Cyclic Calculation of Inverse Kinematics for n-link Serial Mechanism

František Trebuňa, Zdenko Bobovsky*

Department of Applied Mechanics and Mechatronics, Faculty of Mechanical Engineering, Letná 9, 04200 Košice, Slovak Republic

BIOGRAPHICAL NOTES
František Trebuňa, Dr.h.c. mult. Prof. Ing. CSc. is a professor of applied mechanics, Dean of Faculty of Mechanical Engineering of Technical university of košice, Head of the Department of Applied Mechanics and Mechatronics. He is author of 9 monographs, 9 university textbook, special book publications, 12 university notebooks and more than 300 publications in journals and conference proceedings at Slovakia abroad. He is author of important projects and engineering works. He received several prizes at home and abroad. He received three honorary Doctor Honoris Causa (Dr.h.c.) including two from foreign universities for the development of applied mechanics and mechatronics.

Zdenko Bobovský, Ing. PhD. is an assistant of the Department of Applied Mechanics and Mechatronics at Faculty of Mechanical Engineering at Technical University of Košice. He received B.Sc. degree in production systems with industrial robots and manipulators (2002), M.Sc degree in Construction of robotics system (2005), M.Sc degree in automotive production (2007), Medal of Dean for excellent student (2007) and Ph.D in Production system at the same faculty (2009). His current fields of interest are mechatronics, metamorphous self-reconfigurable modular robotic systems and applied mechanics.

KEY WORDS
Inverse Kinematics, Cyclic Coordinate Descent, Serial Chain Robot

ABSTRACT
Inverse kinematics of mechanic system is area of solution at many universities and scientific research institutions. This paper dealt with of solution problem of inverse kinematic by a cyclic coordinate descent method (CCD). This method allows use n-link chain mechanism without complicated changing of control algorithm. The article is focuses on the issue of setting the number of elements and also presents a modified method the incremental CCD, which is compared with this method.

1. Inverse kinematic problem
Inverse kinematic problem can be described as „finding“ the right parameters for each kinematic joint of mechanism to achieve the desired result in a pre-defined position of member. The available literature provides several definitions for this problem. In defined shape of mechanism, Fig. 1, we know where its base is. With the control system and sensors know the current mechanism shape and position of the end point of the mechanism. If we define a new position in which to get the mechanism end
point, for the computation algorithm, the actual end point position is starting position and its task is to calculate values for individual joints $q_1$ to $q_n$, so the end point has the desired position with reasoning error limit [1].

2. Cyclic computation

This method allows computing the inverse kinematics of redundant mechanisms with principle of gradual calculation parameters for each kinematics joint separately.

2.1 Cyclic coordinate descent

In this method there is gradual calculation for each kinematic joint separately from the last joint in a series of mechanism to the first. This cycle is repeated until it reaches desired position of the end point of mechanism respectively until value of error isn't smaller then error limit. The $q$ value is a calculated difference between actual angle and desired angle. This $q$ value can be limited by the maximum value of step size. The algorithm of calculation in basic form for one “finding” point with defined max step size and without optimization algorithm has form:

If (error > errorLimit)
cycle = 0
For $i = n : 1$

Find vector between the i-joint and end point
Find vector between the i-joint and desired point
Identify different between these vectors
If (difference > maxStep)
difference = maxStep
end
Change angle of i-joint for difference value
Recalculate position of all joint from i to n
cycle = cycle +1;
Find error between end point and desired point
If (error < errorLimit)
End cycle For
Else
$i = i - 1$
if $i == 0$
$i = n$
end
end
end
end

2.2 Incremental cyclic coordinate descent

This method allows with last link “follow” desired point. Procedure of computation is in the next algorithm. The algorithm is for one “finding” point with defined a max step size and without an optimized algorithms. Difference between CCD and iCCD is marked:

While (error > errorLimit)
g = 0
cycle = 0
For $i = n : 1$

Find vector between the i-joint and end point
Find vector between the i-joint and desired point
Identify different between these vectors
If (difference > maxStep)
difference = maxStep
end
Change angle of i-joint for difference value
Recalculate position of all joint from i to n
cycle = cycle +1;
Find error between end point and desired point
If (error < errorLimit)
End of cycle For
Else
$i = i - 1$
If $i == g$ or $i == 0$
$l = n$
$g = g - 1$

Fig. 1: New rope without primary and subsequent lubrication.

There are several computation methods that are described in available literature. Whatever the method using Jacobian and their subsequent modifications [1,3,4], or other methods [2]. The biggest problem with these methods arises with the redundant manipulators, when the desired position is possible to achieve by the infinity combinations of $q_i$ values.
If \( g == -1 \)
\[
g = n - 2
\]
end
end
end
end

3. Mathematical model

Between a main criteria for choose the right inverse kinematics algorithm belongs a number of calculation cycles to an achieve desired task and a time of computation. Computation times are changed depending on the type of computer on which the calculation is performed. Therefore, all calculations for this post were taken on a single computer. This part of article presents obtained results for a chain link mechanism composted from of \( n = 10 \) to 100 links with step 10 for changing maximum step size of i-joint angle - 1°, 10° a 100°.

Starting position of end point is \( x_p = n \) a \( y_p = 0 \) and desired position of the mechanism is defined as follows:
1. point: \( x_k = n/2 \) a \( y_k = n/2 \),
2. point: \( x_k = 0 \) a \( y_k = n/2 \),
3. point: \( x_k = -n/2 \) a \( y_k = n/2 \).

Link \( n \) has unit length and error limit is set to 0,01. The computation was made in application programmed in MATLAB®.

3.1 Cyclic coordinate descent

Fig. 2 a 3 shows graphs of times for an individual computation with CCD method. The computation of a third point and step size set to 100°, the number of links 30 and more. The mechanism has not reached the desired point after \( n \times 10000 \) cycles. For this reason there are zero values in the table, Tab. 1.

Same problem occurred with the third point and the movement on the track with a max angle step 1° and a number of links 50, Tab. 2.

<table>
<thead>
<tr>
<th>Time [s]</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path 1</td>
<td>0,136</td>
<td>0,536</td>
<td>2,400</td>
<td>6,400</td>
<td>8,830</td>
<td>27,562</td>
<td>47,782</td>
<td>81,993</td>
<td>126,907</td>
<td>182,757</td>
</tr>
<tr>
<td>Path 2</td>
<td>0,067</td>
<td>0,457</td>
<td>1,882</td>
<td>2,572</td>
<td>5,320</td>
<td>10,792</td>
<td>42,308</td>
<td>61,188</td>
<td>83,604</td>
<td>113,460</td>
</tr>
<tr>
<td>Path 3</td>
<td>0,079</td>
<td>0,571</td>
<td>2,278</td>
<td>5,788</td>
<td>11,464</td>
<td>19,584</td>
<td>30,981</td>
<td>46,562</td>
<td>62,888</td>
<td>91,520</td>
</tr>
<tr>
<td>Path 4</td>
<td>0,193</td>
<td>0,898</td>
<td>3,198</td>
<td>8,086</td>
<td>18,926</td>
<td>37,964</td>
<td>67,749</td>
<td>113,359</td>
<td>166,890</td>
<td>234,334</td>
</tr>
<tr>
<td>Path 5</td>
<td>0,111</td>
<td>0,823</td>
<td>2,763</td>
<td>7,897</td>
<td>14,608</td>
<td>25,456</td>
<td>40,538</td>
<td>60,932</td>
<td>83,399</td>
<td>125,879</td>
</tr>
<tr>
<td>Path 6</td>
<td>0,114</td>
<td>0,868</td>
<td>2,921</td>
<td>6,190</td>
<td>11,582</td>
<td>20,288</td>
<td>31,874</td>
<td>47,114</td>
<td>67,325</td>
<td>94,081</td>
</tr>
<tr>
<td>Path 7</td>
<td>0,309</td>
<td>3,369</td>
<td>54,841</td>
<td>113,994</td>
<td>0,000</td>
<td>0,000</td>
<td>0,000</td>
<td>0,000</td>
<td>0,000</td>
<td>0,000</td>
</tr>
<tr>
<td>Path 8</td>
<td>0,324</td>
<td>2,395</td>
<td>11,343</td>
<td>27,405</td>
<td>53,639</td>
<td>97,185</td>
<td>171,605</td>
<td>252,785</td>
<td>344,304</td>
<td>503,813</td>
</tr>
<tr>
<td>Path 9</td>
<td>0,284</td>
<td>2,384</td>
<td>8,680</td>
<td>21,999</td>
<td>45,443</td>
<td>84,939</td>
<td>141,645</td>
<td>217,743</td>
<td>307,705</td>
<td>456,498</td>
</tr>
</tbody>
</table>
3.2 Incremental cyclic coordinate descent

Fig. 4 a 5 shows graphs of times for the individual computation for a method iCCD. The computation the path at point 3, step size set to 1° and the number of links 50 and more. The mechanism has not reached the desired point after n*10000 cycles. For this reason there are zero values in the table, tab. 3 and 4.

Table 3: Times for calculating the point with iCCD method.

<table>
<thead>
<tr>
<th>time [s]</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point 1 1°</td>
<td>0.035</td>
<td>0.073</td>
<td>0.309</td>
<td>0.776</td>
<td>3.782</td>
<td>6.417</td>
<td>10.135</td>
<td>15.312</td>
<td>21.706</td>
<td>29.543</td>
</tr>
<tr>
<td>Point 1 10°</td>
<td>0.026</td>
<td>0.212</td>
<td>0.713</td>
<td>1.835</td>
<td>3.719</td>
<td>6.471</td>
<td>10.364</td>
<td>15.658</td>
<td>22.627</td>
<td>31.258</td>
</tr>
<tr>
<td>Point 1 100°</td>
<td>0.038</td>
<td>0.209</td>
<td>0.735</td>
<td>1.823</td>
<td>3.664</td>
<td>6.504</td>
<td>10.419</td>
<td>15.661</td>
<td>22.406</td>
<td>31.169</td>
</tr>
<tr>
<td>Point 2 1°</td>
<td>0.056</td>
<td>0.148</td>
<td>0.324</td>
<td>0.937</td>
<td>3.097</td>
<td>5.533</td>
<td>8.904</td>
<td>11.435</td>
<td>14.279</td>
<td>17.150</td>
</tr>
<tr>
<td>Point 2 10°</td>
<td>0.022</td>
<td>0.184</td>
<td>0.690</td>
<td>1.511</td>
<td>3.133</td>
<td>6.230</td>
<td>8.534</td>
<td>10.950</td>
<td>16.620</td>
<td>19.466</td>
</tr>
<tr>
<td>Point 2 100°</td>
<td>0.043</td>
<td>0.195</td>
<td>0.651</td>
<td>1.647</td>
<td>3.168</td>
<td>5.657</td>
<td>9.120</td>
<td>13.781</td>
<td>19.830</td>
<td>26.949</td>
</tr>
<tr>
<td>Point 3 1°</td>
<td>0.141</td>
<td>0.755</td>
<td>2.449</td>
<td>5.170</td>
<td>9.254</td>
<td>15.010</td>
<td>21.562</td>
<td>26.845</td>
<td>24.706</td>
<td>28.009</td>
</tr>
<tr>
<td>Point 3 10°</td>
<td>0.118</td>
<td>0.539</td>
<td>1.344</td>
<td>3.364</td>
<td>6.526</td>
<td>5.140</td>
<td>11.910</td>
<td>56.669</td>
<td>46.003</td>
<td>21.148</td>
</tr>
<tr>
<td>Point 3 100°</td>
<td>0.101</td>
<td>0.505</td>
<td>1.430</td>
<td>2.319</td>
<td>3.998</td>
<td>6.405</td>
<td>7.941</td>
<td>11.140</td>
<td>17.332</td>
<td>45.435</td>
</tr>
</tbody>
</table>

Table 4: Times for calculation desired point on the path with iCCD method.

<table>
<thead>
<tr>
<th>time [s]</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path 1 1°</td>
<td>0.129</td>
<td>0.493</td>
<td>2.256</td>
<td>6.056</td>
<td>14.873</td>
<td>27.046</td>
<td>45.013</td>
<td>76.148</td>
<td>111.710</td>
<td>176.468</td>
</tr>
<tr>
<td>Path 1 10°</td>
<td>0.062</td>
<td>0.435</td>
<td>1.717</td>
<td>5.044</td>
<td>11.843</td>
<td>23.064</td>
<td>15.347</td>
<td>23.578</td>
<td>79.357</td>
<td>48.142</td>
</tr>
<tr>
<td>Path 1 100°</td>
<td>0.073</td>
<td>0.493</td>
<td>1.999</td>
<td>4.964</td>
<td>9.861</td>
<td>17.077</td>
<td>27.102</td>
<td>41.341</td>
<td>58.715</td>
<td>82.051</td>
</tr>
<tr>
<td>Path 2 1°</td>
<td>0.179</td>
<td>0.864</td>
<td>3.012</td>
<td>7.829</td>
<td>18.570</td>
<td>36.683</td>
<td>70.122</td>
<td>104.240</td>
<td>169.993</td>
<td>248.164</td>
</tr>
<tr>
<td>Path 2 10°</td>
<td>0.141</td>
<td>0.783</td>
<td>2.613</td>
<td>7.610</td>
<td>14.005</td>
<td>22.367</td>
<td>36.287</td>
<td>57.265</td>
<td>80.107</td>
<td>107.370</td>
</tr>
<tr>
<td>Path 2 100°</td>
<td>0.099</td>
<td>0.874</td>
<td>2.862</td>
<td>6.206</td>
<td>11.528</td>
<td>20.013</td>
<td>31.161</td>
<td>45.986</td>
<td>65.892</td>
<td>90.376</td>
</tr>
<tr>
<td>Path 3 1°</td>
<td>0.310</td>
<td>3.319</td>
<td>23.524</td>
<td>114.173</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Path 3 10°</td>
<td>0.317</td>
<td>2.267</td>
<td>10.997</td>
<td>26.712</td>
<td>52.412</td>
<td>94.981</td>
<td>163.662</td>
<td>233.079</td>
<td>330.214</td>
<td>476.499</td>
</tr>
<tr>
<td>Path 3 100°</td>
<td>0.276</td>
<td>2.372</td>
<td>8.566</td>
<td>21.666</td>
<td>44.685</td>
<td>83.221</td>
<td>138.506</td>
<td>213.658</td>
<td>302.851</td>
<td>444.241</td>
</tr>
</tbody>
</table>
4. Shape of n-link chain

As a further criterion for the select right inverse kinematic calculation algorithm can by the shape of mechanism after achieving the desired position. This section includes a results from the calculation when the start point has same characteristic like before and the end point has characteristic of the second point from a previous part. The mechanism has shape with 100 links.

4.1 Cyclic coordinate descent

Disadvantages of CCD method is that when is setup higher value of the max step size. The calculation is quick, but the final shape of mechanism is twisted. This happens with the smaller number of links. If there is higher number of links and the large distance between the start point and the end point, the mechanism didn’t reached the end point (tab.1). Like is shown on Fig. 6 for the max angle step size 90° and 180°.

For 100 links, maximal cycle number n*10000 and maximal angle step 1° reached the desired point for 8207 cycles, and for maximum step 90° and 180°, the mechanism didn’t get to the desired position.

4.2 Incremental CCD

With this method are all shapes about the same regardless on the max angle step size. Like is shown on Fig. 7. Every mechanism achieved the desired position for about 32 500 cycles.

5. Conclusion

The inverse kinematics of mechanical systems is a complex issue. To a date, there are several methods for its solution. Not all are applicable to mechanisms with more degrees of a freedom because of the complex and the subsequent compilation of Jacobian and his numerical complexity. This article presents a comparison of the CCD method and its modified version incremental CCD in terms of a speed of calculation, the number of cycles, but also shape of mechanism after achieving the desired point.

Calculation with the CCD method has reached a low number of cycles to achieve the desired position but maximal angle step size must be setup to value 1°. For higher values, there is the “twist” of mechanism and the subsequent slow convergence to the desired point. The iCCD first leads to movement of end links with successive addition of links, with which performs the movement, which
is advantageous from the view point of the shape of mechanism for achieving the desired position when the end point is orientated towards the desired point. There is no way to twist mechanism, at the expense of cycles and the time.

The results show that the calculation method by the CCD is advantageous for "finding" a single point and the ICCD method is turn advantageous to move the end point of the mechanism on the pre-defined trajectory when the time is short compared to the calculation method CCD. The final confirmation of this conclusion for the whole workspace mechanism is necessary to perform additional calculations and simulations.

6. Acknowledgement

The paper has been created realization of the project "Centrum výskumu riadenia technických, environmentálnych a humánnych rizík pre trvalý rozvoj produkcie a výrobných v strojárstve" (IMTS:26220120060), based on operating program support Research and Development financed from European Regional Development Fund and of project applied study - Stimuly "Komplexný modulárny robotický systém strednej kategórie s vyššou inteligenciou" (Req-00169-0001).

7. References


