CHAPTER 8

ITERATIVE INVERSE KINEMATICS

8.1 STAGE V: GENERATION OF OBJECTIVE FUNCTION

8.1.1 Introduction

In order to obtain the inverse kinematics solution, an iterative technique is used. In this technique, an initial guess for the joint parameters is made. Forward analysis is performed to determine the position and orientation of the end effector for the selected joint parameters. The objective function is to minimize the difference between the position and orientation calculated from the forward analysis and goal position and orientation, which is represented as an error.

8.1.2 Objective Function

An objective function in terms of joint, D-H parameters F ($\theta 1$, θ_2, θ_3, θ_4, θ_5) is formulated, and search techniques are used to obtain the set of input parameters to minimize F. The objective function is the sum of the squares of the position error and orientation errors.

Thus the objective function could be written as

$$F(\theta_1,\theta_2,\theta_3,\theta_4,\theta_5) = (n_{xp}-n_{xg})^2+(n_{yp}-n_{yg})^2+(n_{zp}-n_{zg})^2+$$

$$(o_{xp}-o_{xg})^2+(o_{yp}-o_{yg})^2+(o_{zp}-o_{zg})^2+$$

$$(a_{xp}\text{-}a_{xg})^2+(a_{yp}\text{-}a_{yg})^2+(a_{zp}\text{-}a_{zg})^2+$$

$$(p_{xp}\text{-}p_{xg})^2+(p_{yp}\text{-}p_{yg})^2+(p_{zp}\text{-}p_{zg})^2$$

Where the subscript ‘p’ refers to a position and orientation calculated by performing a forward kinematics analysis using the current design parameters, and the subscript ‘g’ refers to the goal position value specified at the start of the inverse kinematics analysis problem. The iterative solution yields correct results when the specified end effector position and orientation is in the reachable workspace of the SCORBOT, because sets of joint angles will exist that will position and orient the end effector as desired. The objective function will thereby attain its optimal value of zero.

The optimal solution for this problem is to obtain a set of values for the design parameters that cause the objective function to equal zero.

8.1. 3 Methodology for Complete Iterative Solution

The complete iterative solution for the inverse kinematics is obtained as follows: The goal position is assumed initially. The input variables for inverse kinematics problem are twelve goal position and orientation vectors.

The goal position and orientation are given as a matrix

$$\begin{bmatrix} n_{xg} & o_{xg} & a_{xg} & p_{xg} \\ n_{yg} & o_{yg} & a_{yg} & p_{yg} \\ n_{zg} & o_{zg} & a_{zg} & p_{zg} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In the first phase, θ_1 starts from -155 and adding +1 in the next iteration keeping θ_2 = -35, θ_3=-130, θ_4=-130,and θ_5=-570 constant. It goes up to 310 iterations (Range of θ_1 Minimum -155 to Maximum +155). In phase two, 311th iteration starts with θ_1 again from -155 and adding +1 in the next iteration keeping θ_2 = -34, θ_3=-130, θ_4=-130, and θ_5=-570 constant. It goes up to 310 iterations and so on. If the objective function attains the zero value the iterative process will stop and solution for the problem will be displayed in the front panel.

8.1. 4 Methodology for Partial Iterative Solution

The partial iterative solution for the inverse kinematics obtained is explained as follows: The goal position is assumed initially. The input variables for inverse kinematics problem are twelve goal position and orientation vectors.

The goal position and orientation are given as a matrix

$$\begin{bmatrix} n_{xg} & o_{xg} & a_{xg} & p_{xg} \\ n_{yg} & o_{yg} & a_{yg} & p_{yg} \\ n_{zg} & o_{zg} & a_{zg} & p_{zg} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The following values are calculated using equations

$$\theta_1=\text{atan2}\ (p_{yg},p_{xg})$$

$$\theta_5=\text{atan2}\ (-\sin(\theta_1)*n_{xg} + \cos(\theta_1)*n_{yg},-\sin(\theta_1)*o_{xg}+\cos(\theta_1)*o_{yg})$$

$$\theta_{234}=\text{atan2}\ (-a_{zg},(\cos(\theta_1)*a_{xg} + \sin(\theta_1)*a_{yg}))$$

If we assume θ_2 and θ_3 iteratively as explained below we can calculate θ_4 using the following relation

$$\theta_4= \theta_{234}- \theta_2- \theta_3$$

In the first phase, θ_2 starts from -35 and adding +1 in the next iteration keeping θ_3=-130 constant θ_1, θ_4, θ_5 are calculated from the equations. It goes up to 165 iterations (Range of θ_2 Minimum -35 to Maximum +130). In phase two, 166th iteration starts with θ_2 again from -35 and adding +1 in the next iteration keeping θ_3=-129 constant θ_1, ,θ_4, θ_5 are calculated from the equations. It goes up to 165 iterations and so on. If the objective function attains the zero value the iterative process will stop and solution for the problem will be displayed in the front panel.

8.1.5 Total Number of Iterations and Phases (CIIKM)

The complete algorithm runs with many phases. The total number of phases expected maximum of 97812000. Total number of iterations per phase is 310. The number of generations required for solving the problem is a maximum of 3.0327x10^{10}. But depending on the goal position the total number of iterations will vary.

Phase 1:

Iteration 1: θ_1=-155, θ_2=-35, θ_3=-130, θ_4=-130, θ_5=-570

Iteration 2: θ_1=-154, θ_2=-35, θ_3=-130, θ_4=-130, θ_5=-570

Iteration 3: θ_1=-153, θ_2=-35, θ_3=-130, θ_4=-130, θ_5=-570

Iteration 4: θ_1=-152, θ_2=-35, θ_3=-130, θ_4=-130, θ_5=-570

Iteration 5: θ_1=-151, θ_2=-35, θ_3=-130, θ_4=-130, θ_5=-570

………and so on.

Iteration 310: θ_1=155, θ_2=-35, θ_3=-130, θ_4=-130, θ_5=-570

Phase 2:

Iteration 311: θ_1=-155, θ_2=-34, θ_3=-130, θ_4=-130 ,θ_5=-570

Iteration 312: θ_1=-154, θ_2=-34, θ_3=-130, θ_4=-130, θ_5=-570

Iteration 313: θ_1=-153, θ_2=-34, θ_3=-130, θ_4=-130, θ_5=-570

Iteration 314: θ_1=-152, θ_2=-34, θ_3=-130, θ_4=-130, θ_5=-570

Iteration 315: θ_1=-151, θ_2=-34, θ_3=-130, θ_4=-130 ,θ_5=-570

………and so on.

Iteration 620: θ_1=155, θ_2=-34, θ_3=-130, θ_4=-130 ,θ_5=-570

Phase 3:……..and so on.

.

.

Last Phase:

Last Iteration: θ_1=155, θ_2=130, θ_3=130, θ_4=130, θ_5=570

8.1.6 Total Number of Iterations and Phases (PIIKM)

The complete algorithm runs with many phases. The total number of phases expected maximum of 260. Total number of iterations per phase is 165. The number of generations required for solving the problem is a maximum of 42900. But depends on the goal position the total number of iterations will vary.

Phase 1:

Iteration 1: θ_2=-35, θ_3=-130, θ_1=10 ,θ_4=195,θ_5=10

Iteration 2: θ_2=-34, θ_3=-130, θ_1=10 ,θ_4=194,θ_5=10

Iteration 3: θ_2=-33, θ_3=-130, θ_1=10 ,θ_4=193,θ_5=10

Iteration 4: θ_2=-32, θ_3=-130, θ_1=10 ,θ_4=192,θ_5=10

Iteration 5: θ_2=-31, θ_3=-130, θ_1=10 ,θ_4=191,θ_5=10

………and so on.

Iteration 165: θ_2=130, θ_3=-130, θ_1=10 ,θ_4=30,θ_5=10

Phase 2:

Iteration 166: θ_2=-35, θ_3=-129, θ_1=10 ,θ_4=194,θ_5=10

Iteration 167: θ_2=-34, θ_3=-129, θ_1=10 ,θ_4=193,θ_5=10

Iteration 168: θ_2=-33, θ_3=-129, θ_1=10 ,θ_4=192,θ_5=10

Iteration 169: θ_2=-32, θ_3=-129, θ_1=10 ,θ_4=191,θ_5=10

Iteration 170: θ_2=-31, θ_3=-129, θ_1=10 ,θ_4=190,θ_5=10

………and so on.

Iteration330: θ_2=130, θ_3=-129, θ_1=10 ,θ_4=29,θ_5=10

Phase 3:……..and so on.

.

.Phase 260:

Iteration 42900: θ_2=130, θ_3=130, θ_1=10 ,θ_4=-230,θ_5=10

But for the assumed goal position (in Stage IV) the iterative process will stop at phase 140 and at 23145th iteration as the objective function FC attains zero.

8.1.7 Programming Algorithm for Complete Iteration

By this approach the algorithm is written and successfully executed in LabVIEW. The twelve input variables of the goal position (n_{xg}, n_{yg}, n_{zg}, o_{xg}, o_{yg}, o_{zg}, a_{xg}, a_{yg}, a_{zg}, p_{xg}, p_{yg} and p_{zg}) are used in the first formula node. In the second formula node θ_1,θ_2,θ_3,θ_4 and θ_5 are continuously varying and objective function is calculated dynamically for the respective assigned values of succeeded iterations. The twelve outputs of the second formula node (position and orientation vector components of the dynamic values of each iteration) are given to the first formula node as inputs. The objective function is checked for optimality. If the objective function is an optimized one, the iteration will stop and the required solution will be displayed.

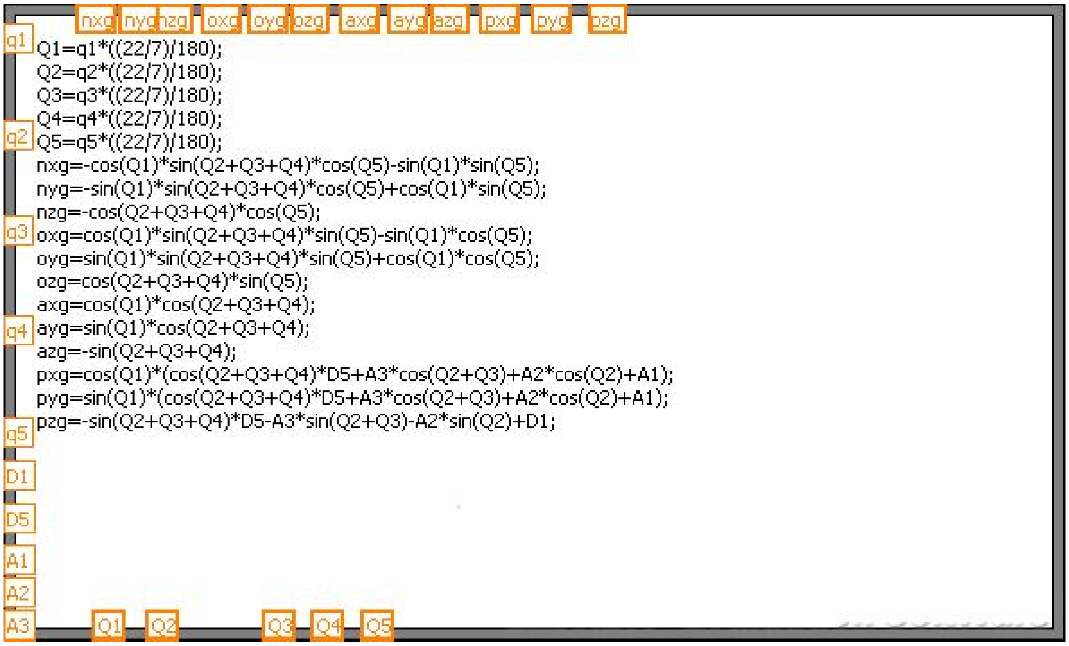
8.1.8 Programming Algorithm for Partial Iteration

By this approach the algorithm is written and successfully executed in LabVIEW. The twelve input variables of the goal position (n_{xg}, n_{yg}, n_{zg}, o_{xg}, o_{yg}, o_{zg}, a_{xg}, a_{yg}, a_{zg}, p_{xg}, p_{yg} and p_{zg}) are used to calculate θ_1, θ_{234}, θ_5 in the first formula node. In the second formula node θ_2,θ_3 are continuously varying and θ_4 is calculated dynamically for the respective assigned values of succeeded iterations (θ_4=θ_{234}- θ_2- θ_3). The 12 outputs of the second formula node (position and orientation vector components of the dynamic values of each iteration) are given to the first formula node as inputs. The objective function is checked for optimizing. If the objective function is an optimized one, the iteration will stop and the required solution will be displayed.

8.2 ITERATIVE SOLUTION

8.2.1 Complete Iteration IK Method (CIIKM)

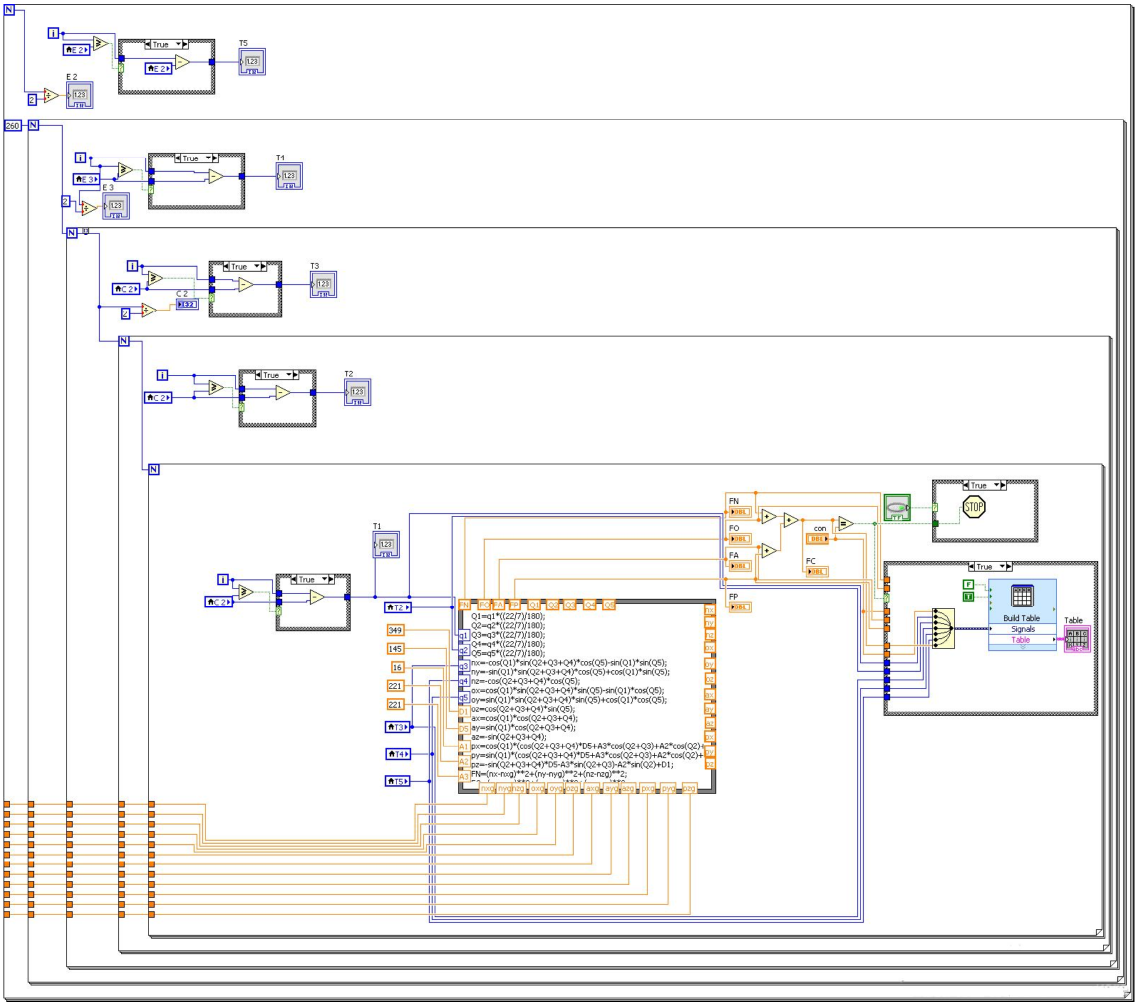
In this method, the goal position is set first before the programme is run as shown in Figure 8.1. The ranges of joints listed in Table 4.2 are as follows. T1: Base (±155), T2: Shoulder (-35to +130), T3: Elbow (±130), T4: Wrist Pitch (±130), T5: Wrist Roll (±570).



```
Q1=q1*((22/7)/180);
Q2=q2*((22/7)/180);
Q3=q3*((22/7)/180);
Q4=q4*((22/7)/180);
Q5=q5*((22/7)/180);
nxg=-cos(Q1)*sin(Q2+Q3+Q4)*cos(Q5)-sin(Q1)*sin(Q5);
nyg=-sin(Q1)*sin(Q2+Q3+Q4)*cos(Q5)+cos(Q1)*sin(Q5);
nzg=-cos(Q2+Q3+Q4)*cos(Q5);
oxg=cos(Q1)*sin(Q2+Q3+Q4)*sin(Q5)-sin(Q1)*cos(Q5);
oyg=sin(Q1)*sin(Q2+Q3+Q4)*sin(Q5)+cos(Q1)*cos(Q5);
ozg=cos(Q2+Q3+Q4)*sin(Q5);
axg=cos(Q1)*cos(Q2+Q3+Q4);
ayg=sin(Q1)*cos(Q2+Q3+Q4);
azg=-sin(Q2+Q3+Q4);
pxg=cos(Q1)*(cos(Q2+Q3+Q4)*D5+A3*cos(Q2+Q3)+A2*cos(Q2)+A1);
pyg=sin(Q1)*(cos(Q2+Q3+Q4)*D5+A3*cos(Q2+Q3)+A2*cos(Q2)+A1);
pzg=-sin(Q2+Q3+Q4)*D5-A3*sin(Q2+Q3)-A2*sin(Q2)+D1;
```

Figure 8.1 CIIKM Formula Node to get Vectors of Goal Position

The five joint parameters T1, T2, T3, T4, and T5 (θ_1, θ_2, θ_3, θ_4, and θ_5 are indicated as T1, T2, T3, T4, and T5 respectively in the LabVIEW model) are started from minimum range up to maximum range set as iterative trial. To do this five loops are used in the model as shown in Figure 8.2.



```
Q1=q1*((22/7)/180);
Q2=q2*((22/7)/180);
Q3=q3*((22/7)/180);
Q4=q4*((22/7)/180);
Q5=q5*((22/7)/180);
nx=-cos(Q1)*sin(Q2+Q3+Q4)*cos(Q5)-sin(Q1)*sin(Q5);
ny=-sin(Q1)*sin(Q2+Q3+Q4)*cos(Q5)+cos(Q1)*sin(Q5);
nz=-cos(Q2+Q3+Q4)*cos(Q5);
ox=cos(Q1)*sin(Q2+Q3+Q4)*sin(Q5)-sin(Q1)*cos(Q5);
oy=sin(Q1)*sin(Q2+Q3+Q4)*sin(Q5)+cos(Q1)*cos(Q5);
oz=cos(Q2+Q3+Q4)*sin(Q5);
ax=cos(Q1)*cos(Q2+Q3+Q4);
ay=sin(Q1)*cos(Q2+Q3+Q4);
az=-sin(Q2+Q3+Q4);
px=cos(Q1)*(cos(Q2+Q3+Q4)*D5+A3*cos(Q2+Q3)+A2*cos(Q2)+
py=sin(Q1)*(cos(Q2+Q3+Q4)*D5+A3*cos(Q2+Q3)+A2*cos(Q2)+
pz=-sin(Q2+Q3+Q4)*D5-A3*sin(Q2+Q3)-A2*sin(Q2)+D1;
FN=(nx-nxg)**2+(ny-nyg)**2+(nz-nzg)**2;
```

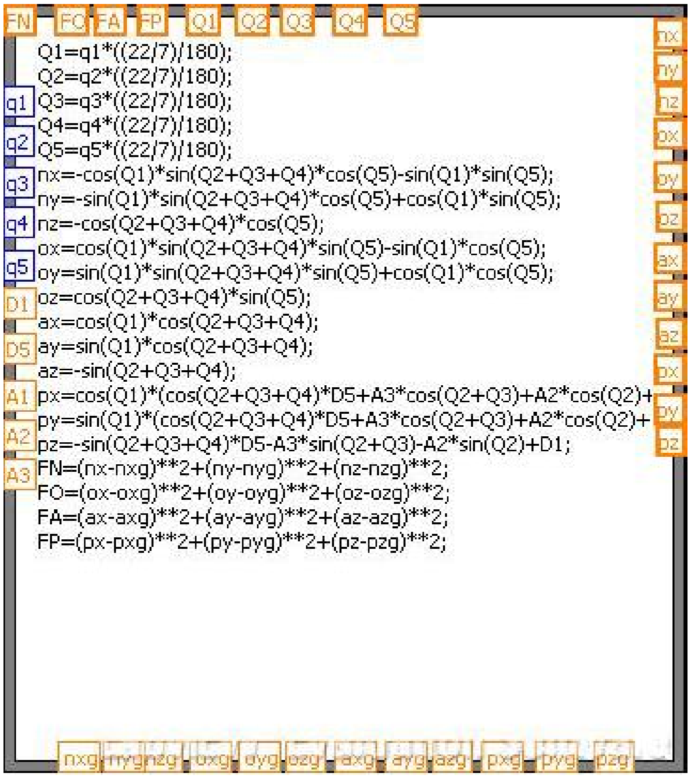
N
E 2
True
T5
E 2
260
E 3
T1
C 2
T3
T2
FN
FO
FA
FP
con
FC
STOP
Build Table
Signals
Table
T2
349
145
16
221
221
T3
T4
T5
FN FO FA FP Q1 Q2 Q3 Q4 Q5
q1 q2 q3 q4 q5 D1 D5 A1 A2 A3
nx ny nz ox oy oz ax ay az px py pz
nxg nyg nzg oxg oyg ozg axg ayg azg pxg pyg pzg

Figure 8.2 Five “FOR” Loops for All the Joint Parameters

From the five loops the iterative parameter values are given as input for the next formula node as shown in Figure 8.3. All the components are substituted in the objective function for obtaining inverse kinematics solution (Figure 8.4).

$F(\theta_1,\theta_2,\theta_3,\theta_4,\theta_5) = \quad (n_{xp}-n_{xg})^2+(n_{yp}-n_{yg})^2+(n_{zp}-n_{zg})^2+(o_{xp}-o_{xg})^2+(o_{yp}-o_{yg})^2+(o_{zp}-o_{zg})^2+(a_{xp}-a_{xg})^2+(a_{yp}-a_{yg})^2+(a_{zp}-a_{zg})^2+(p_{xp}-p_{xg})^2+(p_{yp}-p_{yg})^2+(p_{zp}-p_{zg})^2)$

A case structure is used to stop the programme, when the objective function attained minimum value as shown in Figure 8.5.



```
Q1=q1*((22/7)/180);
Q2=q2*((22/7)/180);
Q3=q3*((22/7)/180);
Q4=q4*((22/7)/180);
Q5=q5*((22/7)/180);
nx=-cos(Q1)*sin(Q2+Q3+Q4)*cos(Q5)-sin(Q1)*sin(Q5);
ny=-sin(Q1)*sin(Q2+Q3+Q4)*cos(Q5)+cos(Q1)*sin(Q5);
nz=-cos(Q2+Q3+Q4)*cos(Q5);
ox=cos(Q1)*sin(Q2+Q3+Q4)*sin(Q5)-sin(Q1)*cos(Q5);
oy=sin(Q1)*sin(Q2+Q3+Q4)*sin(Q5)+cos(Q1)*cos(Q5);
oz=cos(Q2+Q3+Q4)*sin(Q5);
ax=cos(Q1)*cos(Q2+Q3+Q4);
ay=sin(Q1)*cos(Q2+Q3+Q4);
az=-sin(Q2+Q3+Q4);
px=cos(Q1)*(cos(Q2+Q3+Q4)*D5+A3*cos(Q2+Q3)+A2*cos(Q2)+
py=sin(Q1)*(cos(Q2+Q3+Q4)*D5+A3*cos(Q2+Q3)+A2*cos(Q2)+
pz=-sin(Q2+Q3+Q4)*D5-A3*sin(Q2+Q3)-A2*sin(Q2)+D1;
FN=(nx-nxg)**2+(ny-nyg)**2+(nz-nzg)**2;
FO=(ox-oxg)**2+(oy-oyg)**2+(oz-ozg)**2;
FA=(ax-axg)**2+(ay-ayg)**2+(az-azg)**2;
FP=(px-pxg)**2+(py-pyg)**2+(pz-pzg)**2;
```

Figure 8.3 CIIKM Formula node for Iterative Joint Parameters

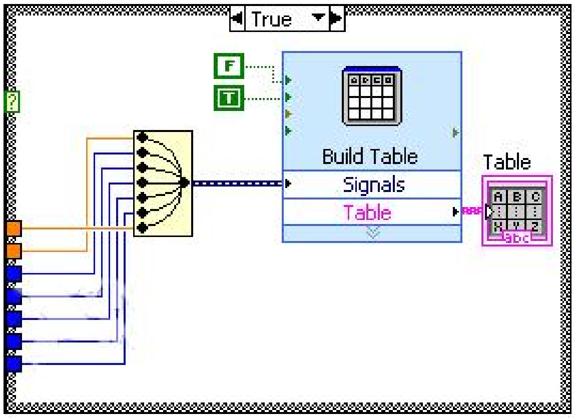
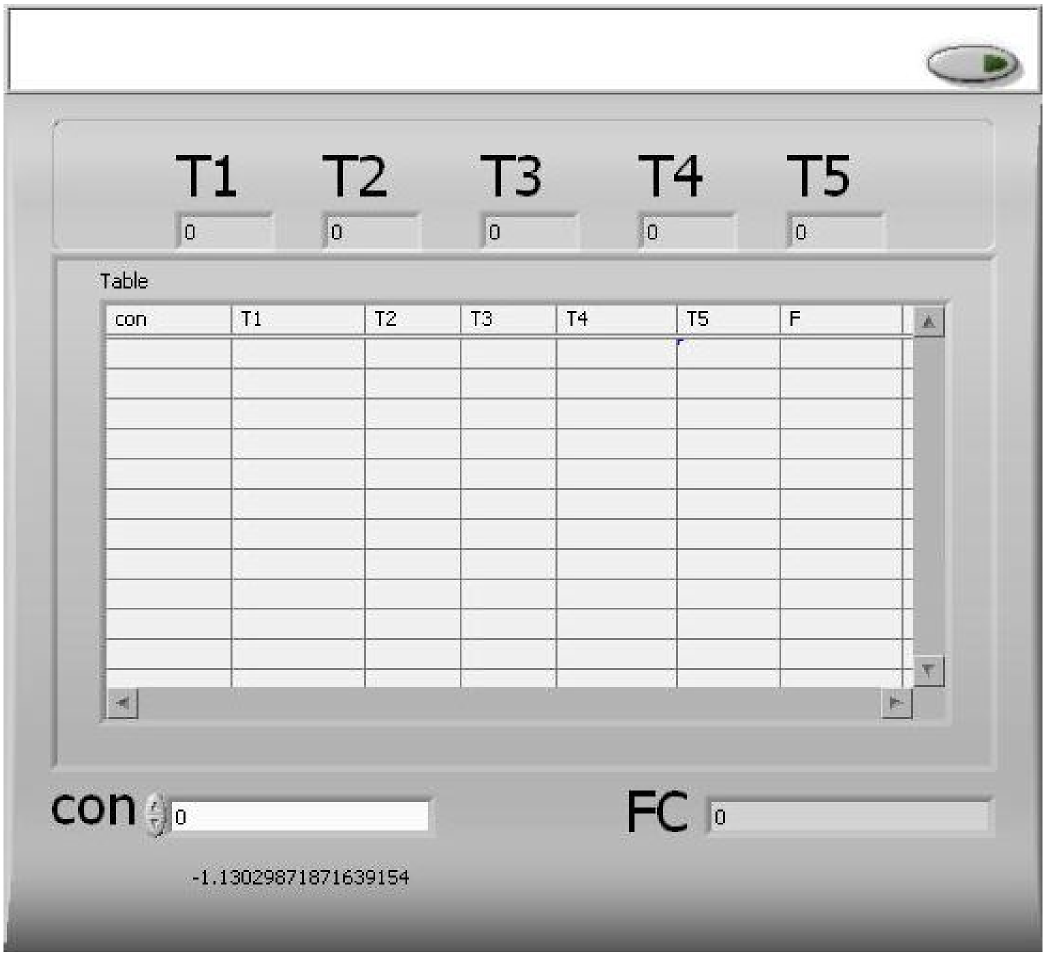


Figure 8.4 Case Structure to Display the Optimised Output



T1 0

T2 0

T3 0

T4 0

T5 0

Table

con	T1	T2	T3	T4	T5	F

con 0

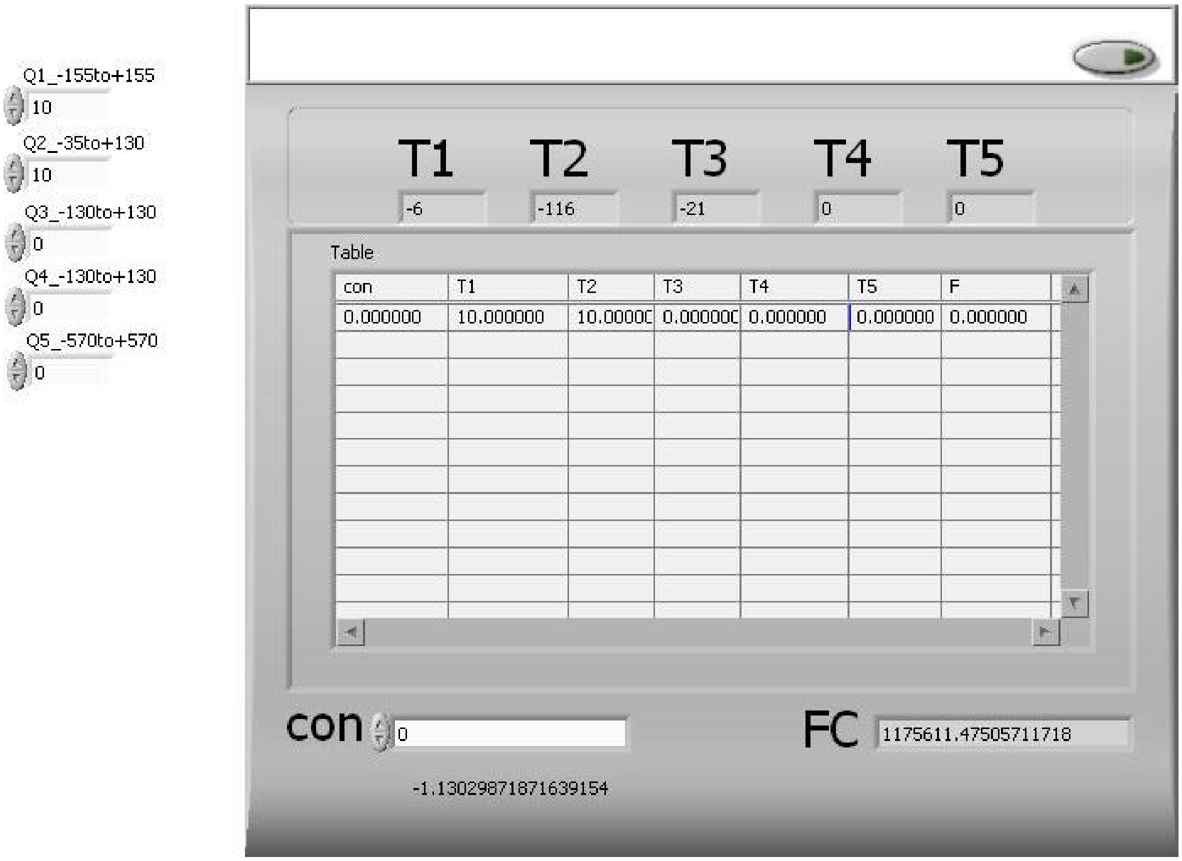
FC 0

-1.13029871871639154

Figure 8.5 Front Panel of CIIKM (Before Execution)

All the components are substituted in the objective function. Now it will be ready for inverse kinematics solution (Refer Figure 8.5).

When the programme is executed, the combined objective function becomes zero at particular set of values, and the values will be indicated in the front panel as shown in the Figure 8.6.



Q1_-155to+155 10

Q2_-35to+130 10

Q3_-130to+130 0

Q4_-130to+130 0

Q5_-570to+570 0

T1	T2	T3	T4	T5
-6	-116	-21	0	0

Table

con	T1	T2	T3	T4	T5	F
0.000000	10.000000	10.00000C	0.00000C	0.000000	0.000000	0.000000

con 0

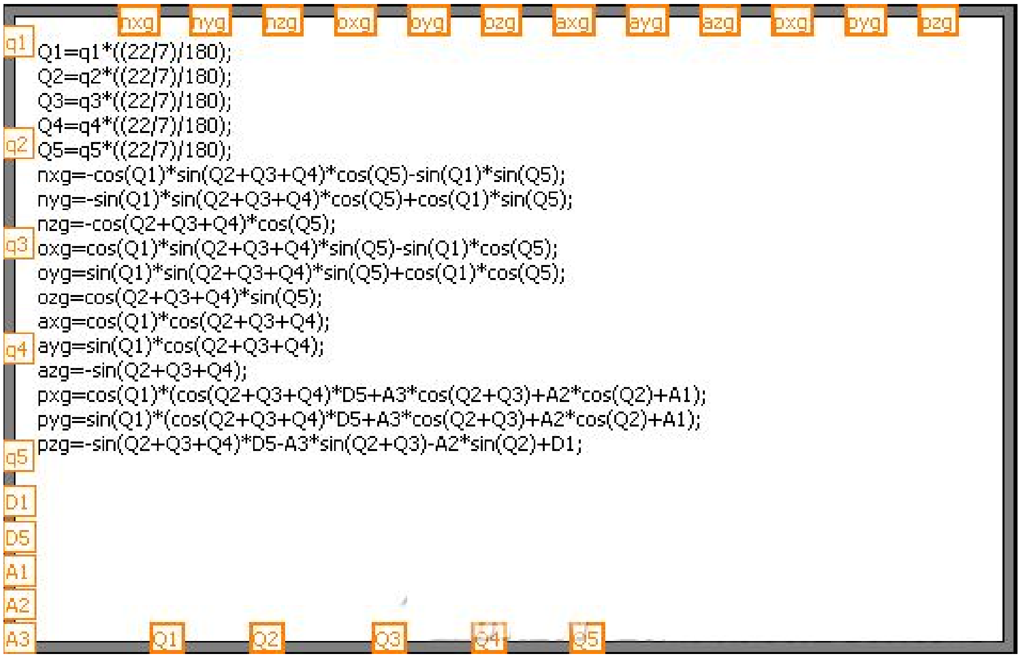
FC 1175611.47505711718

-1.13029871871639154

8.6 CIIKM Front Panel (After Execution)

8.2.2 Partial Iteration IK Method (PIIKM)

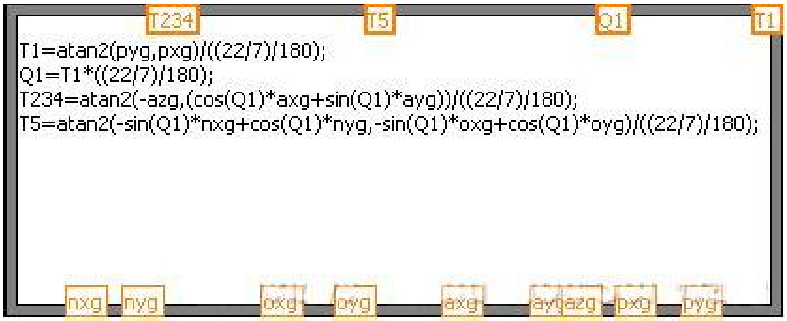
The ranges of joints listed in Table 4.2 are as follows. T1: Base (±155), T2: Shoulder (-35to +130), T3: Elbow (±130), T4: Wrist Pitch (±130), T5: Wrist Roll (±570). (θ_1, θ_2, θ_3, θ_4, and θ_5 are indicated as T1, T2, T3, T4, and T5 respectively in the LabVIEW model). In this method, the goal position is set first before the programme is run as shown in Figure 8.7.



```
Q1=q1*((22/7)/180);
Q2=q2*((22/7)/180);
Q3=q3*((22/7)/180);
Q4=q4*((22/7)/180);
Q5=q5*((22/7)/180);
nxg=-cos(Q1)*sin(Q2+Q3+Q4)*cos(Q5)-sin(Q1)*sin(Q5);
nyg=-sin(Q1)*sin(Q2+Q3+Q4)*cos(Q5)+cos(Q1)*sin(Q5);
nzg=-cos(Q2+Q3+Q4)*cos(Q5);
oxg=cos(Q1)*sin(Q2+Q3+Q4)*sin(Q5)-sin(Q1)*cos(Q5);
oyg=sin(Q1)*sin(Q2+Q3+Q4)*sin(Q5)+cos(Q1)*cos(Q5);
ozg=cos(Q2+Q3+Q4)*sin(Q5);
axg=cos(Q1)*cos(Q2+Q3+Q4);
ayg=sin(Q1)*cos(Q2+Q3+Q4);
azg=-sin(Q2+Q3+Q4);
pxg=cos(Q1)*(cos(Q2+Q3+Q4)*D5+A3*cos(Q2+Q3)+A2*cos(Q2)+A1);
pyg=sin(Q1)*(cos(Q2+Q3+Q4)*D5+A3*cos(Q2+Q3)+A2*cos(Q2)+A1);
pzg=-sin(Q2+Q3+Q4)*D5-A3*sin(Q2+Q3)-A2*sin(Q2)+D1;
```

Figure 8.7 PIIKM Formula Node to get Vectors of Goal Position

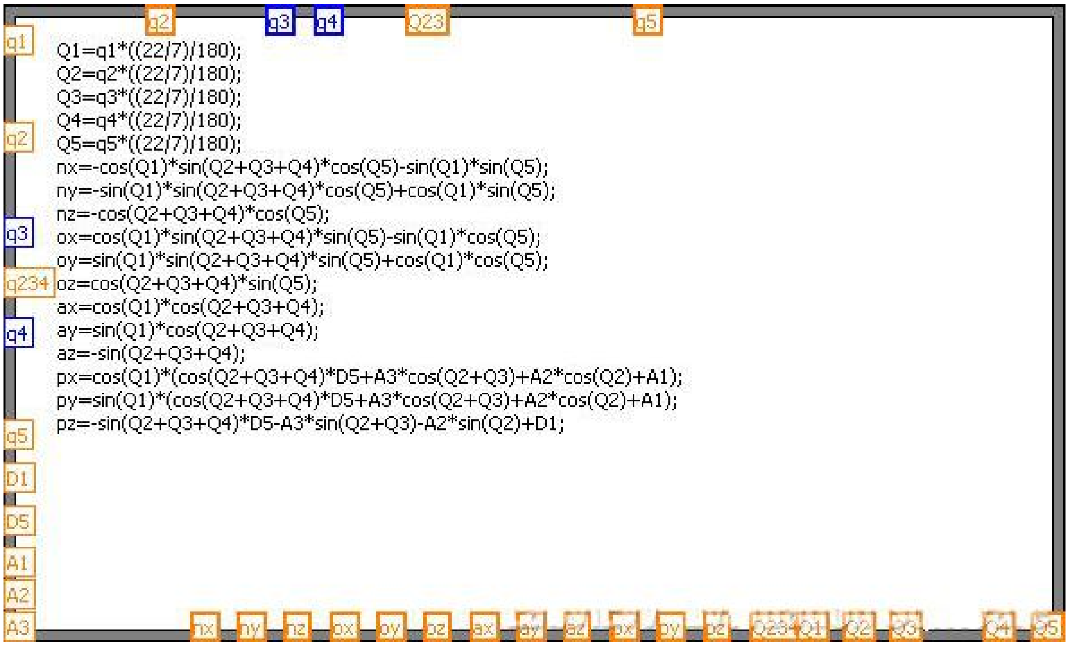
The three formulae used to find the joint parameters are as follows: T1, T5, T234 (θ_1, θ_5 and θ_{234} are indicated as T1, T5 and T234 respectively in the LabVIEW model). It is shown in Figure 8.8.



```
T1=atan2(pyg,pxg)/((22/7)/180);
Q1=T1*((22/7)/180);
T234=atan2(-azg,(cos(Q1)*axg+sin(Q1)*ayg))/((22/7)/180);
T5=atan2(-sin(Q1)*nxg+cos(Q1)*nyg,-sin(Q1)*oxg+cos(Q1)*oyg)/((22/7)/180);
```

Figure 8.8 Formula Node to find the Three Joint Parameters using IK Formulae

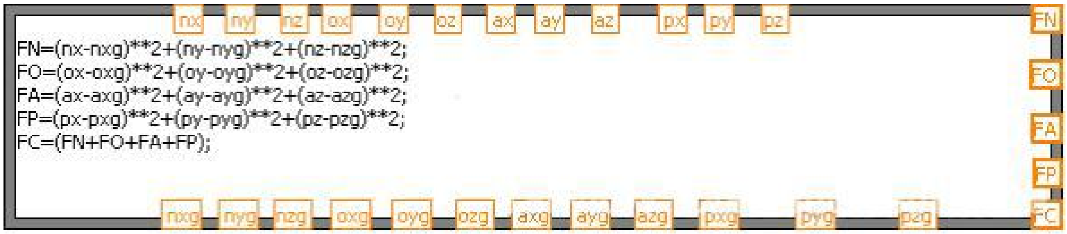
The other two parameters T3 (θ_3) and T4 (θ_4) are started from minimum range -130 up to maximum range +130 as iterative trial. The values are given as input for the next formula node shown in Figure 8.9.



```
Q1=q1*((22/7)/180);
Q2=q2*((22/7)/180);
Q3=q3*((22/7)/180);
Q4=q4*((22/7)/180);
Q5=q5*((22/7)/180);
nx=-cos(Q1)*sin(Q2+Q3+Q4)*cos(Q5)-sin(Q1)*sin(Q5);
ny=-sin(Q1)*sin(Q2+Q3+Q4)*cos(Q5)+cos(Q1)*sin(Q5);
nz=-cos(Q2+Q3+Q4)*cos(Q5);
ox=cos(Q1)*sin(Q2+Q3+Q4)*sin(Q5)-sin(Q1)*cos(Q5);
oy=sin(Q1)*sin(Q2+Q3+Q4)*sin(Q5)+cos(Q1)*cos(Q5);
oz=cos(Q2+Q3+Q4)*sin(Q5);
ax=cos(Q1)*cos(Q2+Q3+Q4);
ay=sin(Q1)*cos(Q2+Q3+Q4);
az=-sin(Q2+Q3+Q4);
px=cos(Q1)*(cos(Q2+Q3+Q4)*D5+A3*cos(Q2+Q3)+A2*cos(Q2)+A1);
py=sin(Q1)*(cos(Q2+Q3+Q4)*D5+A3*cos(Q2+Q3)+A2*cos(Q2)+A1);
pz=-sin(Q2+Q3+Q4)*D5-A3*sin(Q2+Q3)-A2*sin(Q2)+D1;
```

Figure 8.9 PIIKM Formula node for Iterative Joint Parameters

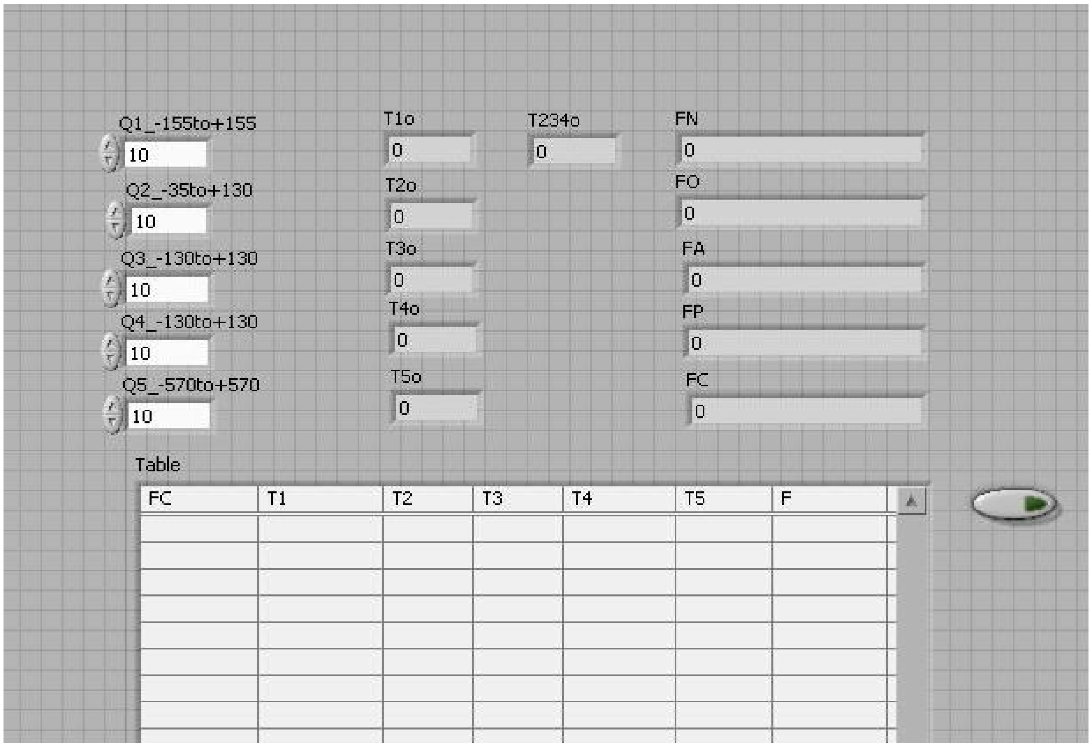
The objective function is shown in the next node as shown in Figure 8.10. Totally four formula node are generated.



```
FN=(nx-nxg)**2+(ny-nyg)**2+(nz-nzg)**2;
FO=(ox-oxg)**2+(oy-oyg)**2+(oz-ozg)**2;
FA=(ax-axg)**2+(ay-ayg)**2+(az-azg)**2;
FP=(px-pxg)**2+(py-pyg)**2+(pz-pzg)**2;
FC=(FN+FO+FA+FP);
```

Figure 8.10 Formula Node for Objective Functions

In this method, the goal position is set first before the programme is run as shown in Figure 8.11. When the programme is executed, the combined objective function becomes zero at particular set of values, and the values will be indicated in the front panel as shown in the Figure 8.12.



Q1_-155to+155
10

Q2_-35to+130
10

Q3_-130to+130
10

Q4_-130to+130
10

Q5_-570to+570
10

T1o
0

T2o
0

T3o
0

T4o
0

T5o
0

T234o
0

FN
0

FO
0

FA
0

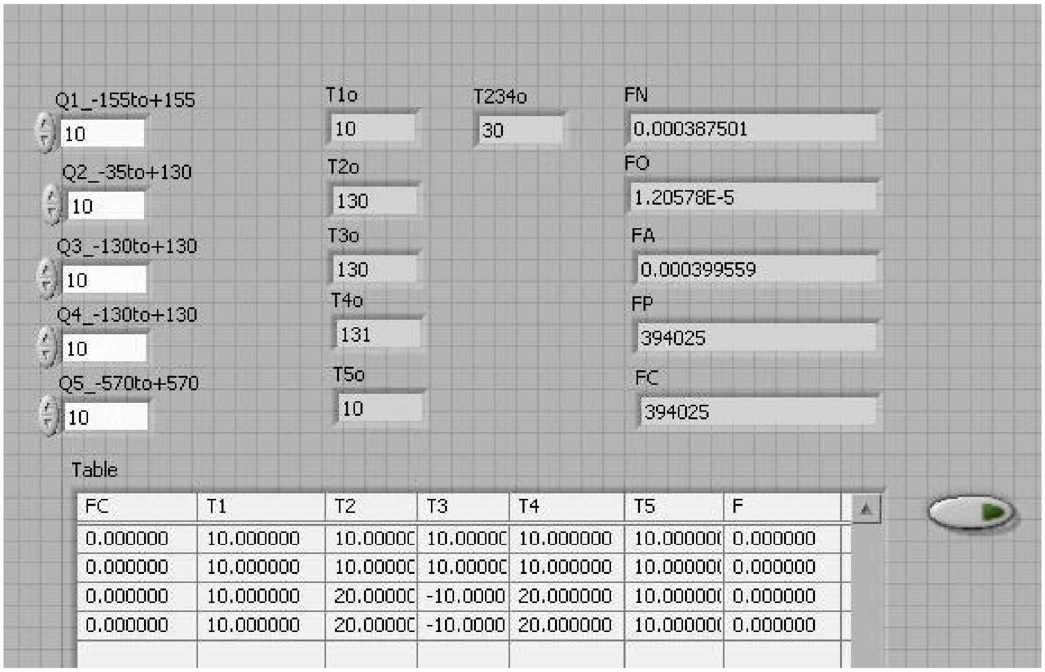
FP
0

FC
0

Table

FC	T1	T2	T3	T4	T5	F

Figure8.11 Front Panel PIIKM (Before Execution)



Q1_-155to+155 10

Q2_-35to+130 10

Q3_-130to+130 10

Q4_-130to+130 10

Q5_-570to+570 10

T1o 10

T2o 130

T3o 130

T4o 131

T5o 10

T234o 30

FN 0.000387501

FO 1.20578E-5

FA 0.000399559

FP 394025

FC 394025

Table

FC	T1	T2	T3	T4	T5	F
0.000000	10.000000	10.00000	10.00000	10.000000	10.00000	0.000000
0.000000	10.000000	10.00000	10.00000	10.000000	10.00000	0.000000
0.000000	10.000000	20.00000	-10.0000	20.000000	10.00000	0.000000
0.000000	10.000000	20.00000	-10.0000	20.000000	10.00000	0.000000

Figure8.12 Front Panel PIIKM (After Execution)

Both CIIKM and PIIKM iterations have been successfully developed using LabVIEW.