Alternative Constraint Handling Technique for Four-Bar Linkage Motion Generation

S Sleesongsom¹ and S Bureerat²

¹Department of Aeronautical Engineering and Commercial Pilot, International Academy of Aviation Industry, King Mongkut's Institute of Technology Ladkrabang, Bangkok, 10520, Thailand
²Sustainable and Infrastructure Research and Development Center, Department of Mechanical Engineering, Faculty of Engineering, KhonKaen University, KhonKaen City, 40002, Thailand

*Corresponding author: Suwin Sleesongsom: suwin.se@kmitl.ac.th

Abstract This paper proposes an extension of a new concept for four-bar path generation and a new penalty constraint handling technique from our previous work to a motion generation problem. The proposed technique could neglect the timing constraint for problems without prescribed timing, which is found to be an efficient technique for a four-bar path generation problem. Later, the remaining constraints are solved with a new constraint handling technique that also gives high performance. The technique is one kind of penalty function techniques. The present work proposes to combine the previous two methods for solving the four-bar motion generation problems. The comparative study, optimisation of motion generation problems with a new and a traditional penalty technique are solved by using self-adaptive population size teaching-learning based optimization (SAP-TLBO), is presented. In this study, three new motion generation test problems are used to test the proposed technique. The results show that the new technique gives acceptable results and can be applied with the motion generation problems without prescribed timing.

1. Introduction

Some applications of a four-bar linkage are window wipers, door closing mechanisms, and rock crushers among others that can be easily found in our daily life. Approximately, there are more than one thousand applications found in manufacturing [1]. This is a reason why this mechanism has been studied continuously until the present time [2-7]. The dimensional synthesis is one kind of mechanism design, which can be subdivided into three types i.e. path generation, motion generation and function generation. In this research, we adapt the previous techniques [1, 8] used for a path generation problem to study the motion generation [9], but the design problem is adopted from our previous work [1]. The path generation or path synthesis problem is a technique used to find the optimum dimensions of linkage where a point on a coupler link moves along the desired path [1,8], while motion generation or motion synthesis has a set of desired points and desired angles as a goal [9]. The design problem is set in the form of an optimisation problem with a penalty constraint handling technique [9]. Later an efficient technique to avoid the constraint of a crank angle sequence is proposed [1]. This technique
has been proved to enhance the performance of the path generation synthesis. Thus, it makes sense to combine with the new constraint handling technique [8] and the new combination can enhance the performance of the motion generation problem. The proposed optimiser in this study is an efficient technique for solving path generation problem [1]. Then, in this research the performance of the proposed technique is studied. The present work is different from the very recently work by the same authors [10] in that they proposed the new constraint handling technique without using a penalty technique. The work only applied with the path synthesis problem, which is an efficiency technique.

The rest of this paper starts from Section 2 that details the position analysis of a four-bar linkage in brief, objective function, and the constraint handling technique. The design problems and optimiser are presented in Section 3. The design results are detailed in Section 4. The conclusions and discussion of the study are summarized in Section 5.

2. Methodology

2.1 Position analysis

A model of a four-bar linkage in this study is composed of four simple links connected with 4 simple joints. A variety of linkage types occur when assigning anyone link to be a frame. This linkage has one degree of freedom, which can operate with one actuator. The kinematic diagram of this linkage is shown in Fig. 1. The position analysis of the four-bar linkage uses a trigonometric formula for the relation of linkage lengths \( r_1, r_2, r_3, \) and \( r_4 \) and its other parameters, which is found in standard mechanics of machinery textbooks as mentioned in [1, 8]. The coupler point \((P)\) in the global coordinate can be expressed as

\[
x_P = x_{O2} + r_2 \cos(\theta_2 + \theta_1) + L_1 \cos(\phi_0 + \theta_3 + \theta_1)
\]

\[
y_P = y_{O2} + r_2 \sin(\theta_2 + \theta_1) + L_1 \sin(\phi_0 + \theta_3 + \theta_1)
\]

where \(x_{O2}\) and \(y_{O2}\) are the coordinates position of the \(O_2\) in the global coordinates [8]. The angles \(\phi, \theta_3, \theta_4, \) and \(\gamma\) inside the kinematic diagram relate the known link lengths \(r_1, r_2, r_3, \) and \(r_4 \) at any crank angle \(\theta_2\) by law of cosine.

![Four-bar linkage in the global coordinate system](image)

**Figure 1.** Four-bar linkage in the global coordinate system [1]

2.2 Optimization problem and constraint handling

The motion generation problem is different from the path generation problem as only the objective function includes the desired angles of the coupler link. As a result to the objective function of this problem composes of two parts where the first part is the position error between the desired points...
The design constraint usually can use an external penalty technique to handle the constraints by adding them to the objective function (2). For this case there are two parts of penalty function value. The first part is assigned to control link lengths to meet the Grashof’s criteria (3-4). The second part is assigned to ensure the input crank can rotate with a complete revolution in either a clockwise or counter clockwise direction (5). The penalty function works by adding a very high value to modify the objective function when some of the constraints are violated. It was found this technique is inefficient for solving the above design problem, as a result, the unknowns \( \theta_i \) are solved for by another technique and removed from the design variables [1, 8]. If the mechanism is a crank-rocker, this means constraint (4) and (5) are met. To tackle both constraints a new technique from [8] is used instead of the traditional penalty technique, but it has never been used for solving a motion generation problem. The constraints can be written in a common form as \( g_i(x) \leq g_0 \). Then, a set of \( N \) input angle values (\( \theta_i \)) is generated that are equally spaced from 0 to \( 2\pi \) radian. Many more intervals indicate higher efficiency, but the disadvantage is that computationally it is more time consuming. Hence the positions of point \( P \) corresponding to all targets are calculated, the objective function is

\[
f(x) = \sum_{j=1}^{N} \min_{i} d_{ij}^2
\]

where \( d_{ij}^2 = (x_{d,i} - x_{p,j})^2 + (y_{d,i} - y_{p,j})^2 \) for \( j = 1, ..., N \). The detail of this technique can be seen in [1, 8].

The usual penalty technique adds a constraint penalty term to the objective function with the constant parameter multiplied with the constraint function. If the design variables are infeasible, a large constant is added to modify the objective function value. Otherwise, the constant is 0. With this technique, the constraints can be induced and approach a feasible solution. For this reason the size of a constant is a sensitive parameter which affects the search performance of an optimizer. An alternative penalty technique was proposed by [8] for solving the path generation problem was shown to be an efficient technique for constraint handling. The focus of this research is to apply this technique in

\[
P_d(x_d, y_d) \text{ and the actual points } P(x_p, y_p). \text{ The second part of objective function is term of the angular error between desire angles } (\theta_d) \text{ and actual angles } (\theta_p). \text{ Then, the design variables includes } r_1, r_2, r_3, r_4, L_1, L_2, \text{ and the coordinates of } O_2 (x_{O2}, y_{O2}) \text{ and the angle of frame } I(\theta). \text{ In this research is focusing only the motion generation problem type, which is called synthesis without prescribed timing. The input set of } \theta_i \text{ values is also set as design variables. The optimization problem without prescribed timing is then written as}
\]

\[
\text{Min} f(x) = \sum_{i=1}^{N} \left[ (x_{d,i} - x_{p,i})^2 + (y_{d,i} - y_{p,i})^2 + (\theta_{d,i} - \theta_{p,i})^2 \right]
\]

subject to

\[
\min(r_1, r_2, r_3, r_4) = \text{crank}(r_2)
\]

\[
2\min(r_1, r_2, r_3, r_4) + 2 \max(r_1, r_2, r_3, r_4) < (r_1 + r_2 + r_3 + r_4)
\]

\[
\theta_1^i < \theta_2^i < \cdots < \theta_N^i
\]

\[
x_{l} \leq x \leq x_{u}
\]
solving four-bar linkage motion generation problems. This technique creates a new equivalent objective function by using four constraint handling rules:

1) Any feasible solution has a better equivalent objective function value than any infeasible one;
2) Between two feasible solutions, the one with the better objective function value has a better equivalent objective function value;
3) Between two infeasible solutions, the one with a similar constraint violation has better equivalent objective function value; and
4) As infeasible solution with a small constraint violation ($c_i < c_0 > 0$) is considered feasible in the early stages of an optimisation run where $c_0$ is a small positive constraint tolerance.

The fourth rule is used to explore promising infeasible solutions which are near constraint boundaries. This “nearby rule” usually enables an algorithm to reach the real optimum faster. When constructing a new equivalent unconstrained objective function from rules 1-3, the function can be defined as [8]

$$f_p(x, c_0) = \begin{cases} f'(x) & \text{if } \max_i c_i(x) \leq c_0 \\ \frac{f(x)}{1 + k} & \text{otherwise} \end{cases}$$

where $k > 0$ is a constant to be defined (for this paper $k = 10$), and $c_0$ starts from a positive value and is iteratively reduced to 0 at about a half iteration.

The constraint equation should be rearranged in normalized form first before being used with (8) as follows

$$c_i(x) = g_i(x) / g_{0i} - 1 \leq 0$$

where $g_{0i}$ is an allowable value for the constraint $g_i$.

### 3. Numerical Experiment

Three optimisation problems without prescribed timing are used to study the performance of the proposed design technique. The design problems are detailed as follows:

**Case-1**: Design variables for $x$ are

$$x = [r_1, r_2, r_3, r_4, L_1, L_2, x_{02}, y_{02}, \theta_1, \theta_2^1, \theta_2^2, \theta_2^3, \theta_2^4, \theta_2^5, \theta_2^6]$$

Target points are $(x_{0d}, y_{0d})$ and $\theta_{3d}$

$$(x_{0d}, y_{0d}) = [(20, 20), (20, 25), (20, 30), (20, 35), (20, 40), (20, 45)]$$

$$\theta_{3d} = [1.9937, 1.9220, 1.8434, 1.7599, 1.6709, 1.5735] \text{ rad}$$

Limits of the variables:

- $5 \leq r_1, r_2, r_3, r_4 \leq 60$
- $-60 \leq L_1, L_2, x_{02}, y_{02} \leq 60$
- $0 \leq \theta_1^1, \theta_1^2, \ldots, \theta_6^6 \leq 2\pi$

**Case-2**: Design variables for $x$ are

$$x = [r_1, r_2, r_3, r_4, L_1, L_2, x_{02}, y_{02}, \theta_1, \theta_2^1, \theta_2^2, \theta_2^3, \theta_2^4, \theta_2^5, \theta_2^6, \theta_2^7, \theta_2^8, \theta_2^9, \theta_2^{10}]$$

Target points are $(x_{0d}, y_{0d})$ and $\theta_{3d}$

$$(x_{0d}, y_{0d}) = [(20, 10), (17.66, 15.142), (11.736, 17.878), (5, 16.928), (0.60307, 12.736), (0.60307, 7.2638), (5, 3.0718), (11.736, 2.1215), (17.66, 4.8577), (20, 10)]$$

$$\theta_{3d} = [0.4208, 0.5117, 0.7433, 0.9910, 1.1394, 1.1296, 0.9599, 0.7322, 0.5257, 0.4208] \text{ rad}$$
Limits of the variables:
\[5 \leq r_1, r_2, r_3, r_4 \leq 80\]
\[-80 \leq L_1, L_2, x_{O2}, y_{O2} \leq 80\]
\[0 \leq \theta_0, \theta_1^1, \ldots, \theta_2^{10} \leq 2\pi\]

**Case-3**: Design variables for \( x \) are:
\[x = [r_1, r_2, r_3, r_4, L_1, L_2, x_{O2}, y_{O2}, \theta_1^1, \theta_2^2, \theta_2^3, \theta_2^4, \theta_2^5, \theta_2^6, \theta_2^7, \theta_2^8, \theta_2^9, \theta_2^{10}, \theta_2^{11}, \theta_2^{12}, \theta_2^{13}, \theta_2^{14}, \theta_2^{15}, \theta_2^{16}, \theta_2^{17}, \theta_2^{18}, \theta_2^{19}, \theta_2^{20}, \theta_2^{21}]\]

Target points are \((x_{d}, y_{d})\) and \(\theta_{3d}\):
\[(x_{d}, y_{d}) = [(0, 0), (0.9356, 0.4064), (1.5983, 0.8220), (2.1704, 1.3002), (2.6621, 1.7986), (2.9957, 2.2692), (3.2614, 2.7276), (3.4941, 3.1408), (3.6635, 3.5485), (3.7798, 3.9623), (3.7992, 4.4734), (3.7170, 4.8783), (3.4675, 5.3624), (3.1242, 5.6980), (2.7158, 5.8800), (2.2517, 5.8780), (1.7997, 5.6882), (1.2732, 5.1741), (0.8336, 4.5628), (0.4003, 3.9064), (0.3, 3.38)]\]
\[\theta_{3d} = [0, 0, 0, 0, 0, 0, 0.06984127, 0.13968254, 0.261904762, 0.436507937, 0.611111111, 0.838095238, 1.047619048, 1.30952381, 1.571428571, 1.658730159, 1.746031746, 1.833333333, 1.746031746, 1.658730159, 1.571428571] \text{ rad}\]

Limits of the variables:
\[5 \leq r_1, r_2, r_3, r_4 \leq 80\]
\[-80 \leq L_1, L_2, x_{O2}, y_{O2} \leq 80\]
\[0 \leq \theta_0, \theta_1^1, \ldots, \theta_2^{10} \leq 2\pi\]

In order to measure the performance of the new constraint handling technique for solving four-bar motion generation problems, SAP-TLBO is used to tackle the design problems. The parameter settings of this algorithm are defined as IReset = 20, IRange = 5, and \(\delta = 1\). The details of this optimiser can be seen in [1].

All algorithms are coded in MATLAB commercial software. In this study the population size is set \(n_p = 100\) for all algorithms, while the maximum number of iterations is 500. The number of run times of each algorithm is set to be 30 times to study statistical performance. The value of \(c_0 = 30\) at the initial and iteratively reduced to 0 at approximately a half of an optimization run to make it more diversified for the optimizers.

### 4. Design Results

The design results are shown in Table 1. The table shows the mean objective function values from 30 optimization runs, the worst result (max), the best result (min), the standard deviation (std), and the number of successful runs (a run that gives a feasible solution). Figs. 2-4 show the best path and angle traced by the coupler point and its kinematic diagram of the best linkages. In Case-1, there are 6 target points and angles. It was found that SAP-TLBO with the new penalty technique gives the best result (error = 0.002591) and the best mean objective value (error = 0.007623), which is better than the original technique. The result of Case-2 showed that SAP-TLBO with the new penalty technique gives the better result than the original one in both mean (error = 0.088245) and min (error = 0.035899). The last result of Case-3 showed that SAP-TLBO with the new penalty technique gives a better result than the original one in both mean (error = 0.407154) and min (error = 0.378623). The number of successful runs of the new penalty is quite similar to the traditional penalty technique as shown in Table 1. The results show that SAP-TLBO with the new technique is better than those using the traditional technique in all cases.
Figure 2. The best coupler curves and the best mechanism of Case-1.

Figure 3. The best coupler curves and the best mechanism of Case-2.

Figure 4. The best coupler curves and the best mechanism of Case-3.
Table 1. Design results of motion generation problem

| Parameters | SAP-TLBO | Case-1 | Case-2 | Case-3 |
|------------|----------|--------|--------|--------|
|            | New penalty | Traditional penalty | New penalty | Traditional penalty | New penalty | Traditional penalty |
| \( r_1 \)  | 0.455280   | 0.747894 | 0.093053 | 0.808245 | 0.900515   | 0.997974 |
| \( r_2 \)  | 0.256455   | 0.478707 | 0.054969 | 0.061523 | 0       | 5.77E-07 |
| \( r_3 \)  | 0.897185   | 0.579066 | 0.638748 | 0.472704 | 0.868487   | 0.253568 |
| \( r_4 \)  | 0.982635   | 0.722648 | 0.622398 | 0.461557 | 0.694746   | 0.744407 |
| \( L_1 \)  | 0.005589   | 0.840120 | 0.502475 | 0.548133 | 0.083085   | 0.573494 |
| \( L_2 \)  | 0.909715   | 0.199440 | 0.824103 | 0.257639 | 0.513827   | 0.657151 |
| \( x_0 \)  | 1          | 0.983981 | 0.558221 | 0.519631 | 0.703590   | 0.522978 |
| \( y_0 \)  | 0.911070   | 0.591794 | 0.554886 | 0.542629 | 0.921755   | 0.462580 |
| \( \theta_0 \) | 0.431581   | 0.084342 | 0.973713 | 0.779455 | 0.551639   | 0.965804 |
| mean       | 0.007623   | 0.209854 | 0.088245 | 1.036778 | 0.407154   | 8.105407 |
| max        | 0.019476   | 1.713401 | 0.507658 | 12.39277 | 0.544994   | 18.59596 |
| min        | 0.002591   | 0.019861 | 0.035899 | 0.375200 | 0.378623   | 6.389301 |
| std        | 0.003665   | 0.412596 | 0.098038 | 2.391289 | 0.035156   | 2.613243 |
| Success.*  | 25         | 22      | 25      | 25      | 24         | 30        |

* Success. = no. of successful runs

5. Conclusions
This paper proposed combining our previous efficient techniques for path generation of a four-bar linkage and applying them to motion generation problems. This technique is a combination of a technique to remove timing constraints from the optimization problem and a new penalty technique for handling Grashof’s criterion constraint. Numerical experiments demonstrated that the new technique with SAP-TLBO outperformed the same optimiser with the traditional penalty technique. However, this is a possibility study of an alternative technique for solving motion generation problem without prescribed timing. For future work the other parameters that affect the performance of this design technique should be investigated.

Acknowledgments
The authors are grateful for the financial support provided by King Mongkut’s Institute of Technology Ladkrabang, the Thailand Research Fund, and the Post-doctoral Program from Research Affairs, Graduate School, KhonKaen University (58225).

References
[1] Sleesongsom S and Bureerat S 2017 Knowledge-Based Systems 135 180–191
[2] Smaili A and Diab N 2007 Mechanism and Machine Theory 42(1) 115–130
[3] Cabrera J A, Nadal F, Muñoz J P and Simon A 2007 Mechanism and Machine Theory 42(7) 791-806
[4] Acharyya S K and Mandal M 2009 Mechanism and Machine Theory 44(9) 1784-1794
[5] Lin W Y 2010 *Mechanism and Machine Theory* **45**(8) 1096–1107
[6] Peñuñuri F, Peón-Escalante R, Villanueva C and Pech-Oy D 2011 *Mechanism and Machine Theory* **46**(10) 1335–1349
[7] Ebrahimi S and Payvandy P 2015 *Mechanism and Machine Theory* **85** 189–204
[8] Sleesongsom S, Bureerat S 2018 *IOP Conf. Series: Materials Science and Engineering* **324** 012012
[9] Gogate G R, Mateka S B 2012 *Mechanism and Machine Theory* **52** 158–179
[10] Sleesongsom S, Bureerat S 2018 *Computational Intelligence and Neuroscience* (in press).