CINEMÁTICA INVERSA DE UN BRAZO ROBÓTICO DE 6 GRADOS DE LIBERTAD

El problema de la cinemática inversa consiste de determinar el valor de las variables de articulación dada la configuración del efector final. Al comienzo de la simulación se llama el procedimiento `initInverseKinematics2` del objeto robot, esto inicializa algunos parámetros para poder realizar la cinemática inversa, los más importantes son el valor de la transformación homogénea T_6 del efector final misma que se calcula en las líneas 154-155, así también la posición del punto terminal P_T respecto del sistema de referencia global (línea 163)y con ello la posición del origen local O_6 del efector final (líneas 166-168). Para el caso en el que se tiene una muñeca esférica se cumple O_4=O_6 (línea 172).

src\Robot.cpp

```
149  void Robot::initInverseKinematics2(){
150  t=0;
151  ///calcular la posicion de O6,k6,i6 en base global y respecto del sistema de
152  TH06.resetIdentity();
153  modelo3D *model;
154  for (int m=0;m<modelos.size();m++){
155
156      model=modelos[m];
157      TH06=TH06* THList[2*m+0]*THList[2*m+1];}
158
159  k60=TH06*kL; /// siempre se conoce del control manual
160  i60=TH06*iL;
161  i6=i60;
162  k6=k60;
163  TPG=TH06*TPL;
164  TPG0=TPG; // posición de punto de interes del efector final en base global v
165  TPVG=TH06*TPVL; //vector de punto terminal en vase blobal
166  O6.entry(0,0)=TH06.entry(0,3);
167  O6.entry(1,0)=TH06.entry(1,3);
168  O6.entry(2,0)=TH06.entry(2,3);
169  O6.entry(3,0)=1;
170  ///calcular O4 a partir de O6 y k6,i6 (se tiene una muñeca esférica)
171  O40=O6-z6*k6; ///debe ser correcto
172  O4=O40;
173  //agregar PT a la curva
174  curva.resize(0);
175  curva.push_back(vector3d(TPG.entry(0,0),TPG.entry(1,0),TPG.entry(2,0)));
176
177  }
```

La cinemática inversa comienza cuando el usuario presiona la tecla 'P', con ello se llama al procedimiento `InverseKinematics2` del objeto `robot` y en caso de arrojar soluciones correctas para las variables de articulación en la variable boolena `status` se almacena el valor `true`, en este caso el tiempo de simulación se actualiza (línea 249), así como la nueva posición del punto terminal a través del procedimiento `Parametrica` del objeto `robot` (línea 250) y con ello la nueva posición del origen O_6 del efector final (línea 251).

main.cpp

```
244                 break;
245
246             case  'P':
247             status= Miclase->robot.InverseKinematics2();
248             if (status==true){
249             Miclase->robot.t=Miclase->robot.t+0.2;  //actualiza el tiemp
250             Miclase->robot.Parametrica(); //nueva posición de punto term
251             Miclase->robot.O6=Miclase->robot.TPG-Miclase->robot.TPVG; //
252             }
253             break;
254            case  'E':
255             Miclase->robot.Estado();  //muestra algunos parámetros de in
256
257             break;
258
```

El procedimiento `InverseKinematics2` del objeto robot primeramente, dado el valor de O_4 coincidente con O_6, resuelve para las primeras variables de articulación (línea 236), esto es posible debido a que se tiene una muñeca esférica, en caso de que se encuentren soluciones consistentes el valor de la variable `status` será `true`, en caso contrario la simulación termina.

src\Robot.cpp

```
229 bool Robot::InverseKinematics2(){
230
231 O4=O6; ///muñeca esférica
232 xg=O4.entry(0,0);
233 yg=O4.entry(1,0);
234 zg=O4.entry(2,0);
235 // encontrar el valor de las 3 últimas variables de articulación
236 bool status=InverseKinematics();  //cinemática inversa para calcular el valor de las primeras tres variables
237 if (status==false){ cout<<" configuracion no alcanzable, utilizar el control manual"<<endl ;return false ;}
238
239
240 TH.resetIdentity();
241 modelo3D *model;
242 for (int m=0;m<4;m++){
243
244     model=modelos[m];
245     TH=TH* THList[2*m+0]*THList[2*m+1];
246
247   }
248 ///TH  es  0 a 3, 0  es el indice de la base
```

Una vez calculadas las variables de articulación de los primeros tres grados de libertad se procede a resolver los valores de las variables de articulación de los últimos tres grados de libertad, esto se realiza en las líneas 242-268.

src\Robot.cpp

```
241     modelo3D *model;
242     for (int m=0;m<4;m++){
243
244         model=modelos[m];
245         TH=TH* THList[2*m+0]*THList[2*m+1];
246
247     }
248     ///TH  es  0 a 3, 0  es el indice de la base
249     A=TH.inversa()*TH06;
250
251     double t41,t42,t43, t51,t52,t53,t61,t62,t63;
252     t41=atan2(-A.entry(1,2),-A.entry(0,2));
253     t42=atan2(-A.entry(1,2),-A.entry(0,2))+PI;
254     t43=atan2(-A.entry(1,2),-A.entry(0,2))-PI;
255
256
257     theta4 =menor3(t41,t42,t43,theta4);
258     float q4=theta4;
259     t61=atan2(-A.entry(2,1),A.entry(2,0));
260     t62=atan2(-A.entry(2,1),A.entry(2,0))+PI;
261     t63=atan2(-A.entry(2,1),A.entry(2,0))-PI;
262
263     theta6 =menor3(t61,t62,t63,theta6);
264
265     t51=asin(A.entry(2,2));
266     t52=PI-asin(A.entry(2,2));;
267     t53=-PI-asin(A.entry(2,2));;
268     theta5 =menor3(t51+PI/2.0,t52+PI/2.0,t53+PI/2.0,theta5);
269
270     /// se acutalizan las últimas 3 transformaciones Tz
```

Finalmente se actualizan las transformaciones T_z de los últimos tres grados de libertad (líneas272-283), se agrega la posición del punto terminal a la curva (línea 286) y se actualiza la posición de O_6.

src\Robot.cpp

```
268  theta5 =menor3(t51+PI/2.0,t52+PI/2.0,t53+PI/2.0,theta5);
269
270  /// se acutalizan las últimas 3 transformaciones Tz
271
272  AplicarTHz(theta4,{0,0,z4});
273  THList[8]=THz;
274  TH=TH*THz*THList[9];
275
276  AplicarTHz(theta5,{0,0,z5});
277  THList[10]=THz;
278  TH=TH*THz*THList[11];
279
280
281  AplicarTHz(theta6,{0,0,z6});
282  THList[12]=THz;
283  TH=TH*THz*THList[13];
284
285  //agregar posición de punto terminal a la curva
286  curva.push_back(vector3d(TPG.entry(0,0),TPG.entry(1,0),TPG.entry(2,0)));
287  O6=TPG-TPVG; //TPG se actualiza en paramétrica de O6
288  /*
289  Estado();
290  cout<<"La posición de O6 calculado con TPG-TPVG "<<endl;
291  O6.mostrar();
292  */
293  Estado();
294  return true;
295  }
```

De esta manera mientras se presiona la tecla ‘P’ el punto terminal se moverá por la ecuación paramétrica defina en el procedimiento `Parametrica` y al mismo tiempo los eslabones del brazo manipulador se configurarán consistentemente. El resultado se muestra a continuación.