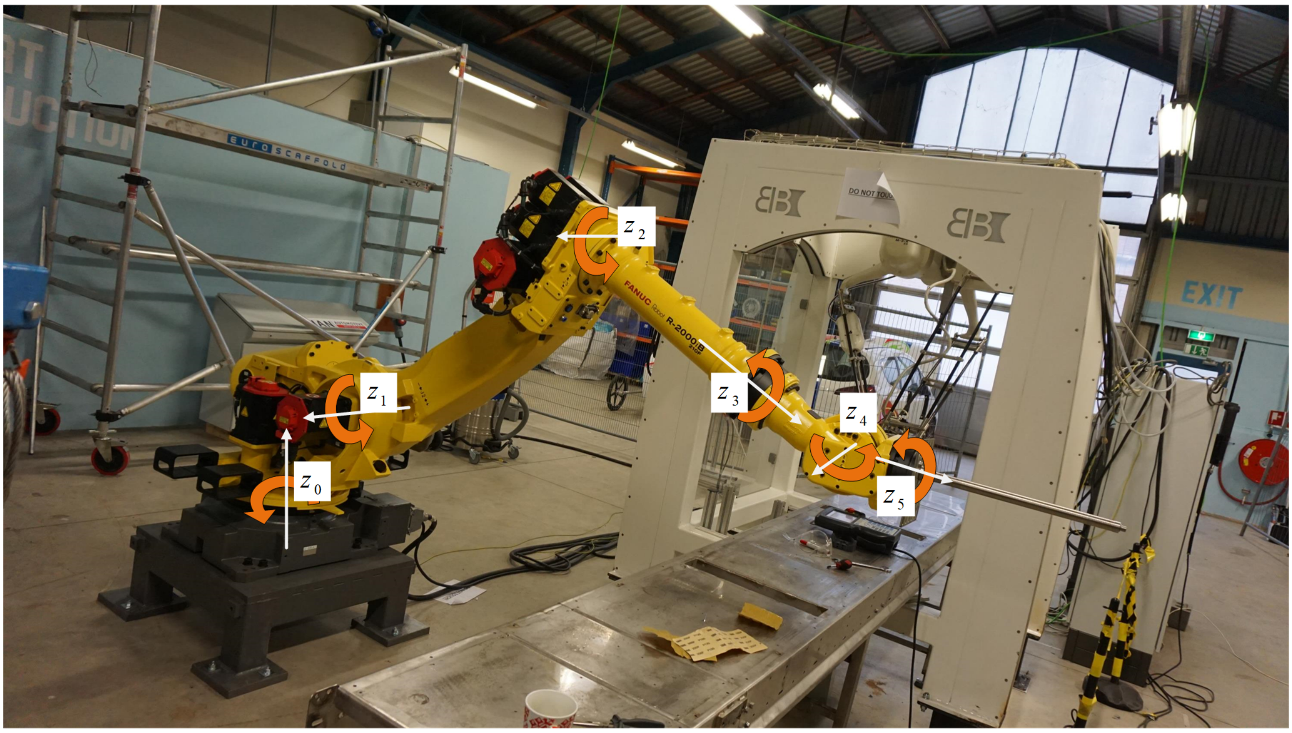

Figure 7.2: z_i axes on the Fanuc 210F with the direction of positive rotation (orange)

The positive direction of the triples z_1, z_2, z_4 and z_0, z_3, z_5, was chosen to make sure, the x-axis would always have the same direction for parallel joints.

Base frame The base frame (0) can be chosen nearly arbitrarily. The origin of the base frame can be any point on z_0. For simplicity, the origin of frame(0) can be put into joint(1). Usually, the x-axis of the base frame is chosen, so that it points in the direction of the End Of Arm Tooling (EOAT) in default position according to its base, [55] but can be chosen in any convenient manner [41].

Assignation of frames (i) Starting from frame (1) in an iterative process, frame (i) can be set up using frame $(i-1)$.
Three cases regarding the relationship of axes z_{i-1} and z_i need to be considered when setting up frames:

Non coplanar: Axes don't intersect and are not parallel. Line containing the common normal z_{i-1} to z_i defines x_i-axis and the point of intersection with z_i is the origin of frame (i)

a) find common normal of the joint axes

b) put origin in intersection of normal with joint axis

c) put z_i axis in the joint axis

d) x_i points in direction of the common normal, facing away from frame (0)

e) add y_i according to right hand rule

Parallel: Axes are parallel. Line containing the normal through origin of frame $(i+1)$ and the z_i-axis defines the x_i-axis which is directed from origin of frame (i) toward the distal joint. The point of intersection of common normal with z_i-axis gives the origin of frame (i).

a) look for normal originating from distal joint

b) put origin in the intersection of the normal with the joint axis

c) put the z_i axis in the joint axis of the link $(i+1)$

d) x_i points follows the common normal with the distal joint, pointing towards the distal joint

e) add y_i according to right hand rule

Intersecting: Axes are intersecting. Line containing the normal to the plane formed by axes z_{i-1} and z_i gives the x_i-axis with positive direction chosen arbitrarily. Point of intersection of axes z_{i-1} and z_i is the origin of frame (i).

a) put origin in intersection point of the axes

b) put the z_i axis in the joint axis of the link $(i+1)$

c) x_i is perpendicular to both joint axes

d) add y_i according to right hand rule

Assignation of EOAT frame As there is no distal joint for the EOAT frame, the steps for this frame are different:

a) put origin on axis of proximal joint

b) axis z_i follows direction of z_{i-1}

c) x_i can be chosen arbitrarily, but is usually determined by screw holes

d) add y_i according to right hand rule

local coordinate reference frames on the Fanuc 210F As described above, the local coordinate reference frames can also be attached to the Fanuc 210F as seen in 7.3.

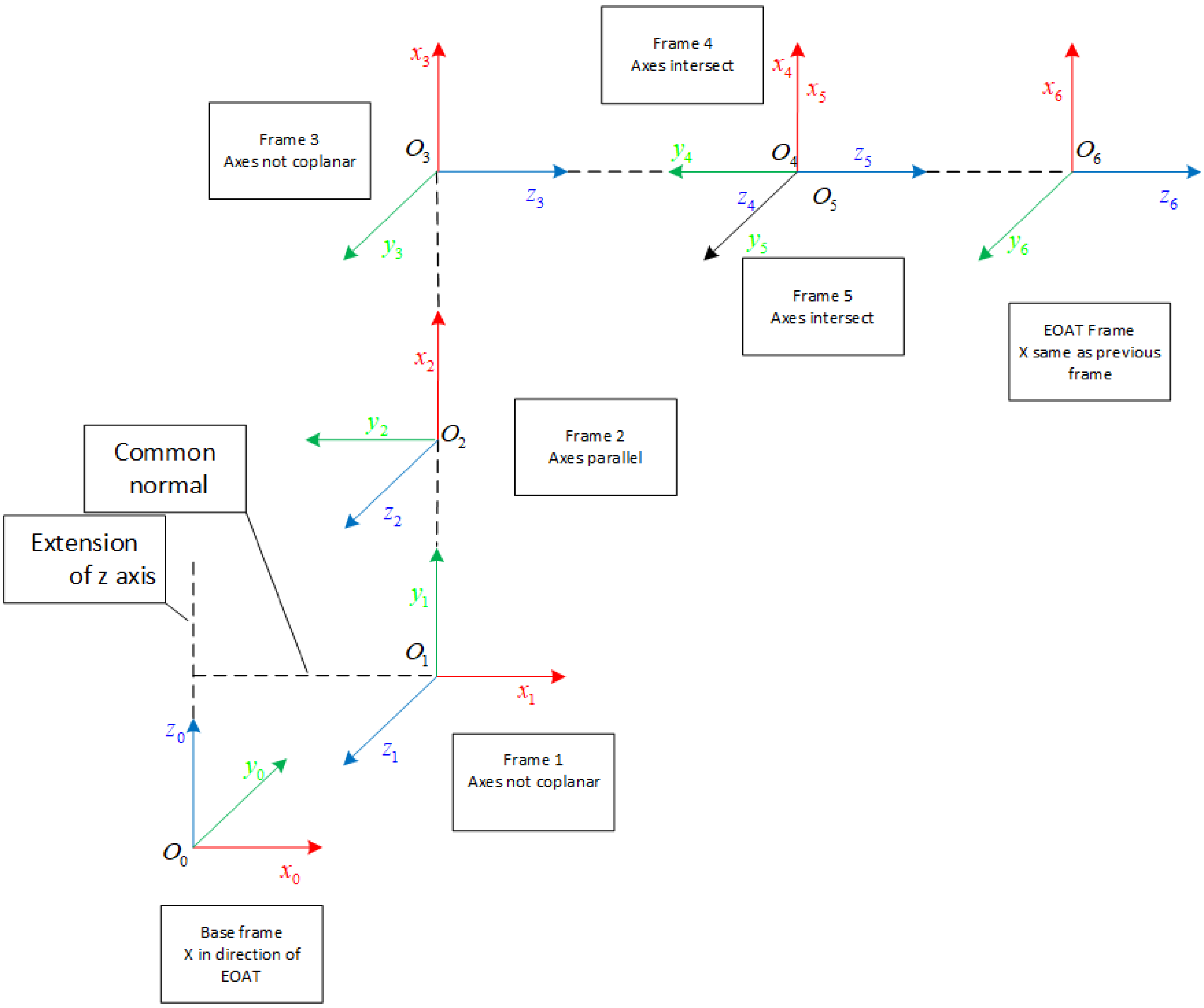

Figure 7.3: Coordinate reference frames for Fanuc 210F

frame 0 (Base) The frame mapping starts with the base frame. For the base frame the z-axis is given through joint j_1. Origin of frame 0 is put into joint 1. The x-axis is chosen to point in direction of of the EOAT in standard pose. For standard pose see 7.4. This pose will also be chosen for alignment of all distal frames.

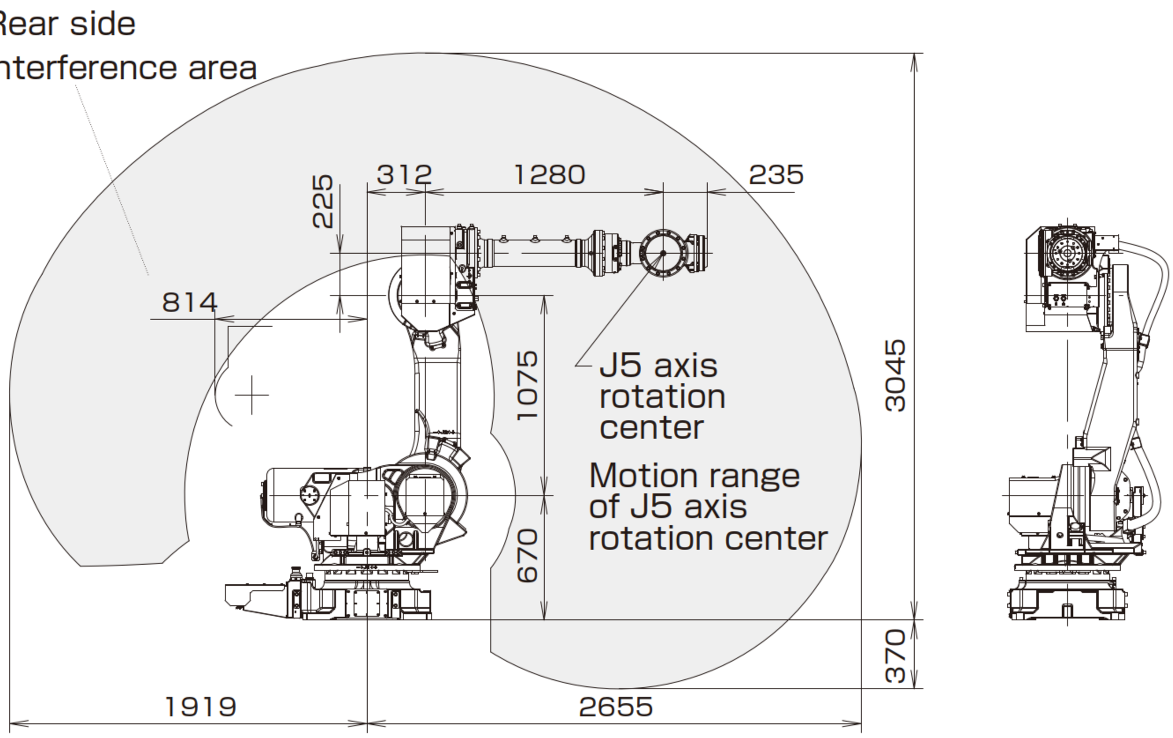

Figure 7.4: Standard pose of the robot

frame 1 For frame 1, the z-axes are not coplanar. Because of that, the z_0-axis is extended until a common normal can be found that intersects with z_i to define the origin O_1. x_1 departs from O_1 along the common normal. y_1 is added according to the right hand rule.

frame 2 As seen, in image 7.2, axes z_2 and z_1 are parallel. O_2 can be found at the intersection of the normal through O_1 with z_2. x_2 follows the common normal with joint 4. y_2 is added according to the right hand rule.

frame 3 Joint 3 and joint 4 are not exactly aligned (see image 7.4), which prevents the z-axes to intersect. That's why axes z_3 and z_2 are not coplanar. z_3 runs parallel through the link 3, due to the orientation of the rotational axis. As the common normal between z_3 and z_2 defines x_3, which in turn gives O_3, the origin can be found far away from the physical position of the joint. y_3 is added according to the right hand rule.

frame 4 Joint 5 lies in line with joint 4. As z_3 follows this line, the axes z_4 and z_3 intersect in the center of joint 5, which gives O_4. x_4 leaves the plane spanned by z_4 and z_3 perpendicular. The positive direction of x_4 is chosen to be similar as in frame 3 for simplicity. y_4 is added according to the right hand rule and points in direction of frame 3.

frame 5 As z_5 lies in line with O_4, the axes z_5 and z_4 intersect in O_4 which puts O_5 at the same position as O_4. x_5 leaves the plane spanned by z_5 and z_4 perpendicular. The positive direction of x_5 is chosen to be similar as in frame 4 for simplicity, which puts x_5 and x_4 on top of each other. y_5 is added according to the right hand rule and as z_5 is turned 90 degrees relative to z_4 around $x_{4,5}$, y_5 is turned 90 degrees relative to z_4 around $x_{4,5}$ as well.

frame 6 (EOAT) z_6 lies in line with z_5 and is consequently parallel. as there are no distal joints, to reference x_6, it can be chosen arbitrarily. For simplicity, it is chosen to be similar as in frame 5. y_6 is added according to the right hand rule.

Establishing DH parameters for each link

After assigning the local coordinate frames, the DH-Transformations have to be determined. This leads to a kinematic chain with each frame determined by the previous one [55].
Link0 - Link1 - Link2 - Link3 - Link4 - Link5 - Link6
Each DH-transformation consists of four elementary transformations[55]:

1) rotation around the x_i-axis with the amount of α_i

2) translation along x_i-axis with the amount of a_i

3) translation along z_i-axis with the amount of d_i

4) rotation around z_i-axis with the amount of θ_i

This shows that the DH robotic convention is a minimal line representation, as with four parameters, all possible lines in the Euclidean Space can be represented ([5], page 210).
Following from this, two pairs of parameters determine the joints and links [8]:

Links: represented by link length (a) and link twist (α)

Joints: represented by link offset (d) and joint angle (θ)

These are the DH-parameters which form the transformation matrices that connect serial link coordinate frames. They can be distinguished into parameters and variables. While constructive parameters are dependent on the construction of the robot and are constant, variables depend on the joint movement [42] [8] [55].

α_i angle between z_{i-1} and z_i, measured in plane normal to x_i (constructive parameter)

a_i distance between z_{i-1} and z_i, measured along x_i; parallel to $z_{i-1} \times z_i$ for intersecting axes (constructive parameter)

d_i distance from the origin O_{i-1} of frame $i-1$ to the intersection of the x_i axis with z_{i-1}, measured along z_{i-1} (constructive parameter in revolute joints, variable in prismatic joints)

θ_i angle between x_{i-1} and x_i, measured in plane normal to z_{i-1} (constructive parameter in prismatic joints, variable in revolute joints)

These parameters describe the relatively complex transformation of the coordinate system $i-1$ into the coordinate system i by four subsequent elementary transformations [40]:

$$T_{i-1,i} = ROT(z_{i-1}, \theta_i) * TRANS(0,0,d_i)^T * TRANS(a_i,0,0)^T * ROT(x_i, \alpha_i) \tag{7.1}$$

$ROT(v, \alpha)$ is a simple rotation around vector v with the angle α and $TRANS(x,y,z)^T$ is a translational movement along the vector $[x,y,z]^T$. For a visual representation of these parameters see figure 7.5

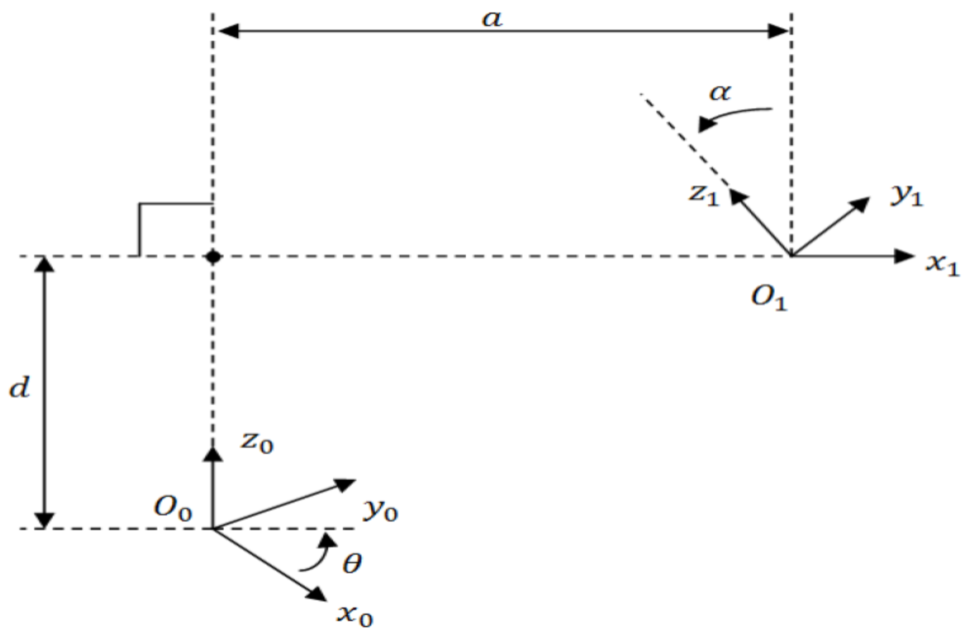

Figure 7.5: Visual representation of a DH-transformation

This visual representation gives a guide to attach the DH-parameters on serial-link mechanisms.

Establishing DH parameters on the Fanuc 210F

As described above, the DH-Parameters can also be determined for the Fanuc 210F. These parameters can be found with the help of the assigned coordinate frames as seen in figure 7.6. For simplicity, only non-zero DH-parameters are shown in this figure.

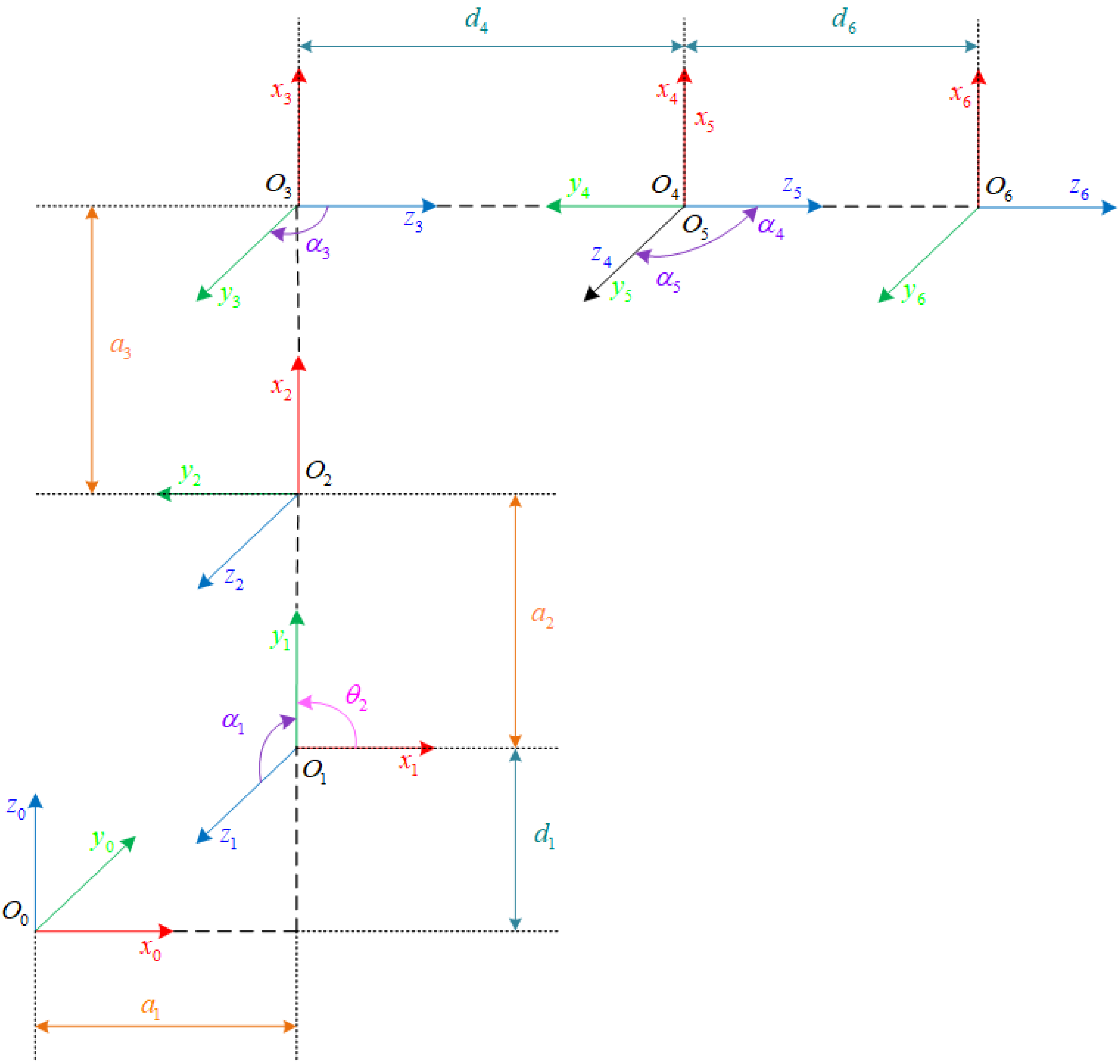

Figure 7.6: Assignment of DH-Parameters on the Fanuc 210F

To wrap this up, a summary of the D-H parameters for each link as follows from figure 7.6 can be found in table 7.1

Link	θ_i	d_i	a_i	α_i
1	θ_1	d_1	a_1	α_1
2	θ_2	0	a_2	0
3	θ_3	0	a_3	α_3
4	θ_4	d_4	0	α_4
5	θ_5	0	0	α_5
6	θ_6	d_6	0	0

Table 7.1: Denavit Hartenberg Parameters for Fanuc 210F

7.2 numeric values of DH-Parameters

As can be seen from table 7.1, many DH-parameters are zero due to a good choice of pose and coordinate frames. All angles are plus or minus 90°.
Most of the frames have translational transformations in either a_i or d_i direction with link 1 and 5 being the exception. By lifting the base frame out of the joint, parameter d_1 could be set to zero as well. For better understanding, the origin of the frame was originally kept inside the joint. Frame 5 shares the origin with frame 4, not requiring any translational movement. Most of the frames are also rotated by 90 degree around the x-axis with link 2 and 6 being the exception. Frame 6 was chosen with the same orientation as frame 5. Frame 2 has the same x-axis orientation as link 1 because joints two and three are parallel. There is just a rotational transformation around the z-axis of 90°. This is only due to the standard pose of the robot, as theta is a variable. For all coordinate frames theta is a variable, as all joints are revolute and none is translational.
Besides the angle values for α_i which were easy to determine from the graphical interpretation in figure 7.6, following numeric values need to be determined through measurements or CAD-Drawings: d_1, d_4, d_6 and a_1, a_2, a_3.
A great help to determine these values is fig 7.4. This technical drawing was obtained from the datasheet for all Fanuc R-2000iB series robots. With this drawing, following DH-parametes can be obtained:

$a_1 =$ 312 $[mm]$

$a_2 =$ 1075 $[mm]$

$a_3 =$ 225 $[mm]$

$d_4 =$ 1280 $[mm]$

$d_6 =$ 235 $[mm]$

The parameter d_1 cannot be determined from this drawing. It can either be measured on the actual robot, or defined as zero, as frame zero can be freely moved along the z-axis. A value of zero would put the origin of the base frame on the same height as frame 1, simplifying subsequent steps. It must be clear though, that all referencing in the forward and backward kinematics will be relative to this point. A change of this point of reference can be added later simply by adding another transformation matrix for translational movement along the z-axis as presented by Richard P. Paul [30].

This results in a table of numeric DH-parameters as seen in 7.2. The transformation ${}^{i-1}T_i$ depends only on a single variable θ_i as all of the other quantities stay constant, and all joints are revolute.

Link	θ_i	d_i	a_i	α_i
1	θ_1	0	312	$\pi/2$
2	θ_2	0	1075	0
3	θ_3	0	225	$\pi/2$
4	θ_4	1280	0	$-\pi/2$
5	θ_5	0	0	$\pi/2$
6	θ_6	235	0	0

Table 7.2: Numeric values of Denavit Hartenberg Parameters for Fanuc 210F

Calculate homogeneous transformation matrices

Locating reference frame $(i-1)$ relative to (i) can be done by executing a series of transformations as given by equation 7.1. For simplification, they can be done with a 4×4 homogenous transformation matrix:

$$ {}^{i-1}T_i(\theta_i, d_i, a_i, \alpha_i) = R_z(\theta_i) * T_z(d_i) * T_x(a_i) * R_x(\alpha_i) \tag{7.2} $$

Equation 7.2 can be expanded into:

$$ {}^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i * \cos\alpha_i & \sin\theta_i * \sin\alpha_i & a_i * \cos\theta_i \\ \sin\theta_i & \cos\theta_i * \cos\alpha_i & -\cos\theta_i * \sin\alpha_i & a_i * \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.3} $$

For further details on how to set up $R_z(\theta_i), T_z(d_i), T_x(a_i) and R_x(\alpha_i)$ and form the transformation matrix in equation 7.3 see chapter one "HOMOGENEOUS TRANSFORMATIONS" in "Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators" [30]. The basic techniques for setting up the translation and rotation matrices as well as how to combine them are described in that chapter .

7.3 Compute Forward Kinematic Equations

In Forward kinematics, the kinematic equations of a robot are used to compute the position of the EOAT from the given joint parameters.

Sum of Transformations Matrix With the transformation matrices from frame (i) to $(i-1)$, a 44 matrix for all joints on n links can be established as in equation 7.4

$$^{0}T_n = \prod_{n}^{i=1} {}^{i-1}T_i \tag{7.4}$$

The resulting matrix has the form (see [54], eq 1):

$$^{0}T_n = \begin{bmatrix} n & o & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.5}$$

short notation The short notations for orientation, rotation and position are defined as:

n normal vector

o orientation vector

a approach vector

p position

Transformation to cartesian coordinates with euler angles These vectors can be transformed into the $[x, y, z, \alpha, \beta, \gamma]$ notation. Vector $p = [p_x, p_y, p_z]$ gives [x, y,z]. The rotational approach $[\alpha, \beta, \gamma]$ can be calculated with the rotation matrix R as seen in eq 7.6.

$$T = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \tag{7.6}$$

The rotation matrix describes the relative orientation of two frames towards each other. As the columns $[n, o, a]$ are the unit vectors along the axes of one frame relative to the other reference frame, the relative orientation of a frame b with respect to a frame a is given by the rotation matrix as seen in eq **??** [8]

$$^{b}_{a}R = \begin{bmatrix} {}_{a}x^{b}) & {}_{a}y^{b}) & {}_{a}z^{b}) \end{bmatrix} = \begin{bmatrix} x^b \cdot x^a & y^b \cdot x^a & z^b \cdot x^a \\ x^b \cdot y^a & y^b \cdot y^a & z^b \cdot y^a \\ x^b \cdot z^a & y^b \cdot z^a & z^b \cdot z^a \end{bmatrix} \tag{7.7}$$

As in equation 7.8 the coordinates relative to the reference frame a, of a point p, of which the coordinates are known with respect to a frame b with the same origin can then be calculated (see [3], "Description of Position and Orientation) .

$${}_{a}p = {}^{b}_{a}R \; {}_{b}p \tag{7.8}$$

7.4 Forward Kinematic Equations on the 210F

As shown in fig.7.2, the Fanuc 210F is a 6-DOF robotic arm with n=six revolute joints in series. Their orientation can be expressed in short form with

$$(R \perp R \parallel R \perp R \perp R \perp R) \tag{7.9}$$

With this, the complete transformation matrix can be derived in the form as described in 7.3:

$$^{0}T_6 =$$

$$\begin{bmatrix} \cos\theta_1 & -\sin\theta_1 * \cos\alpha_1 & \sin\theta_1 * \sin\alpha_1 & a_1 * \cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 * \cos\alpha_1 & -\cos\theta_1 * \sin\alpha_1 & a_1 * \sin\theta_1 \\ 0 & \sin\alpha_1 & \cos\alpha_1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} *$$
$$\begin{bmatrix} \cos\theta_2 & -\sin\theta_2 * \cos\alpha_2 & \sin\theta_2 * \sin\alpha_2 & a_2 * \cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 * \cos\alpha_2 & -\cos\theta_2 * \sin\alpha_2 & a_2 * \sin\theta_2 \\ 0 & \sin\alpha_2 & \cos\alpha_2 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} *$$
$$\begin{bmatrix} \cos\theta_3 & -\sin\theta_3 * \cos\alpha_3 & \sin\theta_3 * \sin\alpha_3 & a_3 * \cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 * \cos\alpha_3 & -\cos\theta_3 * \sin\alpha_3 & a_3 * \sin\theta_3 \\ 0 & \sin\alpha_3 & \cos\alpha_3 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} *$$
$$\begin{bmatrix} \cos\theta_4 & -\sin\theta_4 * \cos\alpha_4 & \sin\theta_4 * \sin\alpha_4 & a_4 * \cos\theta_4 \\ \sin\theta_4 & \cos\theta_4 * \cos\alpha_4 & -\cos\theta_4 * \sin\alpha_4 & a_4 * \sin\theta_4 \\ 0 & \sin\alpha_4 & \cos\alpha_4 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} *$$
$$\begin{bmatrix} \cos\theta_5 & -\sin\theta_5 * \cos\alpha_5 & \sin\theta_5 * \sin\alpha_5 & a_5 * \cos\theta_5 \\ \sin\theta_5 & \cos\theta_5 * \cos\alpha_5 & -\cos\theta_5 * \sin\alpha_5 & a_5 * \sin\theta_5 \\ 0 & \sin\alpha_5 & \cos\alpha_5 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} *$$
$$\begin{bmatrix} \cos\theta_6 & -\sin\theta_6 * \cos\alpha_6 & \sin\theta_6 * \sin\alpha_6 & a_6 * \cos\theta_6 \\ \sin\theta_6 & \cos\theta_6 * \cos\alpha_6 & -\cos\theta_6 * \sin\alpha_6 & a_6 * \sin\theta_6 \\ 0 & \sin\alpha_6 & \cos\alpha_6 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.10}$$

This set of matrices can be filled with the numerical values from table 7.2:

$$
{}^0T_6 =
\begin{bmatrix}
\cos\theta_1 & -\sin\theta_1 * \cos(\pi/2) & \sin\theta_1 * \sin(\pi/2) & 312 * \cos\theta_1 \\
\sin\theta_1 & \cos\theta_1 * \cos(\pi/2) & -\cos\theta_1 * \sin(\pi/2) & 312 * \sin\theta_1 \\
0 & \sin(\pi/2) & \cos(\pi/2) & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} *
\begin{bmatrix}
\cos\theta_2 & -\sin\theta_2 * \cos(0) & \sin\theta_2 * \sin(0) & 1075 * \cos\theta_2 \\
\sin\theta_2 & \cos\theta_2 * \cos(0) & -\cos\theta_2 * \sin(0) & 1075 * \sin\theta_2 \\
0 & \sin(0) & \cos(0) & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} *
\begin{bmatrix}
\cos\theta_3 & -\sin\theta_3 * \cos(\pi/2) & \sin\theta_3 * \sin(\pi/2) & 225 * \cos\theta_3 \\
\sin\theta_3 & \cos\theta_3 * \cos(\pi/2) & -\cos\theta_3 * \sin(\pi/2) & 225 * \sin\theta_3 \\
0 & \sin(\pi/2) & \cos(\pi/2) & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} *
\begin{bmatrix}
\cos\theta_4 & -\sin\theta_4 * \cos(-\pi/2) & \sin\theta_4 * \sin(-\pi/2) & 0 * \cos\theta_4 \\
\sin\theta_4 & \cos\theta_4 * \cos(-\pi/2) & -\cos\theta_4 * \sin(-\pi/2) & 0 * \sin\theta_4 \\
0 & \sin(-\pi/2) & \cos(-\pi/2) & 1280 \\
0 & 0 & 0 & 1
\end{bmatrix} *
\begin{bmatrix}
\cos\theta_5 & -\sin\theta_5 * \cos(\pi/2) & \sin\theta_5 * \sin(\pi/2) & 0 * \cos\theta_5 \\
\sin\theta_5 & \cos\theta_5 * \cos(\pi/2) & -\cos\theta_5 * \sin(\pi/2) & 0 * \sin\theta_5 \\
0 & \sin(\pi/2) & \cos(\pi/2) & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} *
\begin{bmatrix}
\cos\theta_6 & -\sin\theta_6 * \cos(0) & \sin\theta_6 * \sin(0) & 0 * \cos\theta_6 \\
\sin\theta_6 & \cos\theta_6 * \cos(0) & -\cos\theta_6 * \sin(0) & 0 * \sin\theta_6 \\
0 & \sin(0) & \cos(0) & 235 \\
0 & 0 & 0 & 1
\end{bmatrix}
\tag{7.11}
$$

With some simplification this results in the following equation:

$^0T_6 =$

$$\begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & 312 * \cos\theta_1 \\ \sin\theta_1 & 0 & -\cos\theta_1 & 312 * \sin\theta_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} *$$
$$\begin{bmatrix} \cos\theta_2 & -\sin\theta_2 * & 0 & 1075 * \cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & 1075 * \sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} *$$
$$\begin{bmatrix} \cos\theta_3 & 0 & \sin\theta_3 & 225 * \cos\theta_3 \\ \sin\theta_3 & 0 & -\cos\theta_3 & 225 * \sin\theta_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} *$$
$$\begin{bmatrix} \cos\theta_4 & 0 & -\sin\theta_4 & 0 \\ \sin\theta_4 & 0 & \cos\theta_4 & 0 \\ 0 & -1 & 0 & 1280 \\ 0 & 0 & 0 & 1 \end{bmatrix} *$$
$$\begin{bmatrix} \cos\theta_5 & 0 & \sin\theta_5 & 0 \\ \sin\theta_5 & 0 & -\cos\theta_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} *$$
$$\begin{bmatrix} \cos\theta_6 & -\sin\theta_6 & 0 & 0 \\ \sin\theta_6 & \cos\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 235 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.12}$$

With matrix 7.5, the forward kinematic equations can be formed:

$$
\begin{aligned}
n_x = {} & \\
& \cos\theta_1(\cos(\theta_2+\theta_3)(\cos\theta_4\cos\theta_5\cos\theta_6-\sin\theta_4\sin\theta_6)-\sin(\theta_2+\theta_3)\sin\theta_5\cos\theta_6) \\
& \quad +\sin\theta_1(\sin\theta_4\cos\theta_5\cos\theta_6-\cos\theta_4\sin\theta_6) \\
& n_y \\
& = \\
& \sin\theta_1(\cos(\theta_2+\theta_3)(\cos\theta_4\cos\theta_5\cos\theta_6-\sin\theta_4\sin\theta_6)-\sin(\theta_2+\theta_3)\sin\theta_5\cos\theta_6) \\
& \quad -\cos\theta_1(\sin\theta_4\cos\theta_5\cos\theta_6-\cos\theta_4\sin\theta_6) \\
& n_z \\
& = \\
& \sin(\theta_2+\theta_3)(\cos\theta_4\cos\theta_5\cos\theta_6-\sin\theta_4\sin\theta_6)-\cos(\theta_2+\theta_3)\sin\theta_5\cos\theta_6 \\
& o_x \\
& = \\
& \cos\theta_1(-\cos(\theta_2+\theta_3)(\cos\theta_4\cos\theta_5\sin\theta_6+\sin\theta_4\cos\theta_6)-\sin(\theta_2+\theta_3)\sin\theta_5\sin\theta_6) \\
& \quad -\sin\theta_1(\sin\theta_4\cos\theta_5\sin\theta_6-\cos\theta_4\sin\theta_6) \\
& o_y \\
& = \\
& \sin\theta_1(-\cos(\theta_2+\theta_3)(\cos\theta_4\cos\theta_5\sin\theta_6+\sin\theta_4\cos\theta_6)-\sin(\theta_2+\theta_3)\sin\theta_5\sin\theta_6) \\
& \quad +\cos\theta_1(\sin\theta_4\cos\theta_5\sin\theta_6-\cos\theta_4\sin\theta_6) \\
& o_z \\
& = \\
& \quad -\sin(\theta_2+\theta_3)(\cos\theta_4\cos\theta_5\sin\theta_6+\sin\theta_4\sin\theta_6)-\cos(\theta_2+\theta_3)\sin\theta_5\sin\theta_6 \\
& a_x \\
& = \\
& \cos\theta_1(\cos(\theta_2+\theta_3)\cos\theta_4\sin\theta_5+\sin(\theta 2+\theta_3)\cos\theta_5)+\sin\theta_1\sin\theta_4\sin\theta_5 \\
& a_y \\
& = \\
& \sin\theta_1(\cos(\theta_2+\theta_3)\cos\theta_4\sin\theta_5+\sin(\theta 2+\theta_3)\cos\theta_5)+\cos\theta_1\sin\theta_4\sin\theta_5 \\
& a_z \\
& = \\
& \sin(\theta_2+\theta_3)\cos\theta_4\sin\theta_5-\cos(\theta_+\theta_3)\cos\theta_5 \\
& p_x \\
& = \\
& \cos\theta_1(a_1+a_2\cos\theta_2+a_3\cos(\theta_2+\theta_3)+d_4\sin(\theta_2+\theta_3)+d_6(\cos(\theta_2+\theta_3)\cos\theta_4\sin\theta_5+\sin(\theta_2+\theta_3)\cos\theta_5)) \\
& \quad +d_6\sin\theta_1\sin\theta_4\sin\theta_5 \\
& p_y \\
& = \\
& \sin\theta_1(a_1+a_2\cos\theta_2+a_3\cos(\theta_2+\theta_3)+d_4\sin(\theta_2+\theta_3)+d_6(\cos(\theta_2+\theta_3)\cos\theta_4\sin\theta_5+\sin(\theta_2+\theta_3)\cos\theta_5)) \\
& \quad +d_6\sin\theta_1\sin\theta_4\sin\theta_5 \\
& p_z \\
& = \\
& a_2\sin\theta_2+a_3\sin(\theta_2+\theta_3)-d_4\cos(\theta 2+\theta_3)+d_6(\sin(\theta_2+\theta_3)\cos\theta_4\sin\theta_5-\cos(\theta_2+\theta_3)\cos\theta_5)
\end{aligned}
\tag{7.13}
$$

In these equations, the numerical values were replaced with their parameter name. These numeric values can be found in section 7.2.
These kinematic equations can be used to simulate the movements of a robot with given angle-values for the joints.

7.5 Inverse Kinematics

With the forward kinematic equations, the position of the end effector can be determined relative to the base frame for given values of the joint variables. In the context of robotics, it is also necessary to determine these joint variables for a given end effector position. This is referred to as the inverse kinematic problem. Solving the inverse kinematic problem for a given serial link actuator allows to transform a motion plan of the EOAT into joint actuator trajectories for the robot.

7.5.1 Existence of solutions

To determine the joint values for a EOAT-position it needs to be determined, if there exists a solution. Not all positions are equally reachable. There are two types of workspace:

Dextrous workspace volume of space that the robot end-effector can reach with all orientations

reachable workspace volume of space, that the robot can reach in at least one orientation

([10], ch4, p.102)
A manipulator with less than 6DOF cannot reach all positions and orientations in 3D-space. Further limitations can be imposed by the axis alignment. A planar manipulator for example cannot reach out of the plane. [10]

7.5.2 multiple solutions

A 6DOF robot has multiple sets of inverse solutions. To understand why there are multiple solutions, it helps to have a look at fig. **??**. The same end effector position and orientation can be reached in two ways.

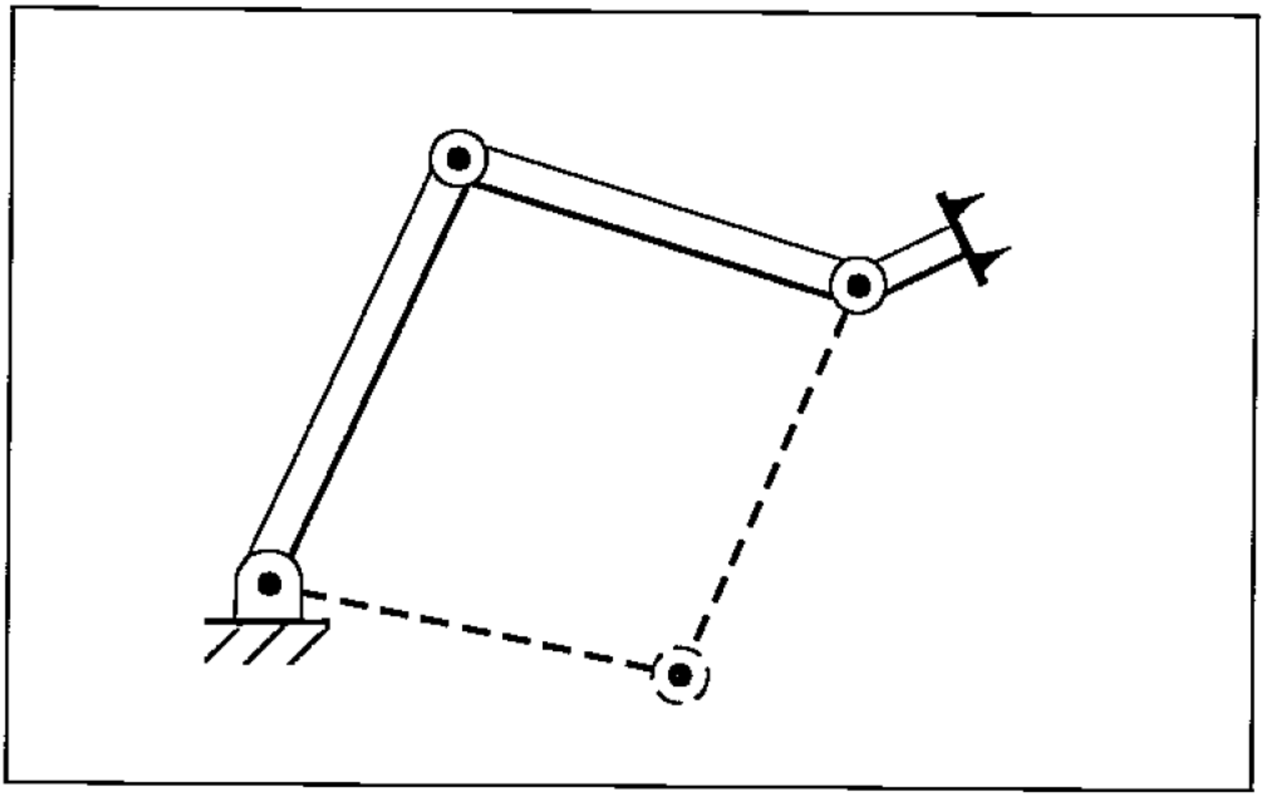

Figure 7.7: Three link manipulator with 2 solutions

This can be extended to the 6DOF serial link manipulators. As stated by YanWu et al. there are 8 groups of inverse solution for most EOAT-positions within the manipulator's workspace [54].

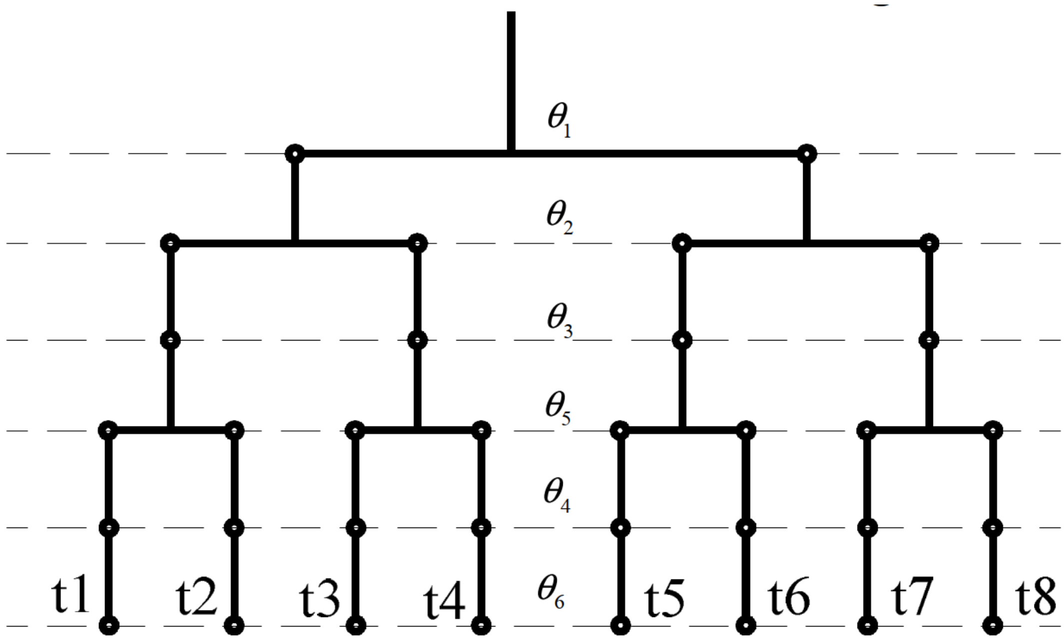

Figure 7.8: Tree of inverse solutions ([54], fig.2)

7.5.3 Inverse solution of 6 DOF Robot

As shown in chapter 7.3 the end effector position is defined by four 3×1 vectors (n, o, a, p). With these known, the joint variables $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$ can be calculated. This problem is solved by the method of using the inverse transformation of unknown link pre-multiplication as presented by Yan Wu et al. [54].

Forward kinematic equation For a 6DOF Robot, the forward kinematic matrix equation from equation eqs. (7.4) and (7.5) can be reformulated into equation 7.14

$$ {}^0_6T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^0_1T(\theta_1){}^1_2T(\theta_2){}^2_3T(\theta_3){}^3_4T(\theta_4){}^4_5T(\theta_5){}^5_6T(\theta_6) \tag{7.14} $$

7.5.4 The inverse Solution

With (n, o, a, p) known, the angle values of the joint variables can be determined by separation of the joint variables. In a first step eq. 7.14 is reformed into eq. 7.15.

$$ {}^2T_3{}^3T_4{}^4T_5 = {}^0T_6({}^0T_1)^{-1}({}^1T_2)^{-1}({}^5T_6)^{-1} \tag{7.15} $$

This gives 2 matrices, formed by the terms found in eq. 7.10. These matrices are the terms found in eq. 7.15. With the help of MATLAB and the symbolic toolbox, the inverses and the overall product of each side can be calculated.

$$ {}^2T_3{}^3T_4{}^4T_5 = $$

$$ \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} c_3c_4c_5 - s_3s_5 & -c_3c_4s_5 - s_3c_5 & c_3s_4 & c_3a_3 - s_3d_4 + a_2 \\ s_3c_4c_5 + c_3s_5 & -s_3c_4s_5 + c_3c_5 & s_3s_4 & s_3a_3 + c_3d_4 \\ -s_4c_5 & s_4s_5 & c_4 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array} \tag{7.16} $$

$$
{}^0T_6({}^0T_1)^{-1}({}^1T_2)^{-1}({}^5T_6)^{-1} =
$$

$$
\begin{pmatrix}
(c_2c_1n_x + c_2s_1n_y - s_2n_z)c_6 - (c_2c_1o_x + c_2s_1o_y - s_2o_z)s_6 & c_2c_1a_x + c_2s_1a_y - s_2a_z & -(c_2c_1n_x + c_2s_1n_y - s_2n_z)s_6 - (c_2c_1o_x + c_2s_1o_y - s_2o_z)c_6 & -(c_2c_1a_x + c_2s_1a_y - s_2a_z)d_6 + c_2c_1p_x + c_2s_1p_y - s_2p_z - c_2a_1 \\
-(s_2c_1n_x - s_2s_1n_y - c_2n_z)c_6 - (-s_2c_1o_x - s_2s_1o_y - c_2o_z)s_6 & -s_2c_1a_x - s_2s_1a_y - c_2a_z & -(-s_2c_1n_x - s_2s_1n_y - c_2n_z)s_6 - (-s_2c_1o_x - s_2s_1o_y - c_2o_z)c_6 & -(-s_2c_1a_x - s_2s_1a_y - c_2a_z)d_6 - s_2c_1p_x - s_2s_1p_y - c_2p_z + s_2a_1 \\
(-s_1n_x + c_1n_y)c_6 - (-s_1o_x + c_1o_y)s_6 & -s_1a_1 + c_1a_2 & -(-s_1n_x + c_1n_y)s_6 - (-s_1o_x + c_1o_y)c_6 & -(-s_1a_x + c_1a_y)d_6 - s_1p_x + c_1p_y \\
0 & 0 & 0 & 1
\end{pmatrix}
\tag{7.17}
$$

In equations eqs. (7.16) and (7.17) $sin(\theta_i)$ and $\cos(\theta_i)$ have been replaced by s_i and c_i to allow for appropriate presentation of the matrix. This abbreviation will be used for following steps as well.

As shown by eq. 7.15, the elements of matrices eqs. (7.16) and (7.17) are corresponding equal. This allows to find analytic expressions for all θ_i. As stated in section 7.5.2, there are two solutions for each θ_i.

θ_1: By taking the elements (3,4) of equations eqs. (7.16) and (7.17) as corresponding equal, equation 7.18 is formed.

$$
-(-s_1a_x + c_1a_y)d_6 - s_1p_x + c_1p_y = 0 \tag{7.18}
$$

Reforming equation 7.18 gives a solution for θ_1 as seen in equation 7.19.

$$
\theta_{1_1} = \operatorname{atan2}(p_y - a_yd_6, p_x - a_xd_6) \quad \theta_{1_2} = \operatorname{atan2}(-p_y + a_yd_6, -p_x + a_xd_6) \tag{7.19}
$$

θ_2: Taking elements (1,4) and (2,4) of equations eqs. (7.16) and (7.17) as corresponding equal, equations 7.20 and 7.21 are formed.

$$
a_3c_3 - d_4s_3 + a_2 = -(c_2c_1a_x + c_2s_1a_y - s_2a_z)d_6 + c_2c_1p_x + c_2s_1p_y - s_2p_z - c_2a_1 \tag{7.20}
$$

$$
a_3s_3 + d_4c_3 = -(-s_2c_1a_x - s_2s_1a_y - c_2a_z)d_6s_2c_1p_xs_2s_1p_y - c_2p_z + s_2a_1 \tag{7.21}
$$

For simplification of the terms in equations eqs. (7.20) and (7.21), middle variables are chosen:

$$
u = c_1(p_xd_6a_x) + s_1(p_y - a_yd_6) - a_1 \ ...]v = p_z - a_zd_6 \tag{7.22}
$$

With these middle variables equations eqs. (7.20) and (7.21) can be rewritten as seen in eqs. (7.23) and (7.24).

$$
a_3c_3 - d_4s_3 = c_2u - s_2v - a_2 \tag{7.23}
$$

$$
a_3s_3 + d_4c_3 = -s_u - c_2v \tag{7.24}
$$

To eliminate s_3 and c_3 equations eqs. (7.23) and (7.24) are squared and then added to form 7.25.

$$
a_3^2 + d_4^2 = u^2 + v^2 - 2a_2c_2u + 2a_2s_2v + a_2^2 \tag{7.25}
$$

$$
\frac{a_3^2 + d_4^2 - u^2 - v^2 - a_2^2}{-2a_2} = c_2u - s_2v \tag{7.26}
$$

This equation can be further simplified by taking another middle variable:

$$
m = \frac{a_3^2 + d_4^2 - u^1 - v^2 - a_2^2}{-2a_2}
$$

Reforming equation 7.26 and inserting the middle variable reveals a solution for θ_2 as seen in eq7.27.

$$
\theta_{2_{1/2}} = \operatorname{atan2}(-v, u) \pm \operatorname{atan2}(\sqrt{v^2 + u^2 - m^2}, m) \tag{7.27}
$$

θ_3**:** To recive θ_3 it helps to take equations eqs. (7.20) and (7.21). Equation 7.20 needs to be reformed into 7.28

$$a_c c_3 d_4 s_3 + a_2 = c_2 u - s_2 v \tag{7.28}$$

Squaring of equations eqs. (7.21) and (7.28) and adding them eliminates s_2 and c_2 as seen in eq 7.29

$$a_2^2 + a_3^2 + d_4^2 + 2a_2(a_3 c_3 - d_4 s_3) = u^2 + v^2 \tag{7.29}$$

$$a_3 c_3 - d_4 s_3 = \frac{a_3^2 + d_4^2 - u^2 - v^2 + a_2^2}{2a_2} \tag{7.30}$$

Again, this equation can be further simplified by taking another middle variable:

$$-\frac{a_3^2 + d_4^2 - u^2 - v^2 + a_2^2}{2a_2}$$

Reforming equation 7.30 and inserting the middle variable reveals a solution for θ_3 as seen in eq7.31.

$$\theta_{3_{1/2}} = \operatorname{atan2}(-d_4, a_3) \pm A\tan(2(\sqrt{d_4^2 + a_3^2 - h^2}, h)) \tag{7.31}$$

θ_5**:** Taking elements (2,2) and (1,2) of equations eqs. (7.16) and (7.17) as corresponding equal, equations 7.32 and 7.33 are formed.

$$-s_3 c_4 s_5 + c_3 c_5 = -c_1 s_2 a_x - s_1 s_2 a_y - c_2 a_z \tag{7.32}$$

$$-c_3 c_4 s_5 - s_3 c_4 = c_1 c_2 a_x + s_1 c_2 a_y - s_2 a_z \tag{7.33}$$

To combine equations eqs. (7.32) and (7.33), eq 7.32 is multiplied with c_3 and eq 7.33 is multiplied by s_2. The resulting equations are then subtracted from each other giving eq 7.34.

$$c_5 = (-c_1 s_2 a_x - s_1 s_2 a_y - c_2 a_z)c_3 - (c_1 c_2 a_x + s_1 c_2 a_y - s_2 a_z)s_3 \tag{7.34}$$

Reforming equation 7.34 gives a solution for θ_5 as seen in equation 7.35.

$$\theta_{5_{1/2}} = \pm\operatorname{acos}[(-c_1 s_2 a_x - s_1 s_2 a_y - c_2 a_z)c_3 - (c_1 c_2 a_x + s_1 c_2 a_y - s_2 a_z)s_3] \tag{7.35}$$

θ_4**:** With θ_5 and θ_{1-3} known, equation 7.33 can be reformed into a solution for θ_4 as seen in equation 7.36

$$\theta_{4_{1/}} = \pm\arccos[\frac{c_1 c_2 a_x + s_1 c_2 a_y - s_2 a_z + s_3 c_5}{-c_3 s_5}] \tag{7.36}$$

θ_6**:** By taking the elements (3,3) of equations eqs. (7.16) and (7.17) as corresponding equal, equation 7.37 is formed.

$$s_6(s_1 n_x - c_1 n_y) + c_6(s_1 o_x - c_1 o_y) = c_4 \tag{7.37}$$

With θ_4 and θ_1 known, equation 7.37 can be reformed into a solution for θ_6 as seen in equation 7.38.

$$\theta_{6_{1/2}} = \operatorname{atan2}(s_1 n_x - c_1 n_y, s_1 o_x - c_1 o_y) \pm \operatorname{atan2}(\sqrt{(s_1 n_x - c_1 n_y)^2 + (s_1 o_x - c_1 o_y)^2 - c_4^2}, c_4) \tag{7.38}$$

With equations eqs. (7.19), (7.27), (7.31), (7.35), (7.36) and (7.38) the values for $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$ can be calculated for all reachable end effector positions.
The atan2(x, y)-function used in this solution which returns an angle between the positive x axis and a line pointing to $(x, y) \neq (0, 0)$. The function takes two arguments and returns a single value θ between $-\pi$ and π equivalent to the phase angle of complex numbers. While arctan can only solve in quadrant 1 and 4, arctan2 allows for calculation in all four quadrants because it differentiates between pos/neg x and y direction independently. It is mostly a question of implementation and solving method, so this new notation arose with programming languages.

7.5.5 Optimization of inverse kinematics solution

As stated in section 7.5.2, there exist eight sets of inverse solutions in the dextrous workspace (see 7.5.1). The optimal set of solutions can be found through an optimization function. This optimization function needs to find the set of solution that can reach the desired position fastest from the current position.

cost function Equation 7.39 is the cost function to be minimized.

$\theta_i(k+1)$: target angle

$\theta_i(k)$: departing angle

k_i power value

$$err = \min \sum_{i=1}^{6} k_i[\theta_i(k+1) - \theta_i(k)] \tag{7.39}$$

As can be seen, this cost function minimizes the sum of angles over all axes for the movement of the end effector from position $[n_i, o_i, a_i, p_i]$ to $[n_{i+1}, o_{i+1}, a_{i+1}, p_{i+1}]$ by using the inverse solution presented in chapter 7.5.4.

limitations of cost function This cost function not necessarily gives the fastest path, as it does not account for the actual movement speed of the different axes. A possible solution for this would require multiplying with a cost factor between 0 and 1 for each joint (1 for the slowest joint) with the respective angle error to account for the difference in angular velocity.

description of implementation An implementation of this function receives the current position either in joint angles or Cartesian coordinates of the end-effector which could be given in form of a rotation matrix as seen in eq7.5 or in form of euclidean space and euler angle coordinates. $[x, y, z, \alpha, \beta, \gamma]$ which could be transformed by the inverse of equation 7.6 into a rotation matrix.
This algorithm would then start the calculation to get all possible paths in the tree of inverse solutions (see figure 7.8) for position $i+1$.

algorithm for minimal computational effort To minimize the computational effort, following algorithm is proposed by Yan Wu et al. [54]:

1. The calculation starts with θ_1 by finding the two solutions for equation 7.19.
2. When calculating θ_2:
   (a) The term $v^2 + u^2 - m^2$ should be > 0, otherwise the program can terminate the calculation of the subsequent steps of this group, return $err = inf$ or NaN , and provide a varible like $P = 1$ to determine the cause for the abortion of subsequent steps.
   (b) If there is a solution for θ_2 it needs to be tested for extraneous roots. An extraneous root is introduced into an equation in the process of solving another equation, but is not a solution of the equation to be solved [45]. This test is done simply by substituting the solution for θ_2 into the original equation 7.25. If an extraneous root is found, the program can terminate and return $err = inf$ or NaN and $P = 2$ for cause of abortion.
3. If the calculation has been continued, θ_3 can be determined. Again a test for extraneous roots is necessary.
4. When calculating θ_5, test for extraneous roots.
5. When calculating θ_4, test for extraneous roots.
6. calculate θ_6
7. determine error as seen in eq 7.39 for all possible paths
8. find $\min(err(t_1^8))$

The resulting set of angle values is the result of the algorithm. It is the unique inverse solution, determined easiest to reach with the given cost function in eq 7.39.

7.6 Denavit-Hartenberg-Convention

TheDH-Convention is a commonly used and simplifies the forward and backward transformation. It was named after Jaques Denavit and Richard Hartenberg who developed a general theory to describe a serial link mechanism. [24]
It consists of following parts:

- DH-Convention for establishing the coordinate systems
- DH-Transformation for generation of the coordinate systems
- DH-Parameters as a result form the transformations

Determining the coordinate systems is done according to set rules. Nevertheless, the choices of coordinate frames are also not uniqe, so different peole will derive different, but correct frame assignments. This freedom of choice should be used to bring as many DH-Parameters as possible to zero. This simplifies subsequent equations and calculations. [55]
Each joint of the robot is described by four parameters.

7.7 Robot configurations

As mentioned before, a 6 DOF manipulator can reach the same position in multiple configurations. The different configurations in the solutions can be seen in the example of the FANUC robot arms (see figure 7.9)

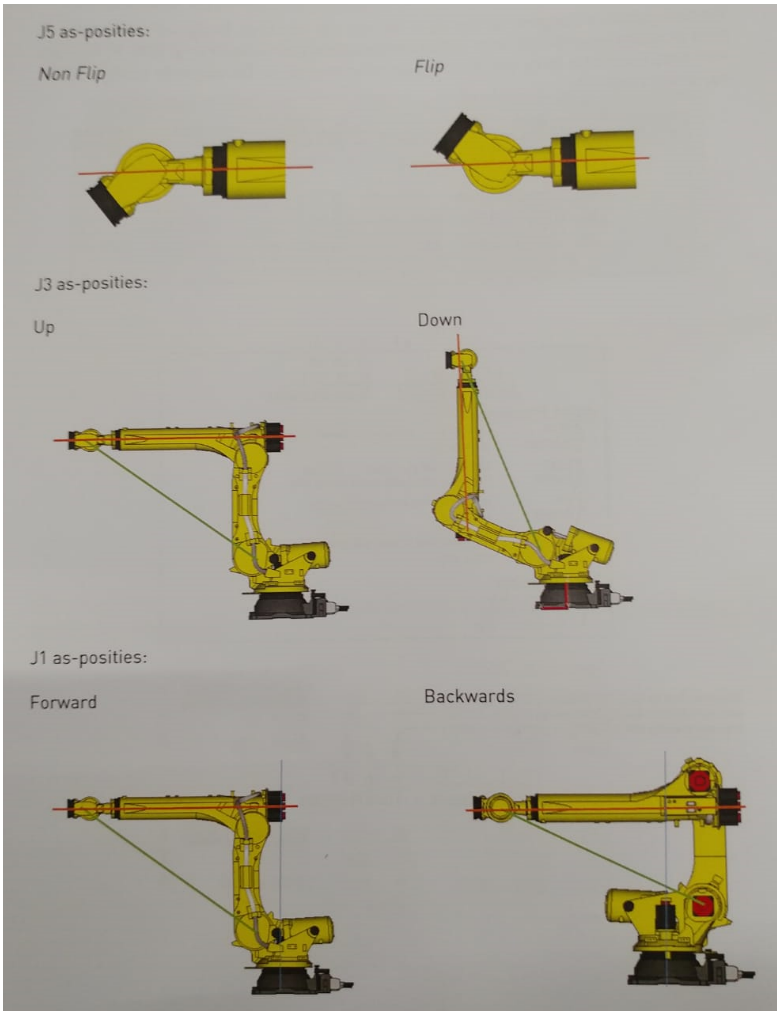

Figure 7.9: 6 axis robot configurations on the example of a FANUC Robot [15]

8. Developement of a control strategy

9. Conclusions and further work

References

[1] Riku Ala-Laurinaho. “Sensor data transmission from a physical twin to a digital twin; Anturidatan lähettäminen fyysiseltä kaksoselta digitaaliselle kaksoselle”. en. G2 Pro gradu, diplomityö. May 6, 2019, pp. 90 + 15. URL: http://urn.fi/URN:NBN:fi:aalto-201905123028.

[2] Ahmed R. J. Almusawi, Lale Dülger, and Sadettin Kapucu. “A New Artificial Neural Network Approach in Solving Inverse Kinematics of Robotic Arm (Denso VP6242)”. In: *Computational Intelligence and Neuroscience* 2016 (Aug. 2016), p. 10. DOI: 10.1155/2016/5720163.

[3] author. *title*.

[4] *automaton*. URL: https://www.merriam-webster.com/dictionary/automaton.

[5] *Autonomous Robot Vehicles*. Aug. 6, 1991.

[6] Lukas Barinka. “Inverse Kinematics - Basic Methods”. In: *booktitle*. Ed. by Ing. Roman Berka. Mar. 21, 2002.

[7] Kamel Bouzgou and Zoubir Ahmed-Foitih. “Workspace Analysis and Geometric Modeling of 6 DOF Fanuc 200IC Robot”. In: vol. 182. 4th WORLD CONFERENCE on EDUCATIONAL TECHNOLOGY RESEARCHES (WCETR-2014). 2015, pp. 703–709. DOI: https://doi.org/10.1016/j.sbspro.2015.04.817. URL: http://www.sciencedirect.com/science/article/pii/S187704281503092X.

[8] Daniel Constantin et al. “Forward Kinematic Analysis of an Industrial Robot”. In:

[9] Peter Corke. *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. 1st. Springer Publishing Company, Incorporated, 2013. ISBN: 3642201431.

[10] J.J. Craig. *Introduction to Robotics: Mechanics & Control*. Introduction to robotics : Mechanics & control / John J ... Craig. found at https://robotics.stackexchange.com/questions/10322/is-it-possible-to-get-all-possible-solutions-of-inverse-kinematics-of-a-6-dof-ar. Addison-Wesley Publishing Company, 1986. ISBN: 9780201103267. URL: https://books.google.nl/books?id=uDNSAAAAMAAJ.

[11] Mahidzal Dahari and Jian-Ding Tan. “Forward and inverse kinematics model for robotic welding process using KR16KS KUKA robot”. In: *2011 4th International Conference on Modeling, Simulation and Applied Optimization, ICMSAO 2011* (Apr. 2011). DOI: 10.1109/ICMSAO.2011.5775598.

[12] *Digital twin*. URL: https://en.wikipedia.org/wiki/Digital_twin#Definitions.

[13] A. El Saddik. “Digital Twins: The Convergence of Multimedia Technologies”. In: *IEEE MultiMedia* 25.2 (Apr. 2018), pp. 87–92. ISSN: 1941-0166. DOI: 10.1109/MMUL.2018.023121167.

[14] *European Train Control System (ETCS)*. URL: https://www.thalesgroup.com/en/european-train-control-system-etcs.

[15] *Fanuc Academy Manual*. Dutch. Robot hands on manual. FANUC.

[16] Roy Featherstone. *Robot dynamics*. Version revision Nr. 91723. Brain Corporation. Oct. 21, 2011. DOI: doi:10.4249/scholarpedia.3829. URL: http://www.scholarpedia.org/article/Robot_dynamics (visited on 01/25/2020).

[17] *FHEM Homepage*. URL: https://fhem.de/ (visited on 01/20/2020).

[18] Michael Grieves. “Digital Twin: Manufacturing Excellence through Virtual Factory Replication”. In: (Mar. 2015). URL: https://www.researchgate.net/publication/275211047_Digital_Twin_Manufacturing_Excellence_through_Virtual_Factory_Replication.

[19] *Inverse Kinematik*. DE. note. Wikipedia. URL: https://de.wikipedia.org/wiki/Inverse_Kinematik (visited on 01/16/2020).

[20] Jamshed Iqbal, Muhammad Ul Islam, and Hamza Khan. “Modeling and analysis of a 6 DOF robotic arm manipulator”. In: *Canadian Journal on Electrical and Electronics Engineering* 3.6 (Jan. 2012), pp. 300–306. URL: https://www.researchgate.net/publication/280643085.

[21] Reza N. Jazar. "Inverse Kinematics". In: *Theory of Applied Robotics: Kinematics, Dynamics, and Control*. Boston, MA: Springer US, 2007, pp. 263–296. ISBN: 978-0-387-68964-7. DOI: 10.1007/978-0-387-68964-7_6. URL: https://doi.org/10.1007/978-0-387-68964-7_6.

[22] Pradeep K. Khosla and Takeo Kanade. "Experimental Evaluation of Nonlinear Feedback and Feedforward Control Schemes for Manipulators". In: *The international Journal of Robotics Research* (Feb. 1988). DOI: 10.1177/027836498800700102.

[23] Werner Kritzinger et al. "Digital Twin in manufacturing: A categorical literature review and classification". In: *IFAC-PapersOnLine* 51.11 (2018). 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018, pp. 1016–1022. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2018.08.474. URL: http://www.sciencedirect.com/science/article/pii/S2405896318316021.

[24] *LESSON Denavit-Hartenberg notation*. EN. Format: Video lecture. Queesland University of Technology. Apr. 9, 2017. URL: https://robotacademy.net.au/lesson/denavit-hartenberg-notation/ (visited on 01/14/2020).

[25] Bin LI et al. "Dynamic Multibody Simulation of a 6-DOF Robotic Arm". In: *Advanced Materials Research* 139-141 (2010), pp. 1001–1004. DOI: 10.4028/www.scientific.net/AMR.139-141.1001.

[26] *Motion planning*. Wikipedia. URL: https://en.wikipedia.org/wiki/Motion_planning (visited on 01/22/2020).

[27] Stephen Mraz. *The Difference between Medial and Lateral, Proximal and Distal, and Superior and Inferior (Biomechanics)*. June 22, 2016. URL: https://www.machinedesign.com/markets/medical/article/21834827/the-difference-between-medial-and-lateral-proximal-and-distal-and-superior-and-inferior-biomechanics (visited on 02/03/2020).

[28] Dr.-Ing. John Nassou. *Inverse Kinematics Serial link manipulators*. Humanoide Roboter (ehem. Robotik). presentation slides. Technische Universität Chemnitz - Fakultät für Informatik. Nov. 22, 2016. URL: https://www.tu-chemnitz.de/informatik/KI/edu/robotik/ws2016/lecture-ik%201.pdf (visited on 01/18/2020).

[29] Trung Nguyen. Company Supervisor for project.

[30] Richard P. Paul. "Robot manipulators : mathematics, programming, and control : the computer control of robot manipulators". In: 1981.

[31] *Program inverse kinematics algorithms with MATLAB*. Mathworks online tutorial.

[32] Qinglin Qi et al. "Digital Twin Service towards Smart Manufacturing". In: *booktitle*. 2018.

[33] Li Ze-xiang Richard M. Murray S. Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[34] *Robot*. URL: https://www.merriam-webster.com/dictionary/robot.

[35] *Roboterkinematik*. DE. Version 1. Wikipedia. May 28, 2005. URL: https://upload.wikimedia.org/wikipedia/de/a/ae/Roboterkinematik.png (visited on 01/15/2020).

[36] Open Robotics. Website of ROS project.

[37] *Robotics 1 U1 (Kinematics) S2 (Kinematic Diagrams) P1 (Denavit-Hartenberg Frames)*. URL: https://www.youtube.com/watch?v=ut4uZ6Yzv6o&list=PLT_0lwItn0sDBE98BsbaZezflB96ws12b&index=7.

[38] *Robotics Toolbox. Introduction*. EN. Homepage of Peter Corke. URL: https://petercorke.com/wordpress/toolboxes/robotics-toolbox (visited on 01/18/2020).

[39] Benjamin Schleich et al. "Shaping the digital twin for design and production engineering". In: *CIRP Annals - Manufacturing Technology* 66 (Apr. 2017), pp. 141–144. DOI: 10.1016/j.cirp.2017.04.040.

[40] Wolfgang Smidt. "Verallgemeinerte inverse Kinematik für Anwendungen in der Robotersimulation und der virtuellen Realität". MA thesis. Institut für Roboterforschung Universität Dortmund, 1998.

[41] Mark W. Spong. *Robot Dynamics and Control*. 1st. USA: John Wiley and Sons, Inc., 1989. ISBN: 047161243X.

[42] Dr. Swet Chandan Tarun Pratap Singh Dr. P. Suresh. "Forward and Inverse Kinematic Analysis of Robotic Manipulators". In: *International Research Journal of Engineering and Technology (IRJET)* 4.2 (Feb. 2017).

[43] *title*. URL: https://specials.han.nl/sites/automotive-research/smart-cell/index.xml.

[44] *title*. URL: https://answers.ros.org/question/12230/what-is-ros-exactly-middleware-framework-operating-system/.

[45] *title*. URL: https://encyclopedia2.thefreedictionary.com/Extraneous+Root.

[46] Ashwin Kumar Venkatesan Keyvan Noury Tiezheng Zhao Fady Tourdas. *CONTROL OF A FANUC ROBOTIC ARM USING MATLAB MANUAL*. USC Viterbi / Aerospace and Mechnaical Engineering department.

[47] Ashwin Kumar Venkatesan Keyvan Noury Tiezheng Zhao Fady Tourdas. *CONTROL OF A FANUC ROBOTIC ARM USING MATLAB MANUAL*. USCViterbi School of Engineering - Aerospace and Mechanical Engineering department.

[48] R. J. Urbanic. "Methods for Evaluating the Functional Work Space for Machine Tools and 6 Axis Serial Robots." In: *SAE International Journal of Materials & Manufacturing.* 9.2 (May 2016), pp. 465–473.

[49] Karl Wallkum. "Using FANUC R-2000iC/210F (6-axis robot) for improved efficiency in FRC parts formation". Major Project Plan. Nov. 2019.

[50] Kong Wei. "Control and Safety Mechanisms for a 3 DOF Manipulator with Human Interaction". Supervisor: Prof.dr.ir. Aart-Jan de Graaf. MA thesis. HAN, Oct. 2014.

[51] *What Is a Robot*. Aug. 24, 2017. URL: https://www.wired.com/story/what-is-a-robot/.

[52] *What's the Difference Between a Simulation and a Digital Twin?* May 24, 2018. URL: https://www.electronicdesign.com/technologies/embedded-revolution/article/21806550/whats-the-difference-between-a-simulation-and-a-digital-twin.

[53] Mike Wilson. *Implementation of Robot Systems*. An Introduction to Robotics, Automation, and Successful Systems Integration in Manufacturing. Butterworth-Heinemann, 2015. DOI: https://doi.org/10.1016/C2012-0-00795-8.

[54] Yan Wu et al. "Inverse Kinematics Solution and Optimization of 6DOF Handling Robot". In: *Applied Mechanics and Materials* 635-637 (2014). DOI: 10.4028/www.scientific.net/AMM.635-637.1355.

[55] Prof. Dr. Klaus Wuest. *Denavit-Hartenberg-Konventionen*. Technische Hochschule Mittelhessen. Wiesenstrasse 14, 35390 Giessen, Deutschland, June 2018. URL: https://homepages.thm.de/~hg6458/Robotik/Denavit-Hartenberg.pdf.

Appendices

A. Appendix 1

B. Appendix 2

C. Robot Quick start guide

A big part of this was derived from [47].

C.0.1 Parts of the robot

A quick overview over the visible parts.

FANUC R-2000iC/210F 6 axis robotic arm

The robot has 6 movable joints with a possible payload attached to the end. Its features include a wide reach (2655 mm), sturdy but flexible arm design, a spring loaded counterbalance, relatively high payload capacity (210Kg) and fast moving axes. The joints also have hashes to indicate the zero positions which help with the calibration of the robot.

R30iA Robot Controller

The Robot is controlled using an original equipment manufacturer controller called the FANUC R30iA controller. Its features include faster sustained speed and superior position accuracies. It also has the I/O ports that are used to connect grippers and other payloads. (It also houses the camera circuits which are required to access the data from the SONY camera provided with the robot. - Delta only) The controller is also provided with a data card slot in which the special SD cards manufactured can be inserted and used as external memory. On the outside of the controller (side of iPendant at Delta) a USB port can be found. With these, programs, firmware files and other files can easily be transferred. Additionally, there is extended connectivity via its Ethernet port e.g. for FTP available.

iPendant

The teaching pendant is the primary user interface to the robot. It is used to move the joints of the robot manually, to program specific trajectories, to control the gripper, and various other actions. It also is an interface that can be used for input and output of the robot controller parameters. The user can access the system variables and position variables. (It is provided with a USB port that can be used to connect to a USB drive for external storage. - Delta only) It can also be used to setup an FTP server and client in order to communicate with the PC.

Gripper

The robot as handed over does not have an active gripper system. It has been tested though with a pneumatic gripper controlled via the DO ports and pneumatic valves. A 2-way pneumatic valve, some piping and a pressure regulator are still available. (Delta Equipment varies here a lot).

C.0.2 iPendant Navigation Manual:

The teaching pendant is the primary user interface to the robot. This section deals with the important buttons on the TP.

Figure C.1: Most important iPendant buttons numbered: This image from another iPendant than the ones available was chosen because of its labelling of buttons. Some buttons of the available iPendants are not labelled, so this picture can be used as a reference. Source: [47]

Emergency stop:

Makes the robot stop immediately by applying brakes. Use it only when necessary as the brakes wear down. There is also an E-stop on the controller. Press down on the button to activate it. Twist it to the right to release. If a slow and gradual halt is required, press HOLD on the iPendant. TP on/off Below the E-stop button. It should be ON to access any function in the Teach Pendant (Set-ups, calibration, programs etc) and OFF when running in AUTO mode. Deadman switch The 2 yellow bars behind the iPendant. 3 modes are available: Fully released, halfway pressed, fully pressed. Only the middle mode activates control.