Twinning Procedure Within Industry 4.0

Digital twin application on safety issue of human-collaborative robot

Student: CHEN Hongyu

S244378

Supervisor: DEMICHELA MICAELA

Abstract

As the global production patterns change gradually into multiple varieties and small batch mode in industry 4.0, automatic production becomes not flexible enough to handle the requirement, therefore the advantages of collaborative robots are becoming increasingly apparent. The safety issues of the collaborative working environment have then also emerged into our sight. This thesis takes collaborative robot as the object of study to discuss a way of improve industrial safety with digital twin application. The purpose of the digital twin application is to achieve the prediction of the working state of the human collaborative robot. The concept of digital twin and the characteristics of human robot collaborative environment is studied, and is compared with other method to specify the target of this thesis. The detailed methodology of building and applying digital twin is designed. The methodology contains the modeling, the calculation of kinematic and dynamic states, the method of guarantee real time characteristic, and the logic design of the control. The software CATIA and Matlab is used as computer aid for the 3D modeling, numerical modeling and calculation. The real time capability is realized by simplification of the data processing process. The logic of control is designed to let the digital twin application smart enough to deal with different situations. Then a case study of Doosan Cobots is conducted to verify the methodology. Each step of building the digital twin of the human robot collaborative environment is described in detail and the performance of detecting and predicting safety issues is simulated and analyzed to draw the final conclusion.

Key words: Digital twin, real time, collaborative robot, prediction, safety

Content

1. Introduction

1.1 Scope

In this thesis, I have designed a way to approve digital twin application in industrial, mainly relative to safety issues of human collaborative robot. Starting with the introduction of some basic conceptions such as digital twin, collaborative robot, safety issue in industrial production and so on. The next part will be a detailed explanation of the design of how to achieve the digital twin application, following with a case study based on a existing collaborative robot as the third part. Then some parameters of the case study is chosen as output to generate an analysis of the application. The conclusion of how this digital twin works on safety insurance and what can be improved in the future will be written in the last part.

1.2 Basic information

In this chapter, basic information relate to the topic, digital twin in industry 4.0, will be introduced. Then according to these information, the general train of thoughts will be decided. The information contains the concept and practical usage of digital twin and the understanding of its structure, the concept of the target of study in this thesis, which is the collaborative robot, and the comparison between different safety insurance system in industry to better understand the advantage and necessity of digital twin application.

1.2.1 Digital Twin

With the improvement of information technologies, many conceptions about intelligent manufacturing have been put forward all over the world. Such as Industrial 4.0 in Germany, Industrial Internet and manufacturing system based on Cyber-Physical Systems(CPS) in USA, and Made in China 2025 and Internet Plus Manufacturing in China. All of these strategies have a common purpose which is to achieve the interconnection and interoperable feature between physical and information world of manufacturing. And to over come the communication issue between the two world, which is the bottle neck, a conception named digital twin is needed.

At present stage, in industrial field, the physical system and the information system is already connected together in a certain extent. However, the connection is mainly implemented by human operation, which is lack of continuity, instantaneous and foresee-ability. For example, in regard to the production factors of the workspace, nowadays, the data of the equipment, material, supplies, goods in process, and etc are all automatically acquired to realize the online entry of the system data. The data of the information system is then synchronized with the data related to the production factors. But in the Industrial 4.0, this is not enough. There need to be an interaction, which fuse the information data and physical state of each factor together. This is also required when it comes to the control of the production process. The use of digital instruments has greatly improved the detection accuracy of the workshop. The computer is used to analyze and optimize the test data, which makes the result more scientific and accurate. Furthermore, the automated actuators improved the control accuracy on the production site. However, there exists a delay between the physical and the information space, in another word, the actual situation of the production process is different from the controlled one. There are no timely handling and response of the system. Although the real-time capability has already been improved as the development of IOT, further efforts are still needed to merge physical and information space together.

At the outset, digital twin was put forward by United States Department of Defense to ensure the maintenance and support of aerospace vehicles. Digital twin is using a digital copy of the physical system to perform real-time optimization[1]. This copy should contain all the factors relate to the physical one, including processes, people, places, systems and devices that can be used for various purposes[2]. There should be sufficient sensors and detectors to update the physical data to the virtual model, while the physical component can be also controlled by operating the virtual data of the digital twin. The interaction between these two is also very important to carry out the real-time control, optimization and even prediction of failure of product and production process.

Digital twin has following characteristics[4].

1. It is the integration of various types of data of the physical object, and is a faithful mapping of it.
2. It exists in the entire life cycle of the physical object, co-evolving with it by continually accumulating relative knowledge and information.
3. It has not only the capability to represent the physical object, but also the ability to optimize the physical part based on the digital model.

The shift to industrial with digital twin can be very difficult for traditional industrial enterprises because it is a conception far from the traditional experience-based design and manufacturing that have been used for decades. Designers may no longer need to rely on the development of the actual physical prototype to verify the design idea, neither the complicated physical experiment to test and verify the reliability of products. Without doing a small batch trial production, the production bottlenecks can be predicted. And don't even need to go to the scene , can designers discern how the customer's products operate.

This way will not only accelerate the product development process, but also improve the effectiveness and efficiency of development and production. The understanding of the usage of products will become more effective and help customers to avoid losses. Furthermore, it can transfer feedback of the customer's actual use more accurately to the designer, implementing effective improvement of the product. Besides, as manufacturer point of view, the cost, time usage will be substantially reduced since there will be less test, experiment and trial production. The mass production line based on cyber-physical system will be more intelligent so that it can quickly adapt to new production requirements. The safety of operator can be guaranteed in a more flexible way. Last but not least, the routine maintenance can be replaced by failure prediction of the digital image and the digital twin which have the ability to control the physical object in a certain way to avoid the predicted accident and dangerous.

1.2.2 Current application of digital twin

Digital twin has already been used in some regions.

Such as, typically, the Grieves combines the physical system to the equivalent virtual system to get a methods of prediction and elimination of complex system failures, which was used in the application experiment in NASA related system[5]. The U.S. Air Force Structural Science Center combines the ultra-high fidelity virtual model of aircraft with the structural deflection and temperature calculation models together to develop the prediction of aircraft structural life based on digital twinning. And the technical advantages of digital twinning is highly admired[6]. Siemens has put forward the concept of "digital twin", which is committed to help manufacturers to build the production system model of integrated manufacturing process in the information space, and enable the digitization of the whole process in the physical space from product design to manufacturing execution[7]. In response to the complex needs of interaction of product user, Dassault Systemes has established a 3D experience platform based on digital twins by using user feedback to constantly improve the product design model of the information world, which optimizes the product entity of the physical world. The aircraft radar was taken as an example for verification[8].

To create digital twin of a system, it is required the data and the information of each asset, which can be read from the CPPS(Cyber-Physical Production System). The CPPS is a set of embedded production systems which communicate and interact with each other in a communication network. The software of the CPPS contains a lot of information about resources, product and process, as well as cycle time, load level of joining points, etc. These information is fundamental to build digital twin and implement a real-time system into the CPPS. The comparison of the information between the current CPPS and the digital image will lead to changes in the production system[9]. Beyond that, the data can be also obtained by multiple sensor. The virtual representation of the physical part is then realized by combining the sensor data and CAD models together. For instance, a digital shop-floor can be rendered in the 3D environment exploiting the capabilities provided by Robot Operating System(ROS) Framework. The workload state and geometrical should also be represented. Furthermore, the dynamic update of the digital twin based on real time sensor and resource data coming from the actual shop-floor is required[10].

One of the most important and fundamental issue of making a digital twin is to guarantee the resource of the data and information. The data can be read from the existing robot’s software or the PLC device and can be obtained from sensor. There is also data about newly integrated factors due to additional mission and requirement, as well as the CAD model of the physical object. All of these data should be then be continuously updated to create a synthesis of them. The data model should be generic enough so that is can deal with multiple cases and can be used by multiple components to carry out their mission. The final job is for the planners of the robot or the mass production line to generate a safe and efficient outline through the digital twin to complete the production missions.

1.2.3 Human-Robot collaboration

As the global production patterns change gradually into multiple varieties and small batch mode, automatic production becomes not flexible enough to handle the requirement, therefore the advantages of collaborative robots are becoming increasingly apparent. This is a producing layout that combines the repetitive performance of robots with the individual skills and ability of people together. People have an excellent capability for solving imprecise exercises, while robots exhibit precision, power and endurance[12]. Compared to the other production patterns, the production using collaborative robot is not as high-yield as automated production, and not as various as manual production, but still has considerable performance. Human-machine collaboration can also bring benefit to efficiency of the production as well as high flexibility. So even there is no particularly outstanding aspect, but the overall utility is the highest.

In order to guarantee the safety during Human-Robot collaboration, relevant safety standards have been established internationally. Like in ISO 10128-1, there are several standards and steps to reduce the risk of accident. One is the safety-rated stop monitoring. In specific spot such as manually assemble station, the robot will stop when human operates get into the stop area. Another one is the monitoring of speed and distance. At the same time the minimum distance for robot stopping should be determined considering also the braking length of the robot. Furthermore, the maximum power and force should be limited during the collaboration. The relevant safety features to minimize risk of a Human-Robot Collaboration application contains safe velocity monitoring, safe workspace and safeguarded zones, safe collision detection, safe force monitoring, safe tool detection, and safe switching of states. The velocity monitoring can combine with the safe workspace so that make a stepwise reduction in velocity of robot dependent on the distance between robot and operator. For example, when the operator is 5 meters away, the robot can move at full speed. While the distance is within 1 meter, the robot will limit the speed to only 20 percent of its maximum.

1.2.4 Ways of improving industrial safety

Nowadays, as the Intelligent Plant has been carried out and improved, most of the mass production line verges to automation or semi-automation. It is commonly considered that the safety relative to the intelligent manufacturing already reach a sufficient high level which is also a bottleneck so that it is not necessary to further consider the safety issue. However, there are still many accidents caused by these smart robots on the production line. In 29th June 2015, at German Volkswagen manufacturer, a 21 years old worker was suddenly impacted and compressed by the robot which he was debugging and died immediately during the accident. In July the same year, on one assembly line of vehicle component in America, a robot accidentally assembled one part on a worker’s head and smashed his head bone leading to his immediate death[3]. It is obvious that there should be a more strict safety standard and a more efficient way to avoid the industrial accident.

One way to improve the safety level is to develop a system, which monitors the speed and separation between persons and the robot in order to keep the stopping distance of the robot small enough to avoid impacts[11]. In this way, the robot can stop when human enters the dangerous region or when a collision is about to occur. The cost of this system is also very low. However, there comes many disadvantages too. One is the big challenge to stop and then restart the robot in a small time and distance due to the high speed and load. Another issue is that this system cannot tell accurately when to stop the robot or not. Obviously, if the contact area is sharp or small, it will be more dangerous to a round and big touching area. And sometimes under the situation of big area's contact, the stop of the robot will be no longer necessary. It will cause much extra time and power to stop and restart the robot. The safety issue will be overestimated. This characteristic is not suitable for a human-robot collaboration system because it is not precise enough.

Another way is called torque different method. The principle is whenever there is an contact between the robot and the human or other objects, the torque required and the torque provide by the motor will be different. If the difference is big enough, the contact can be classified as an collision. And the motor can be stopped. This method can also stop the robot whenever an impact occurs. The cost is also low since only need to add torque-meter to the system. However, some dangerous will be underestimated. For example, this method cannot avoid the contact between human and the sharp tool on the robot, since it works only after the contact really happens. Besides, the speed of the robot should always be limited in order to be stopped in time.

The concept of using digital twin on safety is to simulate the whole process in a virtual environment which is of high imitation of the actual one. The damage can be detected and be evaluated according to standard whether it is required a stop or path changing of the robot. The detection is more accurate compared with the previous safety system. Meanwhile, the huge amount of time of stopping and restarting of the robot each time human enters the stop area can be saved. By creating digital twin, the stop of the process can even be avoided since there contains all the data and information required to generate another safe working route. The solution will be very flexible and can deal with many different cases. However, if there exists lack of information in the digital twin, some danger may become non-detectable. So the safety system is also required in some special place.

	Detect-ability*	Accuracy*	Cost	Robustness	Flexibility
Safety region system	Too high	Low	Low	High	Low
Torque different method	Low	Medium	Low	Medium	Low
Digital twin	High	High	High	Medium	High

*Those are the detect-ability and accuracy for detecting a danger.

Table 1 Comparison of the characteristics of different safety system

Therefore, by comparison among different methods of safety control of collaborative robot, digital twin application has the best performance in detect-ability and accuracy, which is the most important to realize a real-time virtual interaction system. The cost reducing need to be considered by simplifying the calculation process, and the robustness can be improved by using Torque different method as back-up solution.

2. Data relative to safety and co-robot in digital twin application

During the production and manufacturing process, the most significant safety problem is related to the equipment failure and operation mistakes. Firstly, the temperature of the machine can't be too high, especially the overheating of the control system should be strictly avoided. The lubrication of the moving parts should be maintained, therefore, the temperature again should be acceptable since it will affect the viscosity of the lubricant. Secondly, the operating time, frequency and load should be set reasonable and recorded all the time to predict and avoid the fatigue of the equipment. Furthermore, the most important safety issue is to avoid the direct collision between the robot and human. So that the distance between the robot and the human, the speed should be detected. There should be sensors in different spot of the robot. Not only because the speed difference of various parts, but also due to the difference of shape and material distribution on the robot. For example, if the robot is programmed to stop when the operator enters into a certain region, the region around the sharp stiff tool should be larger compared with the region near the round deform-able arm. It is obvious that after a collision with the same speed and force, a sharp tool will lead to a much bigger damage than a round arm. So a longer braking distance is required.

2.1 Data of robot

There are three kinds of data of a robot arm, geometrical data, kinematic data and dynamic data.

The geometric data contains the dimension and shape of all parts of the robot, as well as the material characteristics. Some data such as the diameter of the robot arm can be neglected in numerical modeling in order to simplify the calculation and guarantee the real-time performance of the digital twin application.

The kinematic data refers to all the information about the movement of the robot arm. In each joint of the robot arm, there is a motor which apply torque to move the arm. The degree of freedom in each joint is normally one, which is the rotation angle of the joint. The angle of rotation, rotational speed and acceleration of each joint constitute the kinematic data. By calculation, the speed and acceleration of all part of the robot can be obtained, which will be used for further calculation.

The dynamic data includes the mass and moment of inertia of the robot, as well we the torque and power applied by each motor. The dynamic data is meaning used for FEM analysis which only works in the background.

These data are all very important as import data to calculate in the digital twin application to avoid safety issues.

2.2 Data relative to human

Compared with the data needed of robot, the data relative to human is quite similar with some exceptions. The dimension, mass, moment of inertial is also needed. However the human body is very complex to model. There will be many approximate treatment of the modelling. At the same time, the degree of freedom of human arm joint is much more complex than a robot. One joint such as shoulder can have three degree of freedom: rotation angle around the upper arm, inclination angle and direction of inclination. All those angle, angular speed and angular acceleration are included in the data needed of human modeling. Besides, there are some constraints of human arm movement. Such as the angle of the elbow between the upper arm and the lower arm can be only change in the range of 0 to 180 degrees. Furthermore, there are many crucial point on human body, of which the position need to be highlighted and mapped on the virtual body.

Another difference from the robot is that the number of joints is low. In other words, even though the degree of freedom of each human body joints in much more complex to model the movement, the whole arm position and movement state is more easy to be caught than the robot arm. For example, if one fix the upper arm and lower arm direction, the position of the elbow is fixed. The same thing can not be done in the robot arm, since there are several joints between two arms. So compared with robot data, which is always same data in each joints, the data concerned in human body is all data at certain joints.

3. Digital twin model creating

The model of the whole working space should be built according to the structure of digital twin application. The model should contains not only the robot, but also the relevant part of the human, as well as the connecting part between them. The workbench and the floor can link the human model and robot model into a same reference system. The material and the mass should be assigned to each component to help decide the collision criteria. The program that controls the actual robot can be also used to control the digital robot. This is convenient because some speed and force data of certain part can be directly read from the control system. Such as the speed and route of the mechanical cutters and the maximum torque of the robot motor. However, these data are far from sufficient for a comprehensive monitoring. It is required to get the speed information of every part of the robot arm, as well as the exact force or pressure of the collision area if it is necessary. This can be achieved by making a simplified numerical modeling of the robot and the human body in the same reference system.

3.1 Robot modeling in digital twin

3D modeling and numerical modeling are both built and used for different purpose. The 3D modeling is built in CATIA and numerical modeling is directly calculated in Matlab. The actual calculation process and modeling process will be done in the part case study.

3.1.1 3D modeling

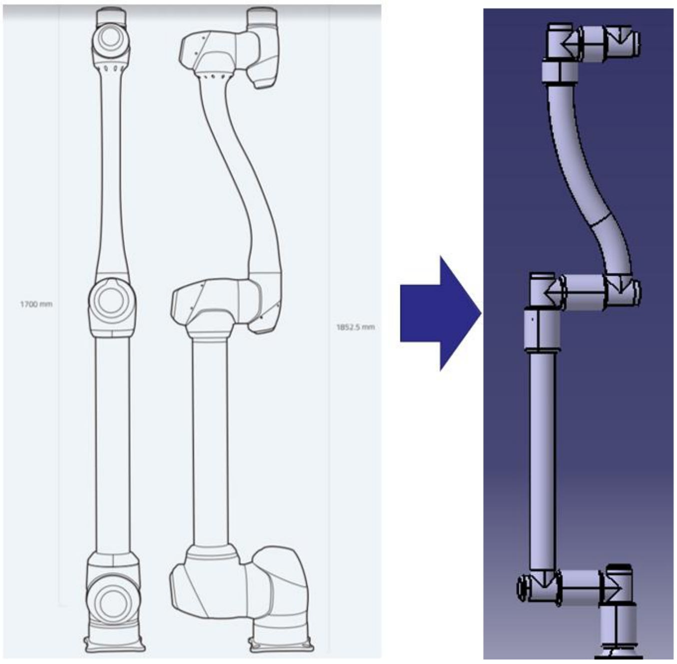


Figure 1 3D modeling of the robot

The 3D modeling of the robot is build for FEM analysis when necessary. According to the useful information includes the number of degrees of freedom, the range of each joint movement, the maximum reach of the robot, the material and the maximum speed of the robot, which can be read from the manual of the robot, an approximate 3D model can be built. The workbench should also be considered, only with a simple display which indicates the position of the robot. The 3D modeling will then be used to FEM analysis in the background to update the data of the control system. Therefore, the 3D modeling and the finite element setting can be as detailed as possible.

3.1.2 Numerical modeling

In order to detect a dangerous collision or even to predict it in time, the calculation of the movement and collision result should be fast enough, which need the amount of processing data not so large. Therefore, apart from the FEM Analysis which requires a lot of time to calculate, a numerical model need to be built for danger detection.

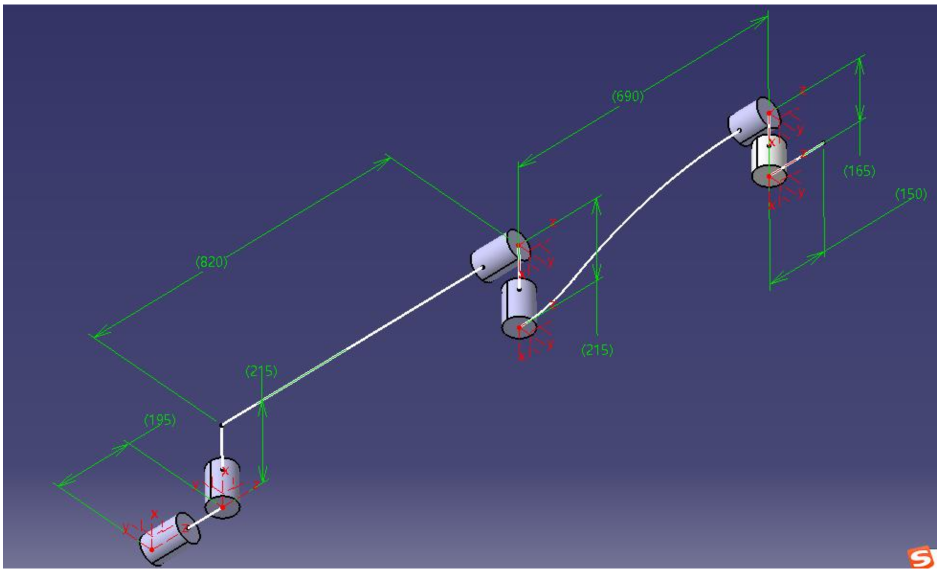


Figure 2 dimension and coordinates of the robot numerical modeling

The first step is to build coordinates for each part between adjacent joints. Then according to these coordinates, the Denavit-Hartenberg's Parameters can be decided. For each degrees of freedom, there will be one parameters which will be the input of the whole digital twin system. And these parameters can be all read from motor control unit. By simulate the real-time movement of robot and human in the virtual environment, when a coming contact is detected. The next step will be to decide whether it is a collision which will lead to injury of human or not. This step need the data of the instantaneous velocity vector of the contact point. This parameter will then be used to obtain an index to indicate whether it is an impact to be avoid which can cause injury to people. The velocity vector of certain point can be obtained by the following procedure:

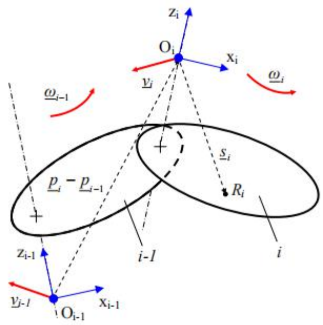


Figure 3 parameters required to calculate the velocity vector

The angular speed of one joint can be calculated by the prior one using formula:

$$\overrightarrow{\omega_{\mathrm{i}}} = \overrightarrow{\omega_{i-1}} + \overrightarrow{q_i} \cdot (1-\delta_i) \cdot \overrightarrow{k_i}$$

In which the q_i is the speed of degree of freedom of joint i, δ_i is the joint i parameter, which is 0 when the joint is revolution and is 1 when the joint is prismatic, ki is the unit vector of axis z_i.

The velocity of joint i origin of its coordinates relative to the reference frame attached to itself can be calculated by:

$$\overrightarrow{\mathrm{v}_i} = \overrightarrow{\mathrm{v}_{i\text{-}1}} + \overrightarrow{\omega_{i\text{-}1}} \times (\overrightarrow{p_i} - \overrightarrow{p_{i-1}}) + \overrightarrow{q_i} \cdot \delta_i \cdot \overrightarrow{k_i}$$

In which the p_i is the position vector of the origin O_i of the reference frame i with respect to the fixed global frame which is frame 0.

Then for calculating any random point on each arm of robot, the velocity can be obtained by using:

$$\overrightarrow{\mathrm{v}_{Ri}} = \overrightarrow{v_i} + \overrightarrow{\omega_i} \times \overrightarrow{s_i}$$

In which the s_i is the position vector of the random point R_i embedded in the link i with respect to frame i.

Other parameters such as angular acceleration, force and torque of the robot can be also calculated using this modeling methods. According to different type of robot, some parameters will become relevant some will not. For example, for normally controlled robot arm, the acceleration is not necessary, since the speed state can be calculated or ready directly by the controlling system of the robot. However, if some machine has function of free movement such as hammer hitting, the acceleration will be come necessary of the prediction of the impact and the post analysis.

3.2 Human modeling in digital twin

To realize a comprehensive simulation of the working statues, the workers should also be properly modeled. The digital mirror of the human body can be called the virtual body. It should contain several functions. The virtual body theoretically should be exactly the same as the physical body. However, as long as human body is extremely complex and many details are irrelevant to the operations, and with a more complex virtual body, the arithmetic will also be more complected to generate more power and cost. So in the point view of efficiency and economy, a minimum modeling including the head, main body part and body joints is necessary. And the virtual body should be able to do a real-time movement tracking and imitation of the human body. Furthermore, the virtual body can interact with the virtual environment and get information about the working and safety statues and show them to our people. In order to satisfy these conditions, the virtual body should be build with enough elements and degree of freedom to guarantee the implement of different movement of human. And the movement tracking should be accurate enough, since very little tolerance can cause serious danger. The robot may aim at the workers hand instead of the mechanical part.

Human modeling also contains 3D modeling and the numerical modeling. A significant part of the difficulty is to effectively get the movement data of human body in a very short time of delay. To summarize, there are three factors which need to be considered of the human modeling. One is to as precisely as possible map the human movement on the digital human model, which need many equipment including position sensor and angle sensor in every joint of human body. The second one is to guarantee the real-time characteristic by either using server with high calculation speed or reducing the calculation amount of data by some simplification. Another one is in the view of the human. It is not possible for a man to do the work with full of sensors on the body, the number of sensors need to be reduced to as less as possible.

3.2.1 Detecting method

Generally, there are quiet large amount of ways to get human body data of virtual modelling.

In order to distinguish human body among surrounding items, a thermopile array sensor is considered. However, it is obvious that there will be much disturbance since most part of the robot will also generate thermal heat. And the heat radiation from surface of human body will lead to a lower accuracy of the detection.

Another way is to use RGB and depth image cameras to get 3D information of human body. There are many drawbacks as well, such as, the color of the environment is limited to avoid disturbance and the light change will also affect the information acquirement. So in the complex working environment of the factory, this methods is also not suitable.

The technology of action recognition is also considered, which is more accurate and efficiency. But at the same time, this method costs too much and is not mature enough to use as a way only to get input information. The most resources of the calculation capacity should be put on the analysis of the movement of the human and robot model and to decide whether to avoid the collision by stopping the robot or other measures.

Therefore, compared to the above methods, instead of using a camera or a sensor from a distance to detect the human movement by observing, a more reliable way is to using some sensors fixed to key points of human body. The sensors can be gravity sensors, gyroscopes, magnetic sensors, and so on. The change of the environment in light and temperature will not easily affect the using of the sensors, and the data is easy to process so that the burden on the system would be less.

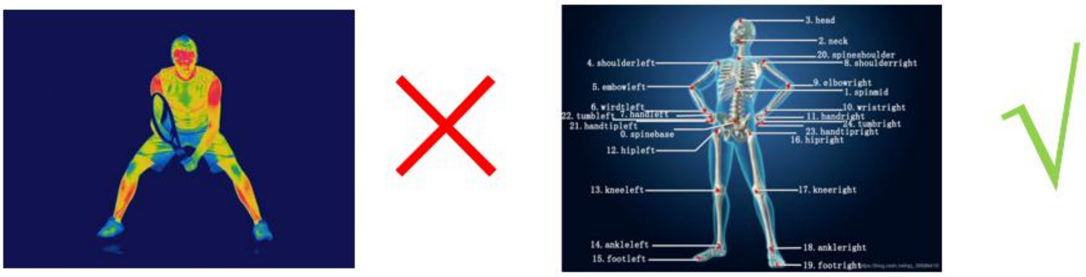


Figure 4 ways of detecting human body

3.2.2 Selection of detecting point

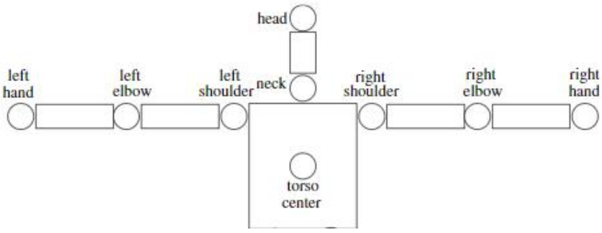


Figure 5 possible detecting point on human body

The detecting point should be selected according to human body joints. There are at least 9 places that has DOF(as shown in the figure) which need to be detected by sensor, neglecting the detection of finger movement of each hand. The feature of hand movement is that hand has small covered space but large number of DOF, so instead of add numbers of sensor to detect each knuckle movement, it is more convenient and easy operated to make the hand as a whole and modeled it according to its covered space. It is same for neglecting the movement of the jaw, change of relative position between torso center and shoulders and so on. The principle of choosing the detecting point is that each point corresponds to one rigid part of human body. For example, one can twist his waist but the upper body can be considered as a rigid part, so there can be only one detecting point to detect the movement of the waist.

3.2.3 Sensor position and function

The sensor position can be selected considering two factors. The first one is that the position is easy to be installed with sensor and the operation of the operator will not be affected a lot. For example, since normally a helmet will be required in a factory, one sensor position can be set on the head. The second factor is to reduce as much as possible the number of the sensors. If there already installed sensors on the head and the upper body, the rotational angle and inclination angle of the neck can be calculated since there is only one way to connect those two part of human body. At the same time, if there are already sensors on the upper body and the neck, the head position and rotational angle can also be calculated.

After considering the two factors mentioned above, among the 9 detecting point mentioned in 3.2.2, five of them can be calculated by data coming from adjacent points. So there are at least 4 points need to be detected for the whole human motion mapping in the upper part of the body. For each points, there can be three kinds of degree of freedom to be detected, which are position, rotation and inclination. So the sensors required includes:

Head: position and rotation angle sensor

Body: position and rotation angle and inclination angle sensor

Arm: on the wrist, detecting the position and the inclination angle

Position of elbow, neck, head and shoulder can be calculated by the rotation and inclination angle of each part based on the position of body and wrist.

The detailed modeling procedure will be discussed in the case study.

4. Standard of the robot control can be used in digital twin

To evaluate whether the human robot collaboration is safe or not, there should be a standard to refer on. There needs to be certification before putting robot into industrial usage. Standards play an important part in this process, to find out what constitutes best practice in safe system behavior, and in design methodology[13].

4.1 Existing standard of robot safety

ISO 10218—the principal safety requirements standard for robots in industrial environments to work autonomously or in collaboration with humans.

The requirements of human robot collaboration in the standard contains five points.

• Stopping functions (10218-1)—requirements are specified for how and when the robot should perform protective or emergency stops when humans are in the robot's workspace.

• Speed and position control (10218-1)—requirements are specified for the maximum allowable speeds of robot arms and end effector when humans are in the robot's workspace.

• Power and force control (10218-1)—requirements are specified for the maximum allowable power and forces applied by robot arms and end effector when humans are in the robot's workspace.

• Design of collaborative operation workspace (10218-2)—requirements are specified for the layout design of workspace around the robot, including safeguarded spaces (where humans are separated from the robot and protected by safeguards) and collaborative spaces where humans are not separated from the robot and hence the robot shall apply the control limits mentioned above.

• Collaborative operation modes (10218-2)—requirements are specified for the specific operating modes that must be designed into the robot's control function when collaborating with a human in the collaborative workspace, including teaching modes and autonomous modes.

These standard can be too restrictive for digital twin application of collaborative robot if put them directly as the index in the controlling program, since the digital twin has a certain capability of prediction. In this thesis, only collaborative spaces are considered, the control limit in the digital twin will be recalculated or using other replaceable criterion in order to satisfy the standard.

4.2 Standard could be used

The injury criteria of automobile crush test can be used, such as: Head Injury criterion(HIC), Neck Injury Criterion(NIC) and so on. These two criterion all need the data of the result of the collision which is very difficult to be calculated in a short delay.

Once the position of contact is detected, the resultant acceleration and speed of the part of human head and spine can be calculated using FEM and then obtain the parameter Injury Index. Then this calculated Injury Index need to be compared with the pre-saved Injury Index Limit. Human body models such as THUMS can reliably be used to robustly predict injuries sustained in real world crashes. The THUMS injury predictions are precise, but it still need time to do the simulation. Therefore a larger DOE variation study is necessary to more fully quantify correlations between the parameters and the injury predictors. Future work could simulate a wide range of impacts to further understand collision parameter effects, robot arm position, speed direction on injury metrics, organ strain metrics, rib fractures, and the effectiveness of injury mitigation systems[14]. Then the simulation results can all be put into a map of injury index matrix and saved in the control unit of the digital twin application.

4.2.1 Brief description of the standard

The Head Injury Criterion(HIC) is shown as below:

$$\mathrm{HIC} = \max\left\{ (t_2 - t_1)\left[\frac{1}{t_2 - t_1} \int_{t1}^{t2} a(t)dt \right]^{2.5} \right\}$$

In which:

a is the resultant head acceleration

t_2-t_1 is less than 36ms

t_2, t_1 is selected so as to maximize HIC.

The HIC limit is set choosing different time interval for different dummy sizes. For example, for a 15 ms interval, the HIC limit is 700 for a large sized male and is 390 for a 1 year old infant. So for large sized male, a much more severe impact can be allowed to occur without causing death or serious injury.

The Neck Injury Criterion(NIC) is shown as below:

$$\mathrm{NIC} = a_{rel} x 0.2 + v_{rel}{}^2$$

where a_{rel} and v_{rel} are the relative horizontal acceleration and velocity between the bottom (T1) and top (C1) of the cervical spine. The constant 0.2 represents the length of the pig neck in meters and was assumed to be also representative of humans. The limit of NIC is set also by experiment with dummy or human volunteers[16].

There are also other injury matrix to determine whether a injury will happen during impact such as Thoracic Trauma Index (TTI), which is calculated from the vertebral body acceleration, and so on.

4.2.2 The Injury Index matrix

The HIC, NIC and TTI are all related to the car impact test, for which the requirement is to keep the passenger or the driver from bad injury or death. Meanwhile, what need for the digital twin application is to avoid any injury including high level of pain which will affect the operator’s working condition. Therefore, these criterion is not strict enough. To solve this problem, one way is to set the limit of these criterion to a more strict value, another way is to use pain study as the basement to make the best standard for injury and pain avoid.

According to reference[15], the pain of different area of human body is related to the strain of the skin, which is also related to some parameters such as impact force, tensile stress, impact energy and so on. These parameters can be tested as an index to build the standard for safety control. The relationship between the pain level and the parameters can be only defined by experiment. For example, with the same impact energy equal to 1.3 J/cm^2, the pain level will be totally different when this impact is happened on the side of the lower arm or on the back of the lower arm, as shown in the figure 6.

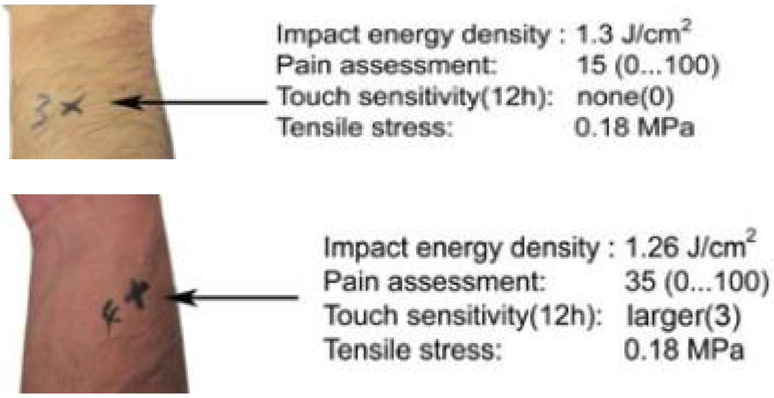

3×
Impact energy density : 1.3 J/cm^2
Pain assessment: 15 (0...100)
Touch sensitivity(12h): none(0)
Tensile stress: 0.18 MPa
4×
Impact energy density : 1.26 J/cm^2
Pain assessment: 35 (0...100)
Touch sensitivity(12h): larger(3)
Tensile stress: 0.18 MPa

Figure 6 pain sensitivity on different part of human body

So the parameter IED need to be further related to the pain level at different area of the human body. The final standard will be set according to this pain level. The pointer and the matrix in between from the dynamic and kinematic information to the final pain level will be the key point for this application.

According to the type of the collaborative robot, the input data is better to be the one that can be directly measured or detected, such as the position on the robot and on the human body, the degree of freedom of the movement, the speed of the moving part and so on. The purpose is to simplify the calculation process of digital twin during daily usage of the collaborative robot.

The input data should includes position on robot, relative speed between human and the robot, position on human body, the torque of robot and so on.

The output data should be the pain level.

The relationship between the input information and the output data is very complex, and contains numerous of different cases, so they can only be decided by simulations or by experiments. Several tests can be designed to get the map. For each map, the independent variable is the relative velocity and the force/torque on the direction of the relative velocity. And the dependent value would be the Impact energy density, which is related to the pain of human.

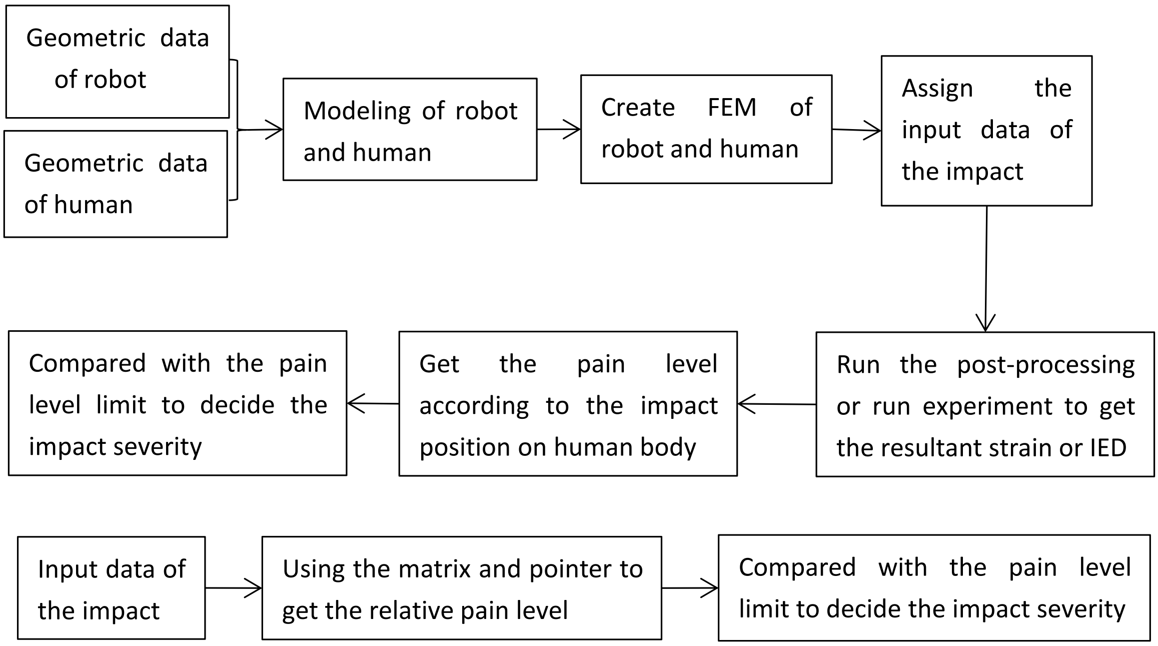

Geometric data of robot
Geometric data of human
Modeling of robot and human
Create FEM of robot and human
Assign the input data of the impact
Run the post-processing or run experiment to get the resultant strain or IED
Get the pain level according to the impact position on human body
Compared with the pain level limit to decide the impact severity
Input data of the impact
Using the matrix and pointer to get the relative pain level
Compared with the pain level limit to decide the impact severity

Figure 7 comparison of the procedure between w/ or w/o big data application

The time for calculation is highly saved by saving relevant information under different conditions in advance in a injury index matrix. This time meaning contains the one for FEM simulation or even for the experimental test. The advantage of time saving is obvious by comparing the flowcharts of the two different methodologies as shown above. As such, the matrix will contains huge amount of information which need always to be updated in the system backgrounds. The time which needed for the pr-treatment as well as the post-processing is saved. The pr-treatment is also considered because the body dimension of human being will be different is the operator is changed. These two steps is not removed from the digital twin application. They will be done by the processing unit when there is free space for them, such as when the robot is switched off or when the robot is moving without a human nearby. The processing unit will calculate and update the matrix to make sure the tolerance within a certain safe range.

5. Logic of digital twin control

The logic should consider the collision and the human guidance at the same time, which means the robot can stop whenever a impact which will cause danger to human occurs, meanwhile, it can also allow the force given by human in order to follow the guidance or allow pushing of button on it. This logic will start whenever a contact between human and robot is detected or predicted. In order to realize a certain predictability, the simulation need to be ran a little further in the time domain. This time ahead should satisfy two requirements. It should be long enough so that larger than the time for the system to calculate the result plus the time for the system to react. This also relate to the maximum speed, mass and nominal power of the robot. On the contrary, it should also be short enough, since the human movement after few seconds can't be predicted at all.

The next step is to derate the motor of the robot. Not only if it is a dangerous collision, but also for a normal guidance action of human, the derating of the motor is necessary. In this case, a dangerous collision becomes not dangerous after the motor derating, it will be seen as a guidance performance of human and the system will keep the Power and Torque low until the contact end, then runs in normal state again. The matrix of injury index is used here to determine the performance and state of each critical point of human body. Then according to the standard relative to human safety, the collision is dangerous or not will be clear.

Following the logic mentioned above, the system will be able to tell when it is a dangerous collision and when it is only a soft contact. But it can't tell the difference between it is the human guidance on purpose or an accidentally touch. Therefore, by using this logic, the robot will derate itself every time there is a contact. In order to separate the accidental touch from the human guidance, a further step is added. There should be a certain area on the robot which is specifically used for human guidance. When the contact detected by the system is apart from this area, a comparison without motor derating will be made, and if it is not dangerous, the derating will not be proceeded and the system is kept in normal working mode. This added logic can save time for some unnecessary derating and re-cranking of the motor.

The logical diagram below in figure 8 is a draft layout of what is mentioned above. During application, more parameters such as the robot type, the calculation capability of the processing unit and so on should be considered. For example, if the robot arm has area on it that is used for operators to press button to apply certain controlling command, this area should be specially considered during the design of the logic diagram.

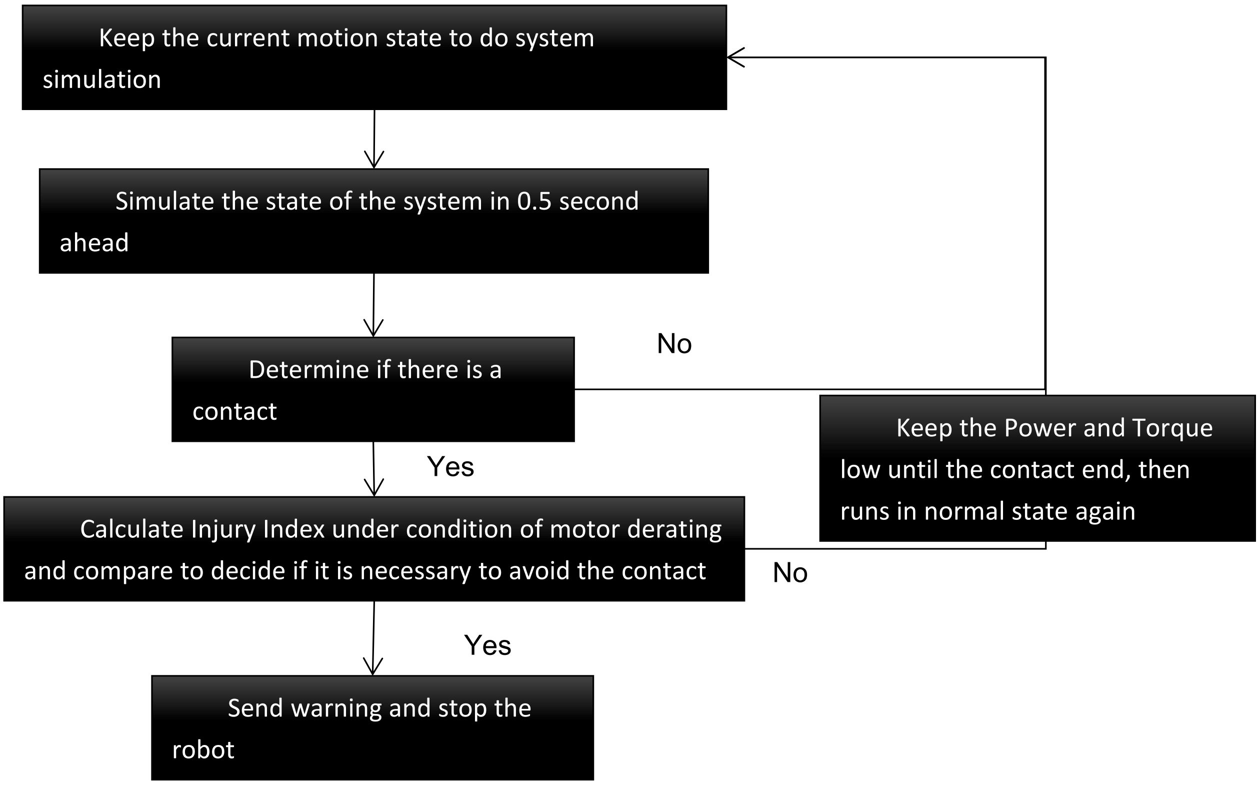

Keep the current motion state to do system simulation
Simulate the state of the system in 0.5 second ahead
Determine if there is a contact
No
Yes
Calculate Injury Index under condition of motor derating and compare to decide if it is necessary to avoid the contact
No
Keep the Power and Torque low until the contact end, then runs in normal state again
Yes
Send warning and stop the robot

Figure 8 logic flow chart of the digital twin application

6. Digital twin application malfunction solution

As the number of sensor has been highly reduced, the margin of error is very low. In this case, if some failure happens, the whole system could be not able to work normally, and the safety can’t be guaranteed anymore.

There are two kinds of possible malfunctions.

The first one is the sensor failure. Since the sensor number is already reduced. The information coming from the failed sensor cannot be replaced by equivalent information from other sensors. So other safety guarantee method such as the Torque different method used as a back up safety system is necessary. There are two ways of detecting a sensor failure in the system. As the sensor layout has no redundancy, if one sensor is failed with no data output from it, the digital twin modeling can’t be built, and will be qualified as an failure. For the sensor failure which have output data but have high tolerance. This Torque different method can be used on the contrary as the detector for the sensor failure. Because the sensor failure can cause the resultant acceleration and torque of the collision simulation much different with the actual one.

The second one is the strategy failure. Such as after long time of usage, the map need to be updated, otherwise there will be high tolerance when calculating Injury Index. To avoid this, the FEM analysis should always be used to simulate and update the map in the backgrounds. However, this will take a large amount of the processing unit and increase the cost a lot, sometimes may even affect the calculation of the numerical system which is not good. In the electric control mode of the control unit, one new mode named updating can be added. The system can get into this mode once one input data is modified and after the first switch off of the robot after that. The system will then start to simulate the FEM analysis using modified input data.

7. Case study

In this chapter, the Doosan Collaborative Robot M0617 is taken as an example to be analyzed and used as the base to build the digital twin application.

7.1 The performance of Doosan Cobots

The Doosan Cobots can already guarantee precise and safe work even in the presence of the operator. There are torque sensors on each single joint, which are the key components of the safety insurance system. The method is the torque different method which has been talked about in chapter 1.2.4. The torque sensors provide high sensitivity in detecting collisions and external forces. The robot also has the ability to remove barriers with intrinsically safe construction and design.

According to the introduction on the website of the company, Cobots Doosan boast state-of-the-art precision force control. The collaborative activities that can be done by cobots are potentially infinite. The system and control unit can handle automatic calculation of process inputs, such as weight and center of gravity. It performs precision tasks quickly and safely. Further more, the wide range of payloads and operating radius allows flexible applications in any workshop.

The Cobos Doosan have a control button (5 buttons) that allows anyone, even without experience, to control the robot arm. The 5 buttons help to select the best way of learning and to save the coordinates of the points.

The following sheet lists most of the mean parameters of the robot.

Performance		
Degree of freedom	6	
Rated payload	6	kg
Max. reach	1700	mm
Tool speed	1	m/s
Repeatability	±0.1	mm
Temperature	5-45	℃
Join Movement		
J1 range/speed	±360/100	° /s
J2 range/speed	±360/100	° /s
J3 range/speed	±360/150	° /s
J4 range/speed	±360/225	° /s
J5 range/speed	±360/225	° /s
J6 range/speed	±360/225	° /s
Other features		
Weight	34	kg
Installation position	Any	

Table 2 basic parameters of the Doosan Cobot

7.2 The existing issues

The current system to guarantee the safety of operator is similar to the torque different method which is mentioned in chapter 1.2.4. The disadvantages is that the danger is sometimes underestimated and the speed of the robot will always be limited in order to be able to be stopped at any time. So the top speed is limited to 1m/s which is not very high and cause the production process slow. And the button position on the robot is at the root of the robot arm, of which the speed is much slower and can the operator can easily operate with the button at this position. In this case, some operation need to be done far from the root of the robot will be difficult.

7.3 Improvement of the safety strategy by digital twin application

The already existing torque different method can be kept as an back up safety system for dealing with the malfunction of the digital twin application. The digital twin application will be added as the main safety strategy in order to achieve the purpose of predict the danger with high flexibility and accuracy.

7.3.1 Robot modeling

According to the dimension and figure of the Doosan Cobots, a 3D model is built in CATIA, with maximum reach equals to 1700 mm. The modeling includes parts creating of arms, joints, inner shafts and housings for motor. Since this thesis is meaning about the method of using digital twin to achieve a way of safety, the detailed part inside the robot like motors and cables can be neglected. Instead the equivalent mass of the robot will be evaluated including the motor moment of inertia when approach the FEM process.

The assembly is shown as below in Figure 9.

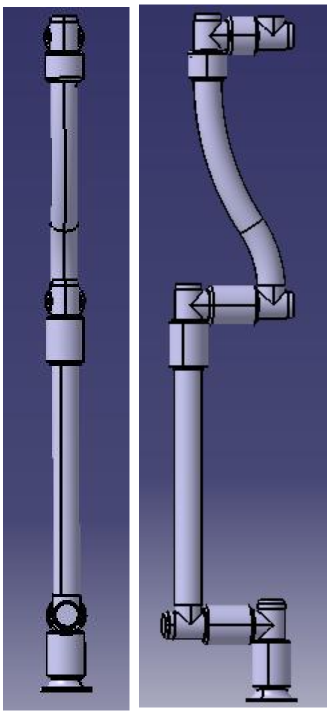


Figure 9 3D modeling of the Doosan Cobot

The 3D modeling is used for calculating the collision resultant performance for deciding the map of collision control. This need large amount of calculation as well as a lot of time, which is not suitable for directly assigning to a real-time system. This calculation is necessary to be made in the back ground in order to update the newly input data to the system. And for guarantee the characteristic of real time, 3D model itself is far from enough, therefore a numerical modeling is built.

The numerical model is built in Matlab. Before creating the model, the first step is to simplify the geometry characteristics of the robot only remain the length of the arm and position of each joints with information of the degrees of freedom.

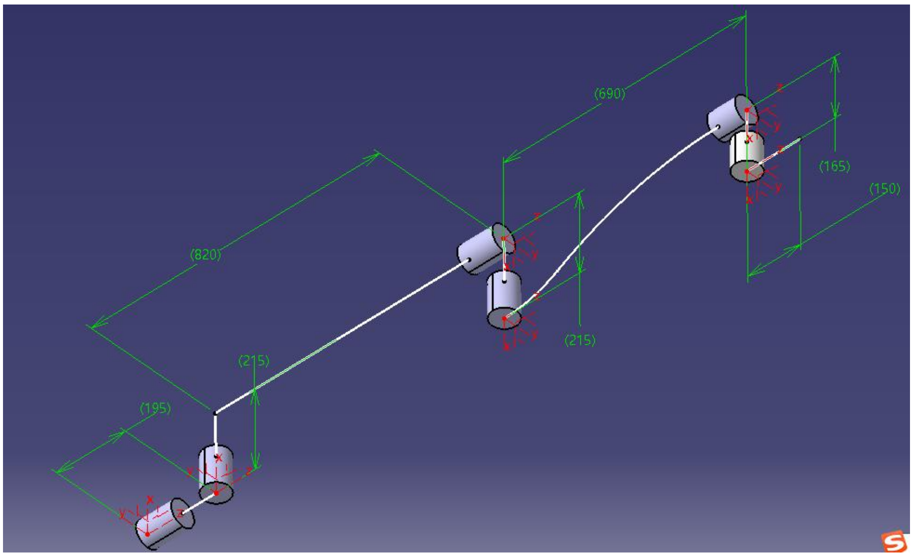


Figure 10 parameters of numerical modeling of Doosan Cobot

Then on each arm, an appropriate choice of the coordinate position is determined. Normally one of x and z axis will be aligned with the rotation direction vector. The Denavit-Hartenberg's parameters are then obtained.

	α	a	d	θ
0_1	0	0	195	q1
1_2	q2	215	820	180+q3
2_3	q4	215	0	0
3_4	0	−165	690	q5
4_5	q6	165	0	0
5_e	0	0	150	0

Table 3 Denavit-Hartenberg's parameters of robot numerical modeling

In this table, qn(n=1, 2, 3, 4, 5, 6) is the degree of freedom of each joint. The number in the first column is the sequences number of each coordinates. For example, 0_1 means from coordinate 0 to coordinate 1, which also means from the robot substrate to the first part of the arm. α_i(i is the raw of the parameters)is the twist angle between axes z_{i-1} and z_i. It is the angle required to rotate the axis z_{i-1} into alignment with the axis z_i in the right-hand sense about axis x_{i-1}. a_{i-1} is the offset distance between axes z_{i-1} and z_i measured along axis x_{i-1}. d_i is transnational distance between axes x_{i-1} and x_i. It is the distance to translate the axis x_{i-1} into incidence with the axis x_i along the positive direction of axis z_i. θ_i is joint angle between the axes x_{i-1} and x_i. It is the angle required to rotate the axis x_{i-1} into alignment with the axis x_i in the right-hand sense about axis z_i.

So as shown in the figure, from coordinates 0 to coordinates 1, the z_0 and z_1 is already aligned together, the distance between x_0 and x_1 is 195 mm, and as the first joint rotates, the angle between the axis x_0 and x_1 will change from -360° to 360° . It is the same for determining other D. H. Parameters. The result is shown in the Table 3.

Then according to these parameters, the numerical model can be built in Matlab. The mean script is shown as in the Annex 1. The green line is to show the route of the end of the robot movement when all the joint of the robot always rotate with the same angle. By setting the degree of freedom as 0 and 360 respectively, the initial and final configuration of the robot can also be calculated and plotted. The figure is shown as below.

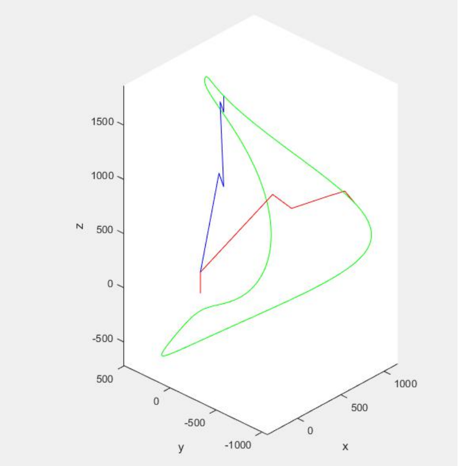


Figure 11 path of the robot effector and the initial and final position of the robot arm

Then to verify the working range of the robot, the trajectory of the end of the robot by inputting the traversal of the range of all the degree of freedom is calculated. The division value of each joint angle is chosen as 45 degrees in order to decrease the calculation time. The script is shown in Annex 2.

If the model is correctly built, by this script, a figure of a sphere with diameter of 1700 mm, which is the maximum reach of the end of the robot, will be plotted. The result is shown in Figure 12.

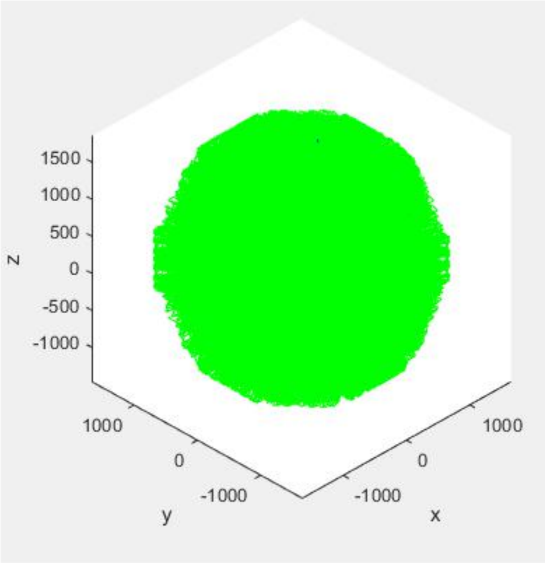


Figure 12 The range and covered space of the robot arm

The next step is to get the speed and acceleration using the motor speed in each joints as the input data. Here the speed of the end of the robot relative to the workbench is calculated as an example.

For simplification, set all the motor speed the same as 30 degrees/s as the input data, which is not exceeding the maximum angular is speed of each motor.

As what has been mentioned in chapter 3.1, the angular speed of one joint can be calculated by the prior one using formula:

$$\overrightarrow{\omega_{\mathrm{i}}} = \overrightarrow{\omega_{i-1}} + \overrightarrow{q_i} \cdot (1-\delta_i) \cdot \overrightarrow{k_i}$$

In which the q_i is the speed of degree of freedom of joint i, δ_i is the joint i parameter, which is 0 when the joint is revolution and is 1 when the joint is prismatic, ki is the unit vector of axis z_i.

The velocity of joint i origin of its coordinates relative to the reference frame attached to itself can be calculated by:

$$\overrightarrow{\mathrm{v}_i} = \overrightarrow{\mathrm{v}_{i\text{-}1}} + \overrightarrow{\omega_{i\text{-}1}} \times (\overrightarrow{p_i} - \overrightarrow{p_{i-1}}) + \overrightarrow{q_i} \cdot \delta_i \cdot \overrightarrow{k_i}$$

In which the p_i is the position vector of the origin O_i of the reference frame i with respect to the fixed global frame which is frame 0.

Then for calculating any random point on each arm of robot, the velocity can be obtained by using:

$$\overrightarrow{\mathrm{v}_{Ri}} = \overrightarrow{v_i} + \overrightarrow{\omega_i} \times \overrightarrow{s_i}$$

In which the s_i is the position vector of the random point R_i embedded in the link i with respect to frame i.

For frame 0 to 5 and e, the x axis of all the frame is parallel to each other, the z axis of all the frame is point to the same direction. As the angular speed of all the joints are set to 30 degrees/s, the speed and angular speed of each joint with respect to the global fame can be calculated according to the frame layout in the figure.

So
$$\begin{aligned}\overrightarrow{\mathrm{v}_E} &= \overrightarrow{\omega_e}\times(\overrightarrow{p_E}-\overrightarrow{p_5})+\overrightarrow{\omega_5}\times(\overrightarrow{p_5}-\overrightarrow{p_4})+\overrightarrow{\omega_4}\times(\overrightarrow{p_4}-\overrightarrow{p_3})+\\ &\overrightarrow{\omega_3}\times(\overrightarrow{p_3}-\overrightarrow{p_2})+\overrightarrow{\omega_2}\times(\overrightarrow{p_2}-\overrightarrow{p_1})+\overrightarrow{\omega_1}\times(\overrightarrow{p_1}-\overrightarrow{p_0})\end{aligned}$$

${}^{0}\omega_{1}$ =[0 0 q1]’=[0 0 30]’;

${}^{1}\omega_{2}$ =[q2 0 0]’=[30 0 0]’;

${}^{2}\omega_{3}$ =[0 0 q3]’=[0 0 30]’;

${}^{3}\omega_{4}$ =[q4 0 0]’=[30 0 0]’;

${}^{4}\omega_{5}$ =[0 0 q5]’=[0 0 30]’;

${}^{5}\omega_{E}$ =[q6 0 0]’=[30 0 0]’;

As the ω_i used in the formula is the one with respect to the global frame, the transformation matrix of each frame with respect to the global frame is required, in which the p_i can be obtained at the same time.

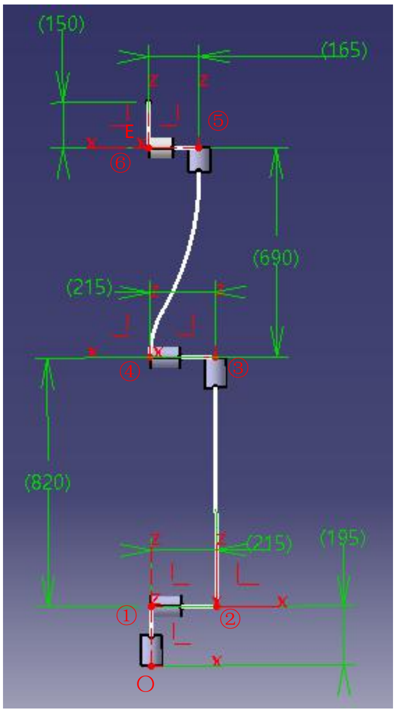


Figure 13 parameters of numerical modeling of Doosan Cobot

The transformation matrix is already calculated in Matlab when build the numerical model of the robot. However, as between frame 1 and 2, there is 2 degrees of freedom which leads the calculation of the speed to a very complicated situation, the frame is chosen again and the new Denavit-Hartenberg’s parameters is shown as below:

	α	a	d	θ
0_1	0	0	195	q1
1_2	q2	215	0	0
2_3	0	0	820	180+q3

3_4	q4	215	0	0
4_5	0	−165	690	q5
5_e	q6	165	150	0

Table 4 Denavit-Hartenberg’s parameters of robot numerical modeling

The transformation matrix of adjacent joints is:

$$A = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & a \\ \sin(\theta)\times\cos(\alpha) & \cos(\theta)\times\cos(\alpha) & -\sin(\alpha) & -d\times\sin(\alpha) \\ \sin(\theta)\times\sin(\alpha) & \cos(\theta)\times\sin(\alpha) & \cos(\alpha) & d\times\cos(\alpha) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

According to the D. H. Parameters, the $^{i-1}A_i$ can be obtained which can transfer frame reference from i to i-1 (i=1 to 6). For example:

$$^{0}A_1 = \begin{bmatrix} \cos(\mathrm{q}1) & -\sin(q1) & 0 & a \\ \sin(q1) & \cos(q1) & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

And if the robot is at the initial position, qi (i=1 to 6) = 0. So

$$^{0}A_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 195 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This is very clear also by directly observing from the figure. From frame 0 to frame 1, without initial rotation, there is only a transverse of 195 mm at the direction of z axis. With the rotation of q1, the matrix will be different.

By extracting the last raw of the transformation matrix, the position vector of each frame origin with respect to the prior frame can be obtained. $^{0}P_1$=[0 0 195 1]’.

With the same procedure, the transformation matrix of each frame with respect to the global frame is listed as below:

$$^{0}A_2 = \begin{bmatrix} 1 & 0 & 0 & 215 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 195 \\ 0 & 0 & 0 & 1 \end{bmatrix} \, ^{0}A_3 = \begin{bmatrix} 1 & 0 & 0 & 215 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1015 \\ 0 & 0 & 0 & 1 \end{bmatrix} \, ^{0}A_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1015 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{0}A_5 = \begin{bmatrix} 1 & 0 & 0 & 165 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1705 \\ 0 & 0 & 0 & 1 \end{bmatrix} \, ^{0}A_E = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1855 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

So the position vector of each frame with respect to the global frame is the last raw of each of these matrix. To collect these position vector in one matrix, the matrix will be like:

$$\begin{bmatrix} 0 & 215 & 215 & 0 & 165 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 195 & 195 & 1015 & 1015 & 1705 & 1855 \end{bmatrix}$$

By formula ${}^{0}\omega_{\mathrm{i}}={}^{0}A_{\mathrm{i\text{-}1}}\cdot{}^{i\text{-}1}\omega_{\mathrm{i}}$, the angular speed of each frame origin can be calculated with respect to the global frame. The velocity of the end of the robot with respect to the global frame is:

${}^{0}\omega_{E}=\begin{bmatrix}2.37 & 0 & 29.76\end{bmatrix}$ rad/s.

And the velocity of the end of robot can be also calculated:

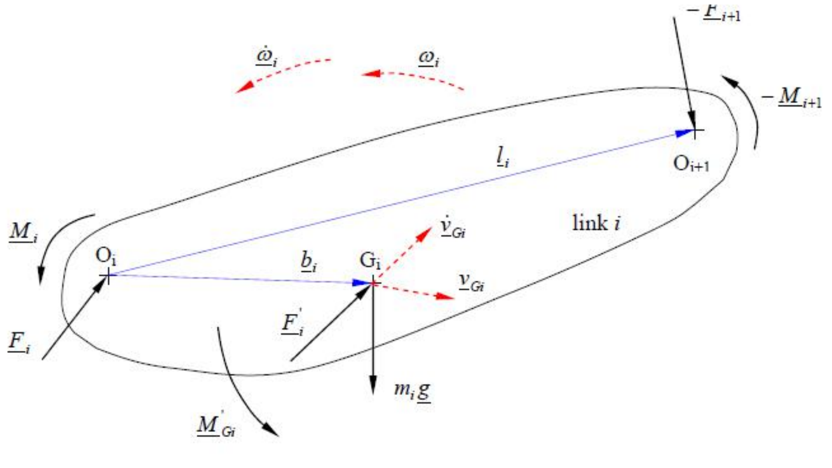
${}^{0}\mathrm{V}_{E}=\begin{bmatrix}0 & -8408 & 0\end{bmatrix}$ mm/s.

The acceleration of the robot is not necessary for predicting the collision, because the movement at every moment is stick to the program of the robot, which can be read directly. Meanwhile the dynamic modeling which can get the information of torque need to be obtained. The reason will be discussed after.

The dynamic equation used to calculate the torque and force is shown as below：

$$\overrightarrow{{}^{\mathrm{i}}F_{i}}={}^{i}A_{i+1}{}^{i+1}\cdot\overrightarrow{F_{i+1}}-m_{i}{}^{i}\cdot A_{0}{}^{0}\cdot g+m_{i}{}^{i}\cdot\overrightarrow{\dot{v}_{Gi}}$$

$$\overrightarrow{{}^{\mathrm{i}}\mathrm{M}_{i}}={}^{i}A_{i+1}{}^{i+1}\cdot\overrightarrow{M_{i+1}}-\overrightarrow{{}^{i}F_{i}}\times\overrightarrow{{}^{i}b_{i}}-({}^{i}A_{i+1}{}^{i+1}\cdot\overrightarrow{F_{i+1}})\times(\overrightarrow{{}^{i}l_{i}}-\overrightarrow{{}^{i}b_{i}})+{}^{i}l_{i}\cdot\overrightarrow{{}^{i}\dot{\omega}_{i}}+\overrightarrow{{}^{i}\omega_{i}}\times({}^{i}l_{i}\cdot\overrightarrow{{}^{i}\omega_{i}})$$



And the mean parameters are shown in the figure above. In order to get the force and torque of each point on the robot arm, the parameters ${}^{i}l_{i}$ and ${}^{i}A_{i+1}$ can be replaced with the parameters relative to the target point.

As the operator will also be modeled and the velocity of each part of human body can be obtained by calculation, the size and direction of the relative velocity between the impact point on human body and the impact point on robot can be get. Then the force can be divided into two directions, one is parallel to the relative velocity vector, the other one is perpendicular to it. This parallel one can then be used as input data for determine the pain level. The equation used to calculate the parallel force is shown as below.

$$\overrightarrow{F_{\mathrm{relative}}} = \vec{F} \cdot \overrightarrow{V_{\mathrm{reletive}}}$$

7.3.2 Operator modeling

The purpose of the numerical modeling of operator is also to predict the contact between operator and the robot, so it is also required to detect the position, velocity and acceleration of each part of human body. For contact detecting, the position sensor is already enough, however, in order to predict the collision in advance, the speed and acceleration is necessary. And since the human joints are much more complex, the sensor setting should be done after the studying of the human joint movement. In this chapter, the detecting of these three factors will all be discussed one by one.

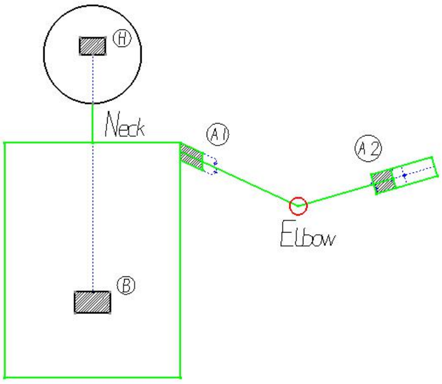


Figure 14 Sketch of human with possible sensor positions

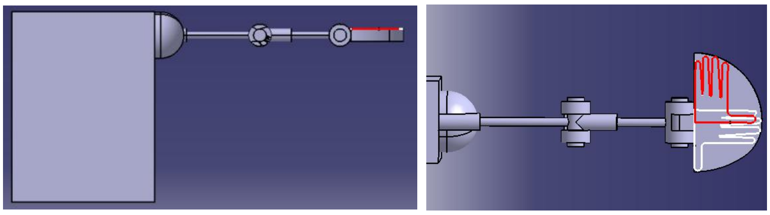


Figure 15 3D modeling of human upper body

The detecting point is shown in the figure and highlighted by section lines, and named according to their position.

The speed of the elbow can be calculated either from the wrist side or from the shoulder side. Since there is no sensor on the shoulder, it is better to get the velocity and acceleration of elbow from the data coming from sensors on the wrist. The neck , the upper arm and the lower arm can always be decided the position by the information coming from the sensor at wrist, body and the head, so if the movement of these three parts can be predicted, the rest of upper body can be predicted as well.

SensorH:

This is the sensor positioned on the head. It should have the ability of detecting the position, faced direction and inclination angle of the head, at the same time, together with the position of human body, it should be able to obtain the state of the neck, such as position and rotation angle.

So for the head movement, without rotation of the neck, it can be divided into two parts. The first is to choose one direction around the head, the second part is to choose the degree of inclination. And for the movement of the neck, it is very complex since it contained many small bones in it. It can rotate around its axis, it can also incline to any direction with any degrees in a large range. The good thing is that, all of these movements mentioned above can be replaced by other equivalent movement of head. Since the head can't rotate without the rotation of the neck, the rotation of head can be detected to reflect the rotation of neck. And since when the neck incline to certain direction with certain degrees, the position of head will move in relative way, these two are exactly linear because if one fix his head, he won't be able to move his neck.

Therefore, to cover the neck and head detecting only by sensors on the head, the function of the sensors should include: angular sensor around neck axis, inclination direction and degree sensor, and position sensor.

SensorB:

This is the sensor positioned on the body, it should be able to detect the position of the upper body of the operator, and can determine the statues of the neck with the information coming from Sensor H.

The movement of the upper body is meaning controlled by waist, and with the same type of movement which can be done by head. So the sensors on human body should also have the functions of angular sensor around waist axis, inclination direction and degree sensor, and position sensor.

SensorA:

This is the sensor for the arm movement. Considering the position and movement statues are already clear, in order to use as less as possible the sensor and to guarantee sensor position won't cause big effect to operator movement, the best place is on the wrist. As shown in the figure, knowing the position of point B and W, the length of the lower arm or the upper arm, as well as the angle of α and β, the movement states of the whole arm can be pictured. Among these parameters mentioned, the position can be obtained by body sensor and the wrist sensor, the angle can also measured at the wrist, and the length of the arm can be input manually before the operator using this robot.

It is quiet clear that, the neck, the upper and lower arm is constrained if the position and inclination angle of wrist, body and the head is fixed. So the whole body movement can always be modeled only by information coming from the sensors at these three position. Further more, by predicting the movement of the head, the body and the wrist, the whole upper body movement can be predicted.

For the hand modeling, there are several solutions to choose. In one hand, there are 15 knuckles in total, which allows the hand to be the most flexible part of each human. So one way is to detect the movement of each finger joints using sensors. This method requires at least 10 sensors by calculation. And the hand movements can be affected to some extent. Another way is to modeling the hand movement covered space as a whole. The advantage is that the hand won't be affected by the sensors, but the impact simulation and prediction will be over estimated, since the hand model will be more easy to impact with the robot than the real hand.

Take the calculation of position of the elbow and upper arm by wrist and body sensor as an example. The position of the shoulder can be easily obtained by human body dimension and the states of the waist from the sensor. At the same time the position of the wrist and the direction of the lower arm can be read also from the sensor. By calculating considering also the length of the lower arm, the position of the elbow is obtained, as well as the position and direction of the upper arm.

7.3.3 Map of control

The control map contains several sub matric in it. First, considering different position of human body will have different pain sensitivity, and the different point on robot arm have different stiffness and function, there should one matrix to decide on which position the contact point is of the human body and robot arm. The human body can be divided according to the main point relative. The robot arm can be separated into different groups.

	Bot_1	Bot_2	Bot_3							Bot_n
$Head_1$	MapH1Bo1									
$Head_m$										
$Body_1$										
$Body_i$										
Arm_1										
Arm_j										MapA1Bon

Table 5 Matrix that stores the map impact of each human and robot point

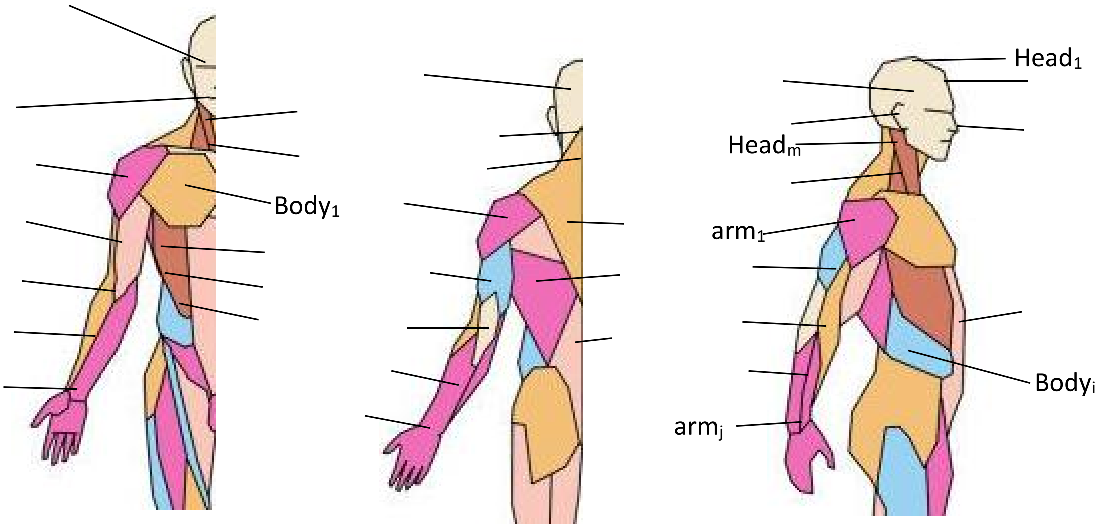


Figure 16 the contact point concerned on human body

The main point on human body can be determined following two steps. The first step is to cover the whole body with detecting points with equal space between each of them. Such as on the upper arm, 4X3 points are set from shoulder to the elbow. On shoulder, elbow and the position in the center of the upper arm, four points around the arm are chosen for each position due to different pain sensitivity on each side of the arm. The next step is to add the point on the critical positions on human body, such as the temple on the side of the head, which is very dangerous for the operator during an impact. So the result points are listed in the table and the layout is shown in the figure.

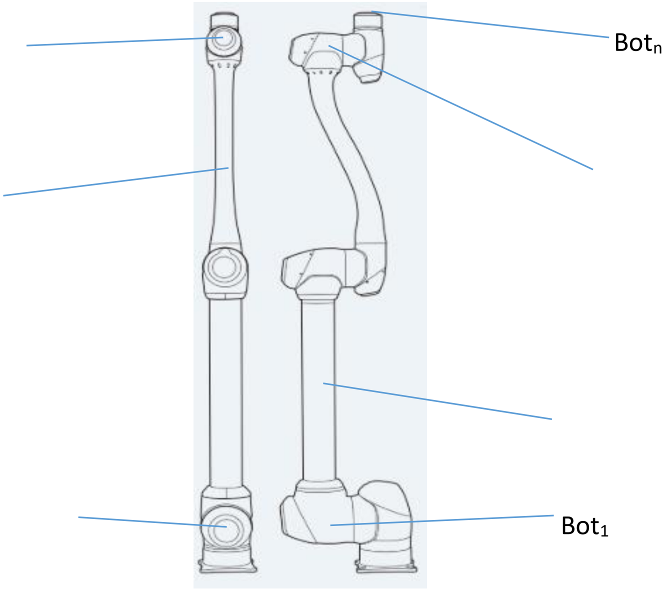


Figure 17 the contact point concerned on robot

For the point on the robot, they can be grouped according to the shape and material. For example, the whole robot lower arm can be grouped together. Since on each position of the arm, the impact with the same relative speed and with the same relative force, will lead to the same impact result. As shown in the figure, along this part of the robot arm, the matrix for each length of it is compared with each other. The matrix indicates the relationship between the relative data includes velocity and forces and the Impact energy density. By the numerical modeling of robot and human, the FEM analysis can only considering a very small inch of the part. As the shear force, the nominal force, the toque, as well as the speed of this small segment is obtained. So the map only can be grouped together since it depends on the geometry and material of the segment, which is the same all along the whole lower arm.

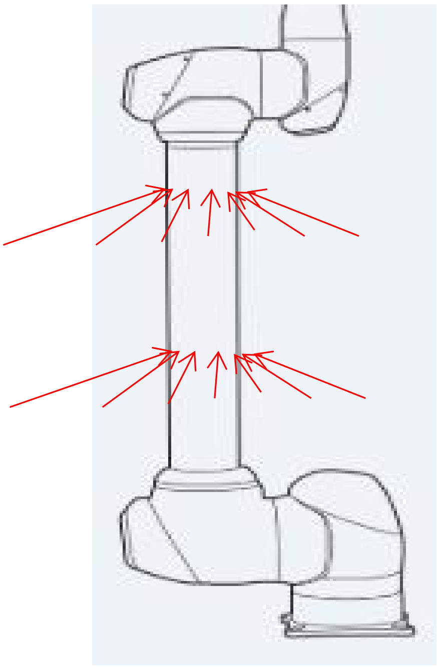


Figure 18 forces along the robot arm

Another factor which can be simplified is the contact mode between the human and the robot. There are three kinds of contact mode, point contact, line contact and surface contact. In practical, considering the deformation of human skin and robot cover, the contact should be a surface with small area. Then the force effect on the contact area is very complex and not easy to calculate the result of the impact. So the contact mode is simplified as point contact on surface. In this kind of impact, the relative velocity can be divided into two directions, one is perpendicular to the surface and the other one is tangent to the surface. As shown is the figure, the red vector which indicates the direction and dimension of the relative velocity, can be divided into the perpendicular one which is the green vector and the tangent one, which is colored in purple. The effect of the tangent velocity can be neglected and only the perpendicular one is considered. Therefore, if the position of the impact point is determined, the direction of the relative velocity will only be considered in the perpendicular one, and the dimension of the velocity will be the only parameter that matters to affect the resultant of the impact.

The perpendicular velocity can be calculated simply by multiplying the relative velocity with the perpendicular vector. As the contact point is known and the shape of the surface is modeled, the perpendicular vector can be obtained as $\overrightarrow{n}$. The the perpendicular velocity can be calculated with the following formula:

$$\overrightarrow{V_n} = \overrightarrow{V} \cdot \overrightarrow{n}$$

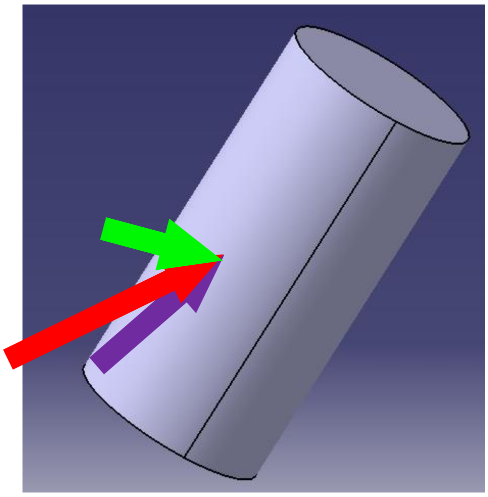


Figure 19 relative velocity divide into two directions

As what is mentioned above, the independent values of each matrix are the relative velocity between robot and human, and the relative force at the direction of the relative velocity. The dependent value is the impact energy density, which can be calculated by the FEM analysis or the experimental test. For example, the MapH1Bo1 will be the matrix for impact between point Head1 on human and the Bot1 on robot. And to do the FEM analysis, the velocity of each impact point on human and on the robot should be calculated respectively.

Then the force and torque should be calculated. The impact point should be picked up by cutting the model on two side, and the force and torque at the cutting section should be the one which need to be concerned.

Then by FEM analysis, the input data should contain the model of the two truncated entity and the force and torque at each section, as well as the relative speed, the result is the Impact density energy which will be related to the pain level of human.

The result of the FEM should all be simulated and stored in the control unit in order to same the time of FEM analysis during the digital twin application and to guarantee the characteristics of real time. The map which relates to the relative speed and the relative force is shown as below. For each pair of impact point, there will be one certain IDE map for the control unit to read the resultant IDE by the calculated relative force and relative velocity.

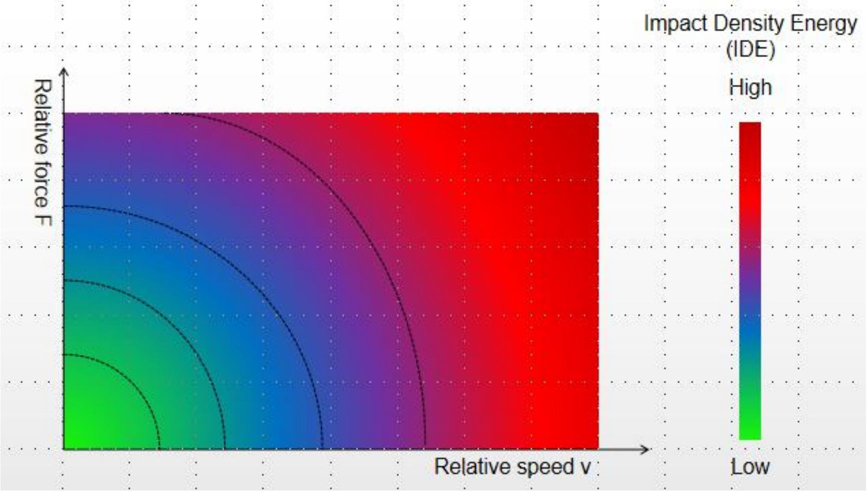


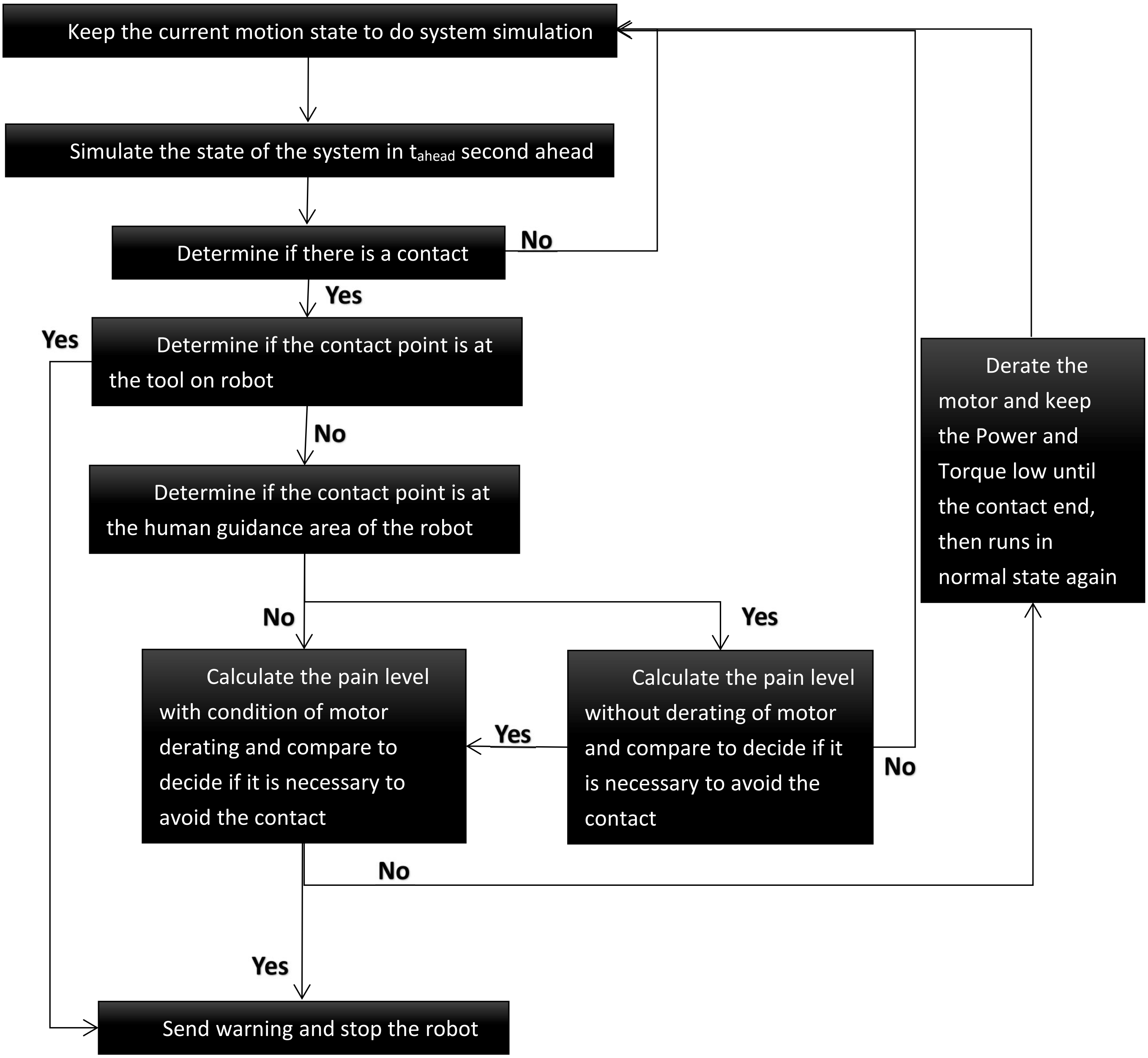
Figure 20 possible layout of map of impact

The IED itself is not sufficient enough to decide whether it is an impact that will lead to pain and injury to human. With the same IED, the pain level will be different on different part of human body. So a coefficient of pain should be set according to the pain sensitivity of different part of human body. As the previous matrix is also related to human body position, the output data IED can then directly transfer into the pain level by multiplies this coefficient.

$$\text{Pain Level} = \text{IED} \cdot \xi_{\text{Pain sensitivity}}$$

Then this pain level can be used to compare with the pain level limit settled according to the safety standard of human robot collaboration and also the position of each point on the human body.

7.3.4 Logic of control



Keep the current motion state to do system simulation
Simulate the state of the system in t_ahead second ahead
Determine if there is a contact
No
Yes
Determine if the contact point is at the tool on robot
Yes
No
Determine if the contact point is at the human guidance area of the robot
No
Yes
Calculate the pain level with condition of motor derating and compare to decide if it is necessary to avoid the contact
Yes
Calculate the pain level without derating of motor and compare to decide if it is necessary to avoid the contact
No
Derate the motor and keep the Power and Torque low until the contact end, then runs in normal state again
No
Yes
Send warning and stop the robot

Figure 21 logic flow chart of digital twin control with Doosan Cobot

The logic is shown as above, which maximize the advantages of digital twin, which is predictability and accuracy. The prediction is done by the first three steps, as the modeling is fully built and all the movement data is obtained either by sensors or read from robot software, the movement states can be predicted in a short period ahead. This ahead period t_{ahead} should be in a range with relate to the reaction time of human and the calculation time required by the digital twin system.

$$t_v + t_F + t_{coml} + t_{comh} + t_{cmd} \langle t_{ahead} \langle t_{reaction}$$

In which t_v is the time for calculate the relative velocity;

t_F is the time for calculate the relative force along the direction of relative velocity. This time includes both the calculation at normal operation condition and at derating working condition.

t_{coml} is the time required to do the comparison between the resultant pain level and the pain level limit under derating condition;

t_{coml} is the time required to do the comparison between the resultant pain level and the pain level limit under normal condition;

$t_{reaction}$ is the reaction time of human. This limit is to increase the predictability of human movement, since if the time ahead is too large, the human movement will become unpredictable.

The other steps are set to determine whether an impact occurs or not. The most important issue in this part is to separate the human guidance to the accidentally contact. Or the system won't work correctly. For example, the motor will derate and the robot will slow down every time the human touch the button on it, and if the human guidance is recognized as an impact which will cause pain and injury, the robot will stop and send warning. To avoid these situations from happening, an area of human guidance is designed on the robot, and the system can run different algorithm in this area.

8. Analyze of the results

The time ahead can only be set in a very small range, since it can’t be larger than the reaction time of human, which is already very low. It also need to be larger than the calculation time and process time of the system control unit, which depends also on the working environment, and can be sometimes even larger than the human reaction time.

The hand modeling as a whole lead to an over estimation of the impact severity. Since the system will consider it as an contact even your finger is not contact with the robot. This will affect a lot the performance of human guidance, and should also be considered in the logic by separating the situation according to the operation area.

There are many maps stored in the system, which depends on the geometric and material of robot and human body. Therefore as the operator change to another person, the whole map should be changed, which will cost a lot of time. One way for this is to store the map of human in different height in advance, by the operator’s height, the right series of matrix will be chosen. Another way is to let the operator wear the equipment and do certain movement to let the system calibrate the data in the maps.

The digital application in this thesis is far from completed. In terms of accuracy, the numerical modeling and the modeling of the hand is simplified to make the calculation more easy. The same for the modeling of the human body, the thickness and shape of the body and arm is not considered. These factors can be neglected compared with other parameters in most cases, but with the operations which need high accuracy, this digital twin application methodology need to be improved. In terms of the real time characteristic, the usage of IDE map and the simplification of the modeling has highly increase the efficiency of the calculation and reduce the reaction delay of the control unit. This delay is still far from achieving the real time effect, further efforts need to be taken to decrease this time.

9. Conclusions

9.1 The digital twin application in this thesis

This thesis discussed a way of safety improvement of collaborative robot environment by digital twin application. With 3D modeling, numerical modeling, FEM analysis and big data processing ability, the digital twin application can be built. By storing the numerous data of the result of collision into the control unit and set the limit on parameters that relative to the standard of safety, the digital twin can simulate in ahead of time to determine whether a dangerous impact is about to happen or not, and can control the physical robot in a reasonable logic to avoid the danger as well as save the time.

The big data stored in the control unit is the key point to achieve real-time characteristic of the digital twin application, since the calculation should be easy enough. The compromise is made with obtaining a more accurate virtual simulation of collision, which needs a lot of computations costing time. So the FEM system will be run in back ground to update the data and guarantee the accuracy of the system

For the human model in the digital twin, the more sensor on human, the more accurate the virtual simulation of human body movement. However, this will also increase the difficulty of human operation. The number of the sensors is then be reduced to minimum on the premise of detecting the complete movement of human body. A compromise layout is used by only assign sensors on head, waist and the wrist, which highly reduced burden and weight on the operator.

By reasonably setting the Injury Index Limit in the map, and cooperate with the logic of the program in the control unit, the possible human guidance action can be done satisfying the collaborative robot safety standard. All the impact situations with different velocity and position are divided into groups according to the relative velocity and the shape characteristics of the robot and human. This is a very effective and valid way to reduce the calculation time of the digital twin application.

All in all, the digital twin application is a strong method which can be used to guarantee the safety environment of human robot collaboration by predicting the danger in the virtual space and control the robot in physical space to avoid the dangerous accident from happening.

9.2 Future work

Further decrease the time between the detecting of contact and action by the control unit. The shorter this time, the stronger the predictability. Since the time ahead can also be reduced, and the movement of the operator will be more predictable.

The hand modeling in this thesis is not discussed in detail. However, fingers is the part of human body that have the most possibility of contact with the robot. So detail modeling should be also be added in the future, with detecting method which will not disturb or influence the movement of the fingers.

The thickness of human body is not considered in this thesis, which will lead to tolerance when determine the impact severity. If the operator is fatter than the model, the impact will be underestimated, on the contrary, if the operator is thin, the impact will be overestimated, the data of the detailed human body of the operator should be considered in the future.

The constrains of degree of freedom of human joints movement should be considered when run the FEM analysis to get the Impact Density Energy. As with the same position of contact, the same relative speed and the same relative force and torque, if the operator can not move his body along the direction of the force applied, the damage will be mach bigger than the case the operator can move his body freely along the direction of the force. This can be considered add another parameter multiply to the final index, which is related to the constrains of all the human body joints.

Acknowledgement

Thanks to the supervisor of this thesis Professor DEMICHELA MICAELA for providing me with the general idea, and correcting me when I study into the wrong direction. The online library resources of Polytechnic University of Turin have provided me with many references and help for this thesis. At the same time, I would like to thank the leaders of the internship company for their support and encouragement of my graduation project. They have concerned about the progress of my thesis many times and reduced my work stress so that I can have enough time to study and finish this project.

Reference

［1］ Söderberg, R., Wärmefjord, K., Carlson, J. S., & Lindkvist, L. (2017)

［2］ "Minds + Machines: Meet A Digital Twin". Youtube. GE Digital. Retrieved 26 July 2017.

［3］ The Development Trend of Mechanical Safety-Intelligent Security. Liu Zhiyong, Li Qin

［4］ BOSCHERT S. ROSEN R. Digital twin-the Simulation aspect[M]//Mechatronic Futures. Berlin, Germany: Springer-Verlag, 2016

［5］ GRIEVES M. VICKERS J. Digital twin: mitigating unpredictable, undesirable emergent behavior in complex systems[M]//Transdisciplinary Perspectives on Complex Systems. Berlin, Germany: Springer-Verlag, 2017

［6］ TUEGEL EJ,INGRAFFEA AR,EASON TG, et al. Reengineering aircraft structural life prediction using a digital twin[J]. International Journal of Aerospace Engineering, 2011 DOI: 10. 1155/2011/154798

［7］ SIEMENS. The digital twin[EB/OL].(2015-11-17)

［8］ FOURGEAU E, GOMEZ E, ADLI H, et al. System engineering workbench for multi-views systems methodology with 3DEXPERIENCE Platform.[M]//Complex Systems Design & Management Asia. Berlin, Germany: Spring International Publishing, 2016

［9］ FLORIAN B. DAVIS M. A digital twin for production planing based on cyber-physical systems[M]//ScienceDirect. Gulf of Naples, Italy, 2018

［10］ NIKI K. CHRISTOS G. Digital twin for adaption of robots' behavior in flexible robotic assembly lines

［11］ TIMO M. TIMO S. ILARI M. JARI M. Dynamic safety system for collaboration of operators and industrial robots.//Open Engineering:formerly Central European Journal of Engineering, 2019

［12］ ISO/TS 15066:2016(en) Robots and robotic devices — Collaborative robots

［13］ Christopher Harper. Gurvinder Virk. Towards the Development of International Safety Standards for Human Robot Interaction//Springer Science & Business Media BV 2010. Published online: 26 June 2010

［14］ Adam J. , Kerry A. Injury prediction in a side impact crash using human body model simulation//Accident Analysis and Prevention 64 (2014) 1 – 8

［15］ B. Povse, S. Haddadin. A tool for the evaluation of human lower arm injury: approach, experimental validation and application to safe robotics.

［16］ Arthur C. Croft. The neck injury criterion: future considerations//Accident Analysis and Prevention 34 (2002) 247 – 255. accepted 1 January 2001

Annex 1

```
%% computation of trajectory of point EE
for i=1:361
q1=(i-1)*pi/180; q2=(i-1)*pi/180; q3=(i-1)*pi/180; q4=(i-1)*pi/180; q5=(i-1)*pi/180;
q6=(i-1)*pi/180; % [rad], [rad],[m] % define the joint variables
%transformation matrixes and vectors
teta1=q1;
A10o=denhar_en01(alfa1,a1,d1,teta1); % (homogeneous) matrix 4x4: positioning of ref frame %O1
with respect to frame O0
alfa2=q2; teta2=q3;
A21o=denhar_en01(alfa2,a2,d2,teta2+180); % (homogeneous) matrix 4x4: positioning of ref
frame %O2 with respect to frame O1
alfa3=q4;
A32o=denhar_en01(alfa3,a3,d3,teta3); % (homogeneous) matrix 4x4: positionining of ref frame %O3
with respect to frame O2
teta4=q5;
A43o=denhar_en01(alfa4,a4,d4,teta4); % (homogeneous) matrix 4x4: positionining of ref frame %O4
with respect to frame O3
alfa5=q6;
A54o=denhar_en01(alfa5,a5,d5,teta5); % (homogeneous) matrix 4x4: positionining of ref frame %O5
with respect to frame O4
A65o=denhar_en01(alfa6,a6,d6,teta6); % (homogeneous) matrix 4x4: positionining of ref frame %O6
with respect to frame O5
A20o=A10o*A21o; % (homogeneous) matrix 4x4: positioning of ref frame O2 with respect to %frame
O0
A30o=A10o*A21o* A32o; % (homogeneous) matrix 4x4: positioning of ref frame O3 with respect
to %frame O0
A40o=A10o*A21o* A32o*A43o; % (homogeneous) matrix 4x4: positioning of ref frame O4 with respect
to %frame O0
A50o=A10o*A21o* A32o*A43o*A54o; % (homogeneous) matrix 4x4: positioning of ref frame O5 with
respect to %frame O0
A60o=A10o*A21o* A32o*A43o*A54o*A65o; % (homogeneous) matrix 4x4: positioning of ref frame O6 with
respect to %frame O0
pEE0o(:,i)=A60o*EE3o; %position of point EE in frame O0
end
%% plot trajectory of point EE
figure(1)
plot3(pEE0o (1,:), pEE0o (2,:), pEE0o (3,:),'g');
xlabel('axis x')
ylabel('axis y')
zlabel('axis z')
legend('trajectory EE');
```

```
%% wire-frame representation of the multibody system
% initial configuration
i=1;
q1=(i-1)*pi/180; q2=(i-1)*pi/180; q3=(i-1)*pi/180; q4=(i-1)*pi/180; q5=(i-1)*pi/180;
q6=(i-1)*pi/180; % [rad], [rad],[m] % define the joint variables
%transformation matrixes and vectors
pOo=[0 0 0 1]';                      %position vector of the global frame origin in the global
frame 0
teta1=q1;
A10o=denhar_en01(alfa1,a1,d1,teta1); % (homogeneous) matrix 4x4: positioning of ref frame %O1
with respect to frame O0
pO10o=A10o(:,4);
alfa2=q2;
A21o=denhar_en01(alfa2,a2,d2,teta2); % (homogeneous) matrix 4x4: positioning of ref frame %O2
with respect to frame O1
A20o=A10o*A21o;
pO20o=A20o(:,4);
teta3=q3+pi;
A32o=denhar_en01(alfa3,a3,d3,teta3); % (homogeneous) matrix 4x4: positionining of ref frame %O3
with respect to frame O2
A30o=A20o*A32o;
pO30o=A30o(:,4);
alfa4=q4;
A43o=denhar_en01(alfa4,a4,d4,teta4); % (homogeneous) matrix 4x4: positionining of ref frame %O4
with respect to frame O3
A40o=A30o*A43o;
pO40o=A40o(:,4);
teta5=q5;
A54o=denhar_en01(alfa5,a5,d5,teta5); % (homogeneous) matrix 4x4: positionining of ref frame %O5
with respect to frame O4
A50o=A40o*A54o;
pO50o=A50o(:,4);
alfa6=q6;
A65o=denhar_en01(alfa6,a6,d6,teta6); % (homogeneous) matrix 4x4: positionining of ref frame %O6
with respect to frame O5
A60o=A50o*A65o;
pO60o=A60o(:,4);
punti_om=[pOo pO10o pO20o pO30o pO40o pO50o pO60o];
punti=punti_om(1:3,:);
% plot initial configuration
 hold on
figure(1)
plot3(punti(1,:),punti(2,:),punti(3,:),'b'),grid
```

```
axis equal
xlabel('x');ylabel('y');zlabel('z')
joint_rev_01(.1,.3,20,A10o,'blu')
joint_rev_01(.1,.3,20,A20o,'blu')
joint_rev_01(.1,.3,20,A30o,'blu')
% final configuration
i=46;
q1=(i-1)*pi/180; q2=(i-1)*pi/180; q3=(i-1)*pi/180; q4=(i-1)*pi/180; q5=(i-1)*pi/180;
q6=(i-1)*pi/180; % [rad], [rad],[m] % define the joint variables
%transformation matrixes and vectors
pOo=[0 0 0 1]';                     %position vector of the global frame origin in the global
frame 0
teta1=q1;
A10o=denhar_en01(alfa1,a1,d1,teta1); % (homogeneous) matrix 4x4: positioning of ref frame %O1
with respect to frame O0
pO10o=A10o(:,4);
alfa2=q2; teta2=q3;
A21o=denhar_en01(alfa2,a2,d2,teta2+180); % (homogeneous) matrix 4x4: positioning of ref
frame %O2 with respect to frame O1
A20o=A10o*A21o;
pO20o=A20o(:,4);
alfa3=q4;
A32o=denhar_en01(alfa3,a3,d3,teta3); % (homogeneous) matrix 4x4: positionining of ref frame %O3
with respect to frame O2
A30o=A20o*A32o;
pO30o=A30o(:,4);
teta4=q5;
A43o=denhar_en01(alfa4,a4,d4,teta4); % (homogeneous) matrix 4x4: positionining of ref frame %O4
with respect to frame O3
A40o=A30o*A43o;
pO40o=A40o(:,4);
alfa5=q6;
A54o=denhar_en01(alfa5,a5,d5,teta5); % (homogeneous) matrix 4x4: positionining of ref frame %O5
with respect to frame O4
A50o=A40o*A54o;
pO50o=A50o(:,4);
A65o=denhar_en01(alfa6,a6,d6,teta6); % (homogeneous) matrix 4x4: positionining of ref frame %O6
with respect to frame O5
A60o=A50o*A65o;
pO60o=A60o(:,4);
punti_om=[pOo pO10o pO20o pO30o pO40o pO50o pO60o];
punti=punti_om(1:3,:);
% plot final configuration
 hold on
```

```
figure(1)

plot3(punti(1,:),punti(2,:),punti(3,:),'r'),grid

axis equal

xlabel('x');ylabel('y');zlabel('z')

joint_rev_01(.1,.3,20,A10o,'red')

joint_rev_01(.1,.3,20,A20o,'red')

joint_rev_01(.1,.3,20,A30o,'red')
```

Annex2

```
%% computation of trajectory of point EE
for i=1:8
    q1=i*45*pi/180;
    teta1=q1;
A10o=denhar_en01(alfa1,a1,d1,teta1); % (homogeneous) matrix 4x4: positioning of ref frame %O1
with respect to frame O0
for j=1:8
    q2=j*45*pi/180;
    for k=1:45:361
    q3=(k-1)*pi/180;
    alfa2=q2; teta2=q3;
A21o=denhar_en01(alfa2,a2,d2,teta2+180); % (homogeneous) matrix 4x4: positioning of ref
frame %O2 with respect to frame O1
A20o=A10o*A21o; % (homogeneous) matrix 4x4: positioning of ref frame O2 with respect to %frame
O0
for l=1:8
    q4=l*45*pi/180;
    alfa3=q4;
A32o=denhar_en01(alfa3,a3,d3,teta3); % (homogeneous) matrix 4x4: positionining of ref frame %O3
with respect to frame O2
A30o=A10o*A21o* A32o; % (homogeneous) matrix 4x4: positioning of ref frame O3 with respect
to %frame O0
for m=1:8
    q5=m*45*pi/180;
    teta4=q5;
A43o=denhar_en01(alfa4,a4,d4,teta4); % (homogeneous) matrix 4x4: positionining of ref frame %O4
with respect to frame O3
A40o=A10o*A21o* A32o*A43o; % (homogeneous) matrix 4x4: positioning of ref frame O4 with respect
to %frame O0
for n=1:8
    q6=n*45*pi/180; % [rad], [rad],[m] % define the joint variables
    alfa5=q6;
A54o=denhar_en01(alfa5,a5,d5,teta5); % (homogeneous) matrix 4x4: positionining of ref frame %O5
with respect to frame O4
A50o=A10o*A21o* A32o*A43o*A54o; % (homogeneous) matrix 4x4: positioning of ref frame O5 with
respect to %frame O0
A65o=denhar_en01(alfa6,a6,d6,teta6); % (homogeneous) matrix 4x4: positionining of ref frame %O6
with respect to frame O5
A60o=A10o*A21o* A32o*A43o*A54o*A65o; % (homogeneous) matrix 4x4: positioning of ref frame O6 with
respect to %frame O0
pEE0o(:,100000*i+10000*j+1000*k+100*l+10*m+n)=A60o*EE3o; %position of point EE in frame O0
```

```
end

end

end

end

end

end
```