

**KERİM MORAL**

**518191043**

**Inverse Kinematic Solutions with Paden-Kahan SubProblems**

**for a Specified Robotic Platform**

**KOM 513E – Modelling and Control of Robots**

**Prof. Dr. Hakan TEMELTAŞ**

Contents

[1. Introduction 2](#_Toc75549162)

[2. Calculation of Inverse Kinematics 3](#_Toc75549163)

[Solving 3](#_Toc75549164)

[Solving 4](#_Toc75549165)

[Solving 5](#_Toc75549166)

[Solving and 5](#_Toc75549167)

[Solving 6](#_Toc75549168)

[3. GUI Design and Simulations on RoboDK 7](#_Toc75549169)

## Introduction

The purpose of this project is to find all kinematic solutions of a GSK-RB20 robotic arm with Paden-Kahan sub problems. Robotic arm can be seen on Figure 1.

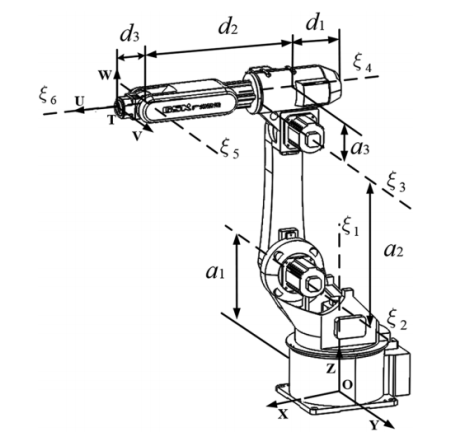
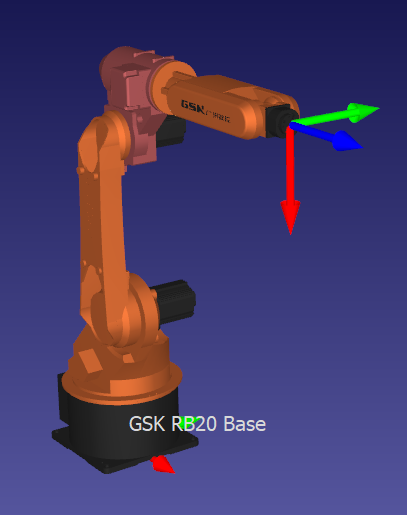


Figure 1 GSK-RB20 Robotic Arm

A GUI has been designed in order for user to enter either one example angle set for all six joints, or a homogenous transformation matrix for desired end-point configuration. This GUI can be seen on Figure 2.

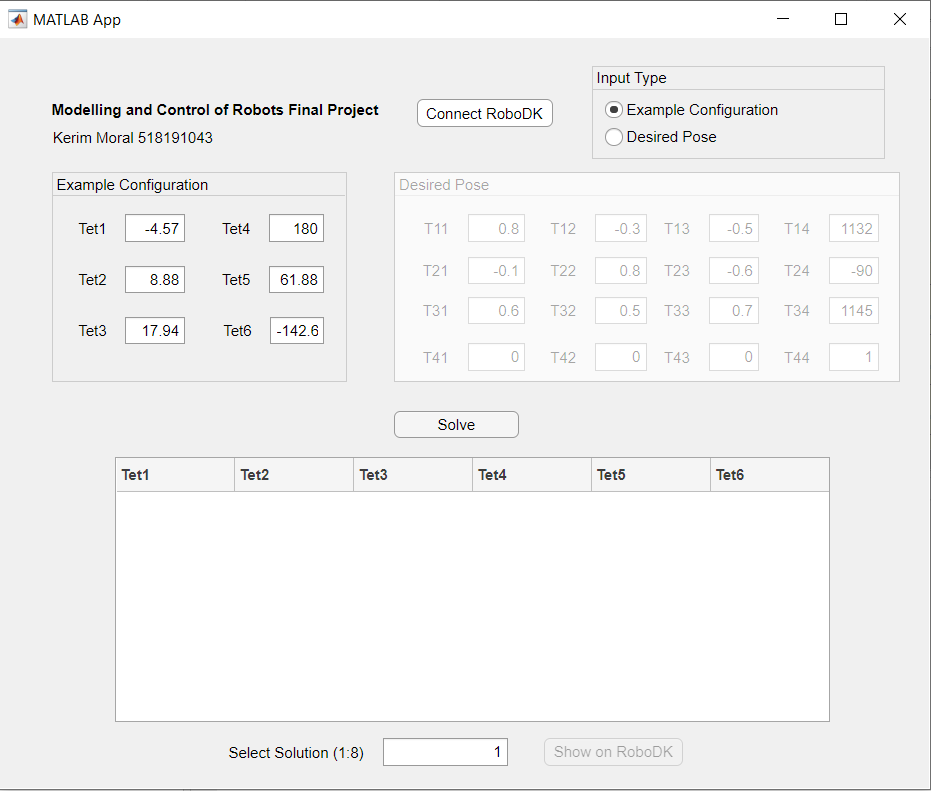


Figure 2 GUI

## Calculation of Inverse Kinematics

## Solving

Robot parameters for calculating desired joint angles can be seen on Figure 1. First, desired configuration matrix has to be found in order to solve IK.

Skrews S1, S2, S3, S4, S5 and S6 have to be calculated in order to achieve Transformation Matrix. And in order to calculate Skrews linear velocities corresponding to that skrew axes have to be calculated.

, ,

, ,

, ,

And corresponding linear velocities and Screws:

,

where

Home position M which can be seen on Figure 1 is

, ,

Skew of screws are needed to calculate Transformation Matrix. A Matlab function has been written in order to get skews:

|  |
| --- |
| function skew\_matrix = skew(w)  skew\_matrix = [0 -w(3) w(2);...  w(3) 0 -w(1);...  -w(2) w(1) 0];  end |

And finally desired Transformation matrix is:

Where to are example desired angles input from GUI.

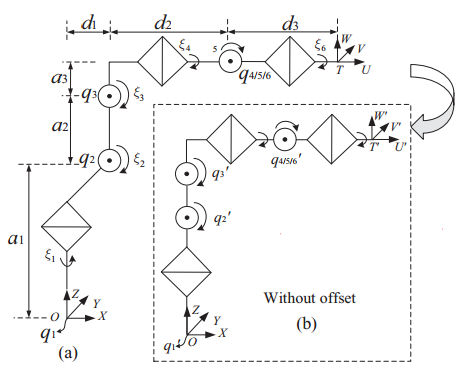


Figure Points and Distances of Platform

As can be seen from Figure 3, q4=q5=q6= is the intersection point of axes S4, S5, S6. For that reason

(1)

Let

As can be seen from Figure 3, rotational angle of is directly equal to . So

## Solving

With current configuration, due to d1 offset it is not possible to find all joints with traditional PK sub problems. Because of that, a new configuration has to be found. In order to eliminate that offset, a new translation matrix is introduced

(2)

New axis points after transformations are

And new home position

(3)

Substituting (2) to (1)

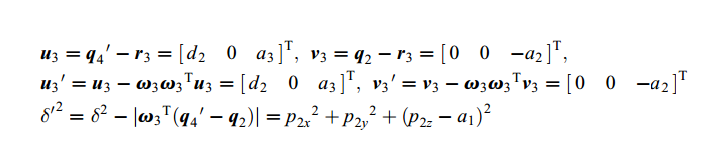
(4)

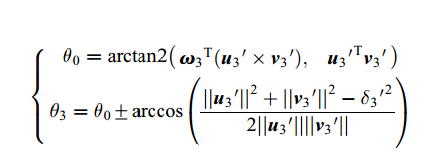
Let

If we take norm after subtracting an intersection point

Distance between and doesn’t depend on S1 and S2 axis, after taking norm

We define r3 as a point on the axis x3, where

Now can be calculated with PK subproblem 3.



## Solving

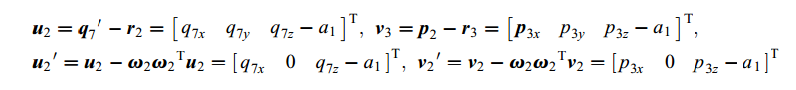
Since now known, by multiplying both sides with inverse of S1 transformation;

is also known

Where

Define p3 as

So

Now can be solved with PK subproblem 1.



## Solving and

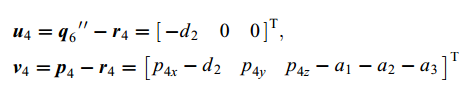
With known angles, equation (4) can be further reduced to

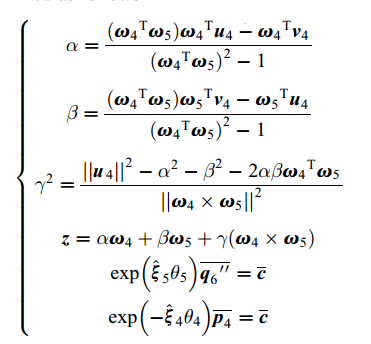
(5)

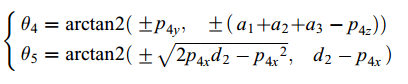
point is defined such as it is on the S6 axis but not intersecting with the S5 axis where

With multiplying (15) with

Define p4 point as

Now and can be calculated with PK subproblem 2. For that and intersection point between S5 and S4 has to be defined





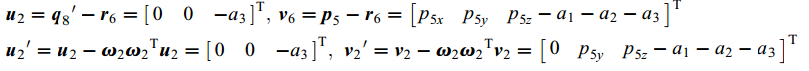
## Solving

After knowing - equation (5) can be further reduced to

Let be a point out of S6 axis where

And r6 as a point on S6,

Now can be solved with PK subproblem 1.





## GUI Design and Simulations on RoboDK

Same configuration as in the original paper has been tested in the designed GUI. After entering desired angles or final pose, GUI calculates and prints all configurations to the screen as on Figure 4. After connecting to RoboDK and hitting “Solve” button, all configurations are printed to screen. User now can select one of them and see configuration on RoboDK by hitting “Show on RoboDK” button. Except some offset caused from limits of RoboDK, simulation results are the same as paper. Two test cases can be seen on

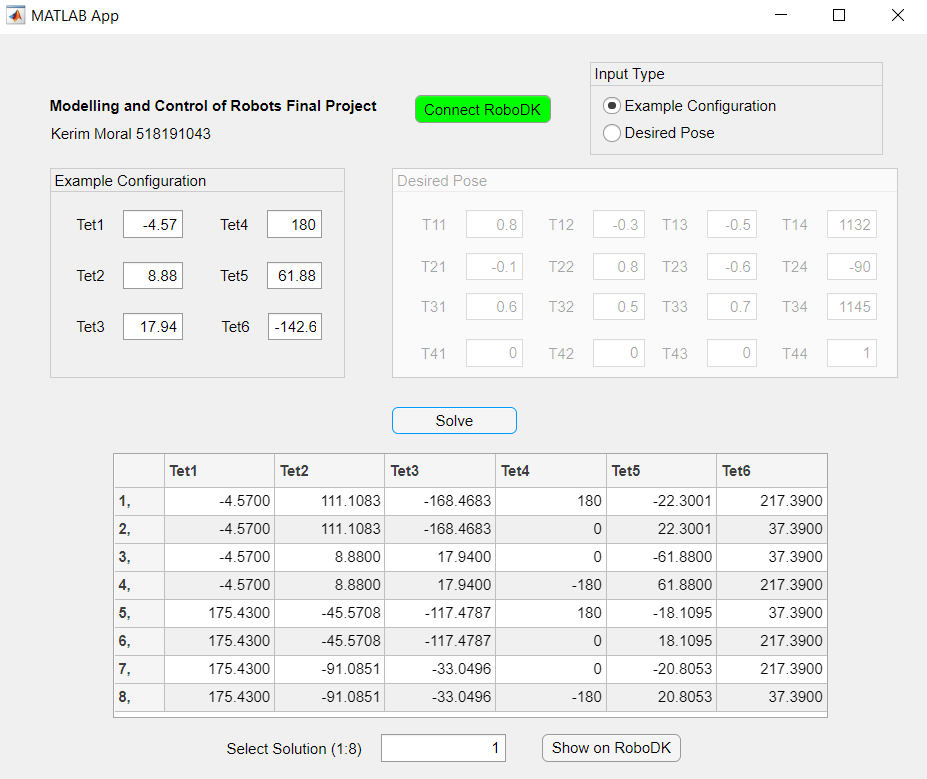
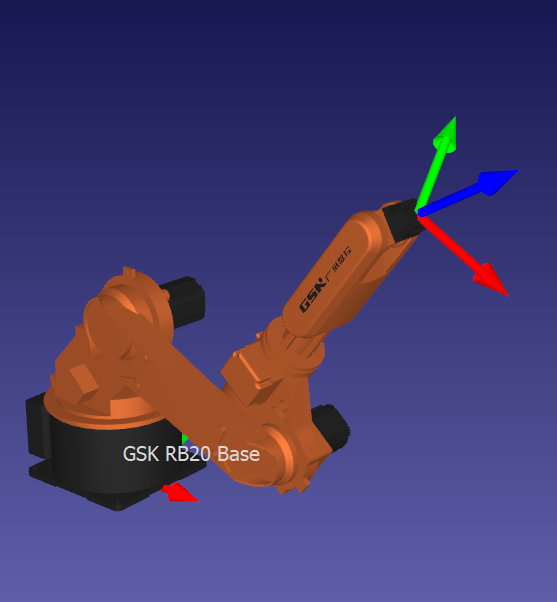
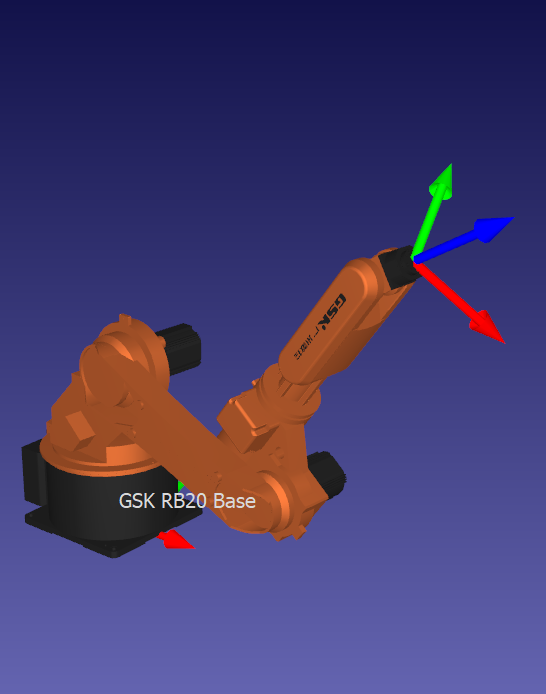
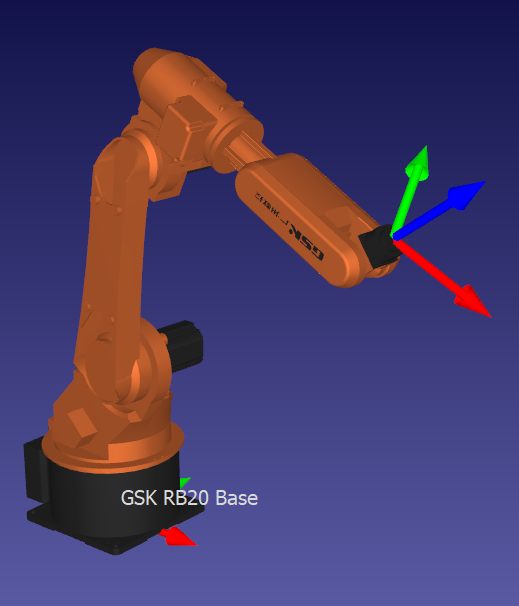
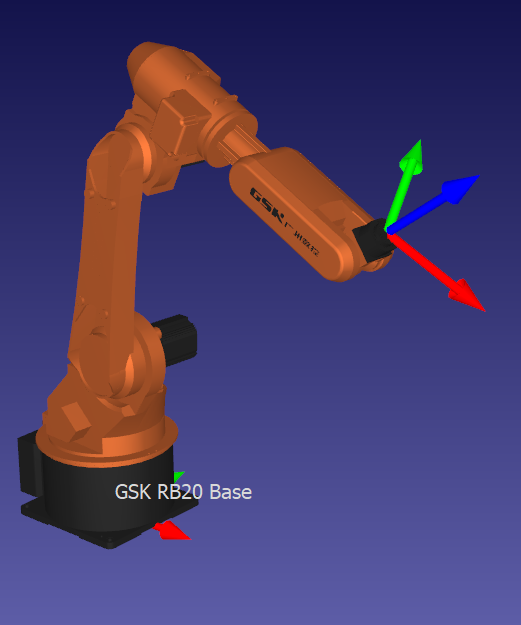
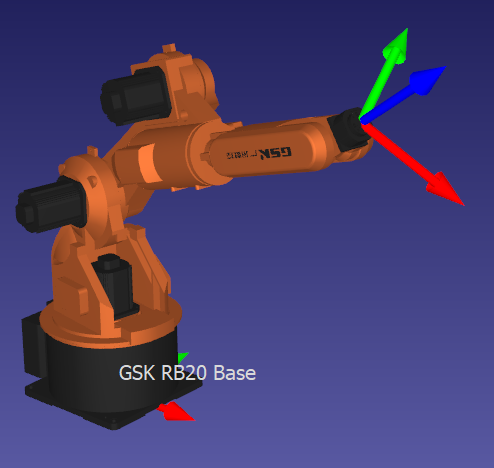
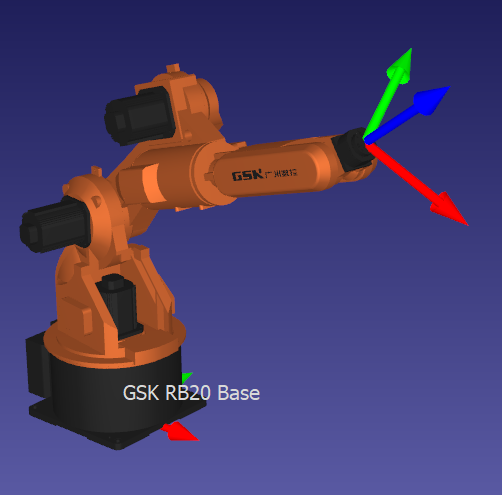


Figure Example Configurations







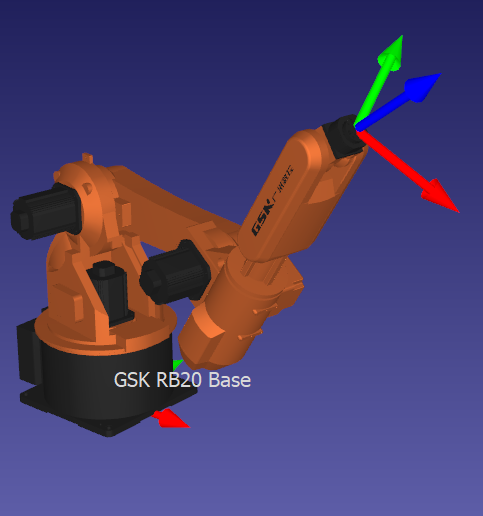
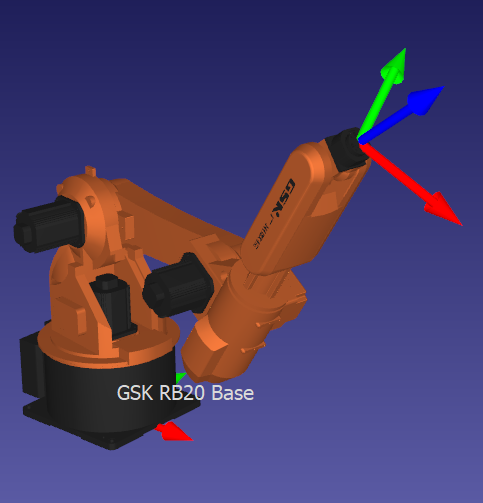


Figure Configuration Test