APPLICATION OF LÉVY-FLIGHT INTENSIFIED CURRENT SEARCH TO OPTIMAL PID CONTROLLER DESIGN FOR ACTIVE SUSPENSION SYSTEM

WATTANARWONG ROMSAI, AUTTARAT NAWIKAVATAN AND DEACHA PUANGDOWNREONG

Department of Electrical Engineering
Southeast Asia University
19/1 Petchkasem Road, Nongkhangphlu, Nonghkaem, Bangkok 10160, Thailand
deachap@sau.ac.th

Received November 2020; revised February 2021

ABSTRACT. This paper presents the application of the Lévy-flight intensified current search (LFICuS) to optimally design the proportional-integral-derivative (PID) controller for the active vehicle suspension system. The LFICuS is one of the newest metaheuristic optimization search techniques modified from the intensified current search (ICuS) based on the original current search (CuS) which was initiated from the flowing behavior of the electrical current through the electric networks. The random number drawn from the Lévy-flight distribution, the adaptive radius (AR) mechanism and the adaptive neighborhood (AN) mechanism are conducted in the LFICuS algorithm to improve its search performance. The proposed LFICuS is tested against ten selected standard multimodal benchmark functions for minimization to perform its effectiveness compared with the ICuS. As results, the proposed LFICuS is much more efficient for function minimization than the ICuS. The LFICuS is then applied to design the optimal PID controller for the active vehicle suspension system. Based on the modern optimization, the LFICuS can provide optimal PID controller for the active suspension system. The considered suspension system controlled by the PID controller designed by the LFICuS yields very satisfactory response superior to the passive suspension system, significantly.

Keywords: Lévy-flight intensified current search, PID controller, Active vehicle suspension system, Modern optimization

1. Introduction. Recently, many theoretical and real-world combinatorial and numeric optimization problems can be successfully solved by metaheuristic algorithms [1-3]. From a literature review, the current search (CuS) is one of the most interesting trajectory-based metaheuristic algorithms [4]. The CuS development scenario begins in 2012 once it was firstly proposed as an optimizer to solve the optimization problems [4]. The CuS algorithm mimics the principle of an electric current flowing behavior of the electric circuits and networks. It performed superior search performance to the genetic algorithm (GA), tabu search (TS) and particle swarm optimization (PSO) [4]. The CuS was successfully applied to control engineering [5] and signal processing [6].

During 2013-2014, the adaptive current search (ACuS) was launched [7] as a modified version of the CuS. The ACuS consists of the memory list (ML) used to escape from local entrapment caused by any local solution and the adaptive radius (AR) mechanism conducted to speed up the search process. The ACuS was successfully applied to industrial engineering [7] and energy resource management [8]. For some particular problems, both the CuS and ACuS are trapped by local optima and consume much search time.

DOI: 10.24507/ijicic.17.02.483
In 2014, the intensified current search (ICuS) was proposed to improve its search performance [9]. The ICuS algorithm consists of the ML, AR and the adaptive neighborhood (AN) mechanisms. The ML regarded as the exploration strategy is used to store the ranked initial solutions at the beginning of search process, record the solution found along each search direction, and contain all local solutions found at the end of each search direction. The ML is also applied to escape the local entrapments caused by local optima. The AR and AN mechanisms regarded as the exploitation strategy are together conducted to speed up the search process. The ICuS was successfully applied to many control engineering problems including single-objective and multi-objective optimization problems [9-12]. For some optimization problems especially in large-space multimodal problems, the ICuS might be trapped by local optima. This is properly because the random number with the uniform distribution in the ICuS algorithm is not efficient enough for such the problems. Thus, it needs to be modified to enhance its search performance and to speed up the search process.

In this paper, the new trajectory-based metaheuristic algorithm named the Lévy-flight intensified current search (LFICuS) is proposed. The proposed LFICuS is the newest modified version of the ICuS. The random number drawn from the Lévy-flight distribution, the AR and AN mechanisms are conducted in the proposed LFICuS algorithm to improve its search performance. Although the proposed LFICuS is based on the ICuS which is one of the trajectory-based metaheuristic algorithms, it consists of many search directions and the AR and AN mechanisms in which the random number drawn from the Lévy-flight distribution is applied. It makes the LFICuS have strong exploitative and explorative properties. The proposed LFICuS algorithm is then applied to design the optimal PID controller for the active vehicle suspension system via the state-variable model based on the modern optimization. This paper consists of six sections. After an introduction given in Section 1, the proposed LFICuS algorithms are described in Section 2. Performance evaluation of the proposed LFICuS algorithms against ten selected standard multimodal benchmark functions are reported in Section 3. Problem formulation of the LFICuS-based PID controller design optimization for the active vehicle suspension system is formulated in Section 4. Results and discussions are illustrated in Section 5. Finally, conclusions are given in Section 6.

2. ICuS and LFICuS Algorithms. In this section, the ICuS algorithm is briefly described. Then, the proposed LFICuS algorithm is elaborately illustrated.

2.1. ICuS algorithm. The ICuS algorithm [9] developed from the CuS [4] is based on the iteratively random search by using the random number drawn from the uniform distribution. The ICuS possesses the ML regarded as the exploration strategy, the AR and AN mechanisms regarded as the exploitation strategy. The ML is used to escape from local entrapment caused by any local solution. The ML consists of three levels: low, medium and high. The low-level ML is used to store the ranked initial solutions at the beginning of search process, the medium-level ML is conducted to store the solution found along each search direction, and the high-level ML is used to store all local solutions found at the end of each search direction. The AR mechanism conducted to speed up the search process is activated when a current solution is relatively close to a local minimum by properly reducing the search radius. The radius is thus decreased in accordance with the best cost function found so far. The less the cost function, the smaller the search radius. The AN mechanism also applied to speed up the search process is invoked once a current solution is relatively close to a local minimum. The neighborhood members will be decreased in accordance with the best cost function found. The less the cost function,
the smaller the neighborhood members. With ML, AR and AN, a sequence of solutions obtained by the ICuS very rapidly converges to the global minimum.

Algorithms of the ICuS can be described by the pseudo code shown in Figure 1, where \( f(\mathbf{x}) \) is the objective function, \( \mathbf{x} = (x_1, \ldots, x_d)^T \) is the solution vector, \( \Omega \) is the search space, \( \Psi \) is the low-level ML, \( \Gamma_k \) is the medium-level ML of the \( k \)th search direction, \( \Xi \) is the high-level ML, \( j_{\text{max}} \) is the maximum allowance of the solution cycling, \( N \) is number of initial solutions (search directions), \( n \) is number of neighborhood members, \( R \) is the search radius, \( i \) and \( j \) are the counting variables, \( X_i \) are the initial solutions, \( x_0 \) is the selected initial solution, \( X_{\text{local}} \) is the local solution, \( X_{\text{global}} \) is the global solution, \( \mathbf{x}^* \) is the best solution in the current search iteration, \( \rho \) is the decreasing factor for the AR and AN mechanisms and TC is the termination criteria.

---

**Figure 1.** Pseudo code of the ICuS algorithm

From Figure 1, after initializing the objective function \( f(\mathbf{x}) \) and relevant search parameters \((i = j = k = 1)\), the ICuS begins the search process by randomly generating the initial solutions \( X_i \). Then, all \( X_i \) will be evaluated by the objective function \( f(X_i) \), ranked and stored in \( \Psi \). After that, the \( k \)th solution in \( \Psi \) is selected and set as the \( x_0 = X_k \). \( X_{\text{global}} = X_{\text{local}} = x_0 \) is also set. From this, the ICuS goes to the search iteration. If \( k \leq N \) and \( j \leq j_{\text{max}} \), the solutions \( x_i \) are randomly generated around \( x_0 \) within \( R \) by using random number drawn from the uniform distribution. All \( x_i \) will be evaluated by the objective function \( f(x_i) \). The best solution among \( x_i \) will be set as \( x^* \).
If \( f(x^*) < f(x_0) \), keep \( x_0 \) into \( \Gamma_k \), update \( x_0 = x^* \) and set \( j = 1 \), otherwise, keep \( x^* \) into \( \Gamma_k \) and update \( j = j + 1 \). The AR and AN mechanisms are thus activated to reduce \( R \) and \( n \), respectively. Then, \( X_{\text{local}} \) is updated by setting \( X_{\text{local}} = x_0 \). \( X_{\text{global}} \) is stored in \( \Xi \). If \( f(X_{\text{local}}) < f(X_{\text{global}}) \), \( X_{\text{global}} \) will be updated by setting \( X_{\text{global}} = X_{\text{local}} \). After that, update \( k = k + 1 \) and set \( j = 1 \) in order to proceed the next search direction. The ICuS algorithms will be iteratively proceeded until all significant local solutions are found and stored in \( \Xi \). Then, one of the local solutions is the global solution found.

Movements of the ICuS over 2D-search space of function minimization can be observed by Figure 2, where \( f(x, y) \) is the objective function, \( x \) and \( y \) are the solutions, \( D_N \) is the \( N \)th search direction (\( N = 3 \) is assumed in Figure 2), \( R_i \) is the initial search radius, \( R_f \) is the final search radius, \( n_i \) is the neighborhood members and \( n_f \) is the final neighborhood members.

**Figure 2.** (color online) Movements of the ICuS over 2D-search space of function minimization

2.2. **Proposed LFICuS algorithm.** Referring to Figure 1, the ICuS algorithm employs the random number drawn from the uniform distribution for generating the neighborhood members as feasible solutions. The probability density function (PDF) \( f(x) \) of the continuous uniform distribution can be expressed in (1), where \( a \) and \( b \) are lower and upper bounds of random process. Mean \( \mu \) and variance \( \sigma^2 \) of the continuous uniform distribution are limited by bounds of random process. The random number drawn from the uniform distribution is considered as non-scale-free characteristics. Another random number with scale-free characteristics is the random number drawn from the Lévy-flight distribution [13], where its PDF is stated in (2) and \( c \) is the scale parameter. The random with Lévy-flight distribution has an infinite mean and infinite variance [13]. Then, it is more efficient
than the random with uniform distribution. Many metaheuristic algorithms including the
cuckoo search (CS) [14] and flower pollination algorithm (FPA) [15] utilize the random
with Lévy-flight distribution for exploring the feasible solutions.

\[
f(x)_{\text{uniform}} = \begin{cases} 
    \frac{1}{b-a} & \text{for } a \leq x \leq b, \\
    0 & \text{for } x < a \text{ or } x > b
\end{cases}
\]

(1)

\[
f(x)_{\text{Lévy}} = \sqrt{\frac{c^2}{2\pi}} \cdot e^{-\frac{c^2}{2(x-\mu)^{3/2}}}
\]

(2)

The proposed LFICuS algorithm uses the random number drawn from the Lévy-flight
distribution to generate the neighborhood members as feasible solutions in each search
iteration. Once applied in the proposed LFICuS algorithm, the Lévy-flight distribution
\( L \) can be approximated by (3) [15], where \( s \) is step length, \( \lambda \) is an index and \( \Gamma(\lambda) \) is
the Gamma function as expressed in (4), when \( \Gamma(n) = (n-1)! \), \( \lambda = n \) (an integer).
The Lévy-flight distribution works well for large space. Therefore, the AR mechanism is
also conducted by setting the initial search radius \( R = \Omega \) (search space). The proposed
LFICuS algorithm can be described by the pseudo code shown in Figure 3. Movements of
the proposed LFICuS over 2D-search space of function minimization can be visualized by

**Figure 3.** Pseudo code of the proposed LFICuS algorithm

- **Initialized:**
  - Objective function \( f(x) \), \( x=(x_1, \ldots, x_d)^T \),
  - Search space \( \Omega \),
  - Memory list ML (\( \Psi \), \( \Gamma_\delta \), and \( \Xi \)),
  - Maximum allowance of solution cycling \( j_{\text{max}} \),
  - Number of initial solutions \( N \),
  - Number of neighborhood members \( n \),
  - Search radius \( R = \Omega \), \( k=j=1 \).
  - Uniformly random initial solution \( x_i \) within \( \Omega \).
  - Evaluate \( f(x_i) \) then rank \( x_i \) and store in \( \Psi \).
  - Let \( x_0 = X_i \) as selected initial solution.
  - \( x_{\text{global}} = x_{\text{local}} = x_0 \).

**while** (\( k \leq N \) or termination criteria: TC);

**while** (\( j < j_{\text{max}} \));

- Lévy-flight random \( x_i \) around \( x_0 \) within \( R \).
- Evaluate \( f(x_i) \) and set the best one as \( x^* \).
- if \( f(x^*) < f(x_0) \);
  - Keep \( x_0 \) into \( \Gamma_\delta \), update \( x_0 = x^* \) and set \( j=1 \).
else
  - Keep \( x^* \) into \( \Gamma_\delta \) and update \( j=j+1 \).
end

- Activate AR by \( R = \rho R \), \( 0 < \rho < 1 \).
- Invoke AN by \( n = \alpha n \), \( 0 < \alpha < 1 \).
end

- Update \( X_{\text{local}} = x_0 \).
- Keep \( x_{\text{global}} \) into \( \Xi \).
- if \( f(X_{\text{local}}) < f(X_{\text{global}}) \);
  - Update \( X_{\text{global}} = X_{\text{local}} \).
end

- Update \( k = k+1 \) and set \( j=1 \).
- Let \( x_0 = X_k \) as selected initial solution.
end
Figure 4. All notations used in Figure 3 and Figure 4 are the same meaning as those in Figure 1 and Figure 2, respectively. By comparison with the ICuS shown in Figure 2, it can be observed that the search performance of the LFICuS can be improved by greater search space at the beginning of search process. Moreover, the neighborhood members in all search iterations of the LFICuS are generated by the random number drawn from the Lévy-flight distribution, while those of the ICuS are generated by the random number drawn from the uniform distribution.

\[ L \approx \frac{\Lambda \Gamma(\lambda) \sin(\pi \lambda/2)}{\pi} \cdot \frac{1}{s^{1+\lambda}} \]  

(3)

\[ \Gamma(\lambda) = \int_{0}^{\infty} t^{\lambda-1} e^{-t} dt \]  

(4)

3. **Performance Evaluation.** The proposed LFICuS algorithm will be tested against ten selected standard multimodal benchmark functions for minimization to perform its effectiveness compared with the ICuS. Such ten selected benchmark functions as summarized in Table 1 [16,17], where \( x^* \) is the optimal solution, \( f(x^*) \) is the optimal function value, \( D \) is a dimension and \( f_{\text{max}} \) is the maximum allowance of \( f(x^*) \). All selected functions can be considered as the nonlinear and unsymmetrical functions which are very difficult for function minimization. For example, the 3D surface of the Schwefel function (SF), \( f_1(x) \), is plotted as shown in Figure 5.
Table 1. Summary of ten selected benchmark functions

<table>
<thead>
<tr>
<th>Function names</th>
<th>Functions, search space, optimal solution and optimal function value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Schwefel function (SF)</td>
<td>$f_1(x) = 418.9829D - \sum_{i=1}^{D} x_i \sin \left(\sqrt{</td>
</tr>
<tr>
<td>(2) Ackley function (AF)</td>
<td>$f_2(x) = -20e^{-0.02\sqrt{\sum_{i=1}^{D} x_i^2}} - e^{\sum_{i=1}^{D} \cos(2\pi x_i)} + 20 + e, \quad -35 \leq x_i \leq 35, \quad i = 1, 2, \ldots, D, \quad D = 2, \quad x^* = (0, 0, \ldots, 0), \quad f_2(x^*) = 0, \quad f_{2_{\text{max}}} = 1 \times 10^{-5}$</td>
</tr>
<tr>
<td>(3) Bohachevsky function (BF)</td>
<td>$f_3(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) - 0.4 \cos(4\pi x_2) + 0.7, \quad -100 \leq x_i \leq 100, \quad i = 1, 2, \quad x^* = (0, 0), \quad f_3(x^*) = 0, \quad f_{3_{\text{max}}} = 1 \times 10^{-9}$</td>
</tr>
<tr>
<td>(4) Drop-Wave function (DWF)</td>
<td>$f_4(x) = \frac{1 + \cos(12\sqrt{x_1^2 + x_2^2})}{0.5(\sqrt{x_1^2 + x_2^2})^2}, \quad -5.2 \leq x_i \leq 5.2, \quad i = 1, 2, \quad x^* = (0, 0), \quad f_4(x^*) = -1, \quad f_{4_{\text{max}}} = -0.9999$</td>
</tr>
<tr>
<td>(5) Egg-Crate function (ECF)</td>
<td>$f_5(x) = x_1^2 + x_2^2 + 25 \left[\sin^2(x_1) + \sin^2(x_2)\right], \quad -5 \leq x_i \leq 5, \quad i = 1, 2, \quad x^* = (0, 0), \quad f_5(x^*) = 0, \quad f_{5_{\text{max}}} = 1 \times 10^{-5}$</td>
</tr>
<tr>
<td>(6) Pen-Holder function (PHF)</td>
<td>$f_6(x) = -\exp\left[\cos(x_1) \cos(x_2)\left</td>
</tr>
<tr>
<td>(7) Rastrigrin function (RF)</td>
<td>$f_7(x) = 10D + \sum_{i=1}^{D} \left[x_i^2 - 10\cos(2\pi x_i)\right], \quad -5.12 \leq x_i \leq 5.12, \quad i = 1, 2, \ldots, D, \quad D = 2, \quad x^* = (0, 0, \ldots, 0), \quad f_7(x^*) = 0, \quad f_{7_{\text{max}}} = 1 \times 10^{-9}$</td>
</tr>
<tr>
<td>(8) Styblinski-Tang function (STF)</td>
<td>$f_8(x) = -\frac{1}{2} \sum_{i=1}^{D} \left(x_i^4 - 16x_i^2 + 5x_i\right), \quad -5 \leq x_i \leq 5, \quad i = 1, 2, \ldots, D, \quad D = 2, \quad x^* = (-2.9035, \ldots, -2.9035), \quad f_8(x^*) = -39.1660D, \quad f_{8_{\text{max}}} = -78.3319$</td>
</tr>
<tr>
<td>(9) Yang-2 function (Y2F)</td>
<td>$f_9(x) = \left(\sum_{i=1}^{D}</td>
</tr>
<tr>
<td>(10) Yang-4 function (Y4F)</td>
<td>$f_{10}(x) = \left[\sum_{i=1}^{D} \sin^2(x_i) - e^{-\sum_{i=1}^{D} x_i^2}\right] \cdot e^{\sum_{i=1}^{D} \sin^2(\sqrt{</td>
</tr>
</tbody>
</table>

Figure 5. (color online) 3D surface of SF function
The proposed LFICuS algorithm was coded by MATLAB version 2017b (License No.#4 0637337) run on Intel(R) Core(TM) i7-10510U CPU@1.80 GHz, 2.30 GHz, 16.0 GB RAM for function minimization tests against ten selected multimodal benchmark functions. Searching parameters of the proposed LFICuS algorithm are set from the preliminary studies with different ranges of parameters, i.e., step length $s$, index $\lambda$, number of state of AR and AN mechanisms activation $h$, number of initial neighborhood members $n$, number of search directions $N$, proceeding all selected benchmark functions. By varying $s = 0.0001, 0.001, 0.01$ and $0.1$, $\lambda = 0.0000001, 0.000001, 0.0001, 0.001, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9$ and $2.0$, $h = 2, 3, 4$ and $5$, $n = 25, 50, 100, 150, 200$ and $300$ and $N = 10, 25, 50, 75, 100, 125, 150$ and $200$, it was found that the best parameters for most benchmark functions are: $s = 0.001$ to $0.01$, $\lambda = 0.01$ to $1.3$, $h = 2$ to $3$, $n = 50$ to $100$ and $N = 50$ to $75$. For this performance evaluation test, $s = 0.01$, $\lambda = 0.3$, $h = 2$, $n = 100$ and $N = 50$ are set for all selected functions. Each search direction is terminated at 1000 iterations. 1000-trial runs are conducted for each algorithm. All algorithms will be terminated once two termination criteria (TC) are satisfied, i.e., (1) the function values are less than $f_{\text{max}}$ in Table 1 of each function or (2) the search meets the maximum search directions $N = 100$. The former criterion implies that the search is success, while the later means that the search is not success. For comparison with the ICuS, searching parameters of the ICuS are fairly set as the same values to those of LFICuS.

The obtained results are summarized in Table 2. Numeric data represented in Table 2 are in the form of $\text{AE} \pm \text{SD}(\text{SR})$, where the $\text{AE}$ is the average number (mean) of function evaluations, the $\text{SD}$ is the standard deviation and the $\text{SR}$ is the success rate. The $\text{AE}$ value implies the searching time consumed. The $\text{SD}$ value implies the robustness of the algorithm. Referring to Table 2, the proposed LFICuS algorithm provides greater $\text{SR}$ and less $\text{AE}$ and $\text{SD}$ values than the ICuS. This can be noticed that the proposed LFICuS algorithm performs much more efficient in function minimization than the ICuS algorithm. The convergent rates of the SF function proceeded by the ICuS and proposed LFICuS algorithms are depicted in Figures 6(a) and 6(b), respectively. Those of other functions are omitted because they have a similar form to that of SF function shown in Figure 6. Referring to Figure 6(a), it was found that the ICuS can reach the global solution with some research directions. From Figure 6(b), the proposed LFICuS can reach the global

### Table 2. Results of function minimization by ICuS and proposed LFICuS algorithm

<table>
<thead>
<tr>
<th>Functions</th>
<th>ICuS</th>
<th>Proposed LFICuS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) SF</td>
<td>$8.8161 \times 10^5 \pm 1.1378 \times 10^6$ (90.80%)</td>
<td>$5.0051 \times 10^4 \pm 0.00$ (100%)</td>
</tr>
<tr>
<td>(2) AF</td>
<td>$1.0944 \times 10^6 \pm 9.4378 \times 10^5$ (95.40%)</td>
<td>$5.3501 \times 10^4 \pm 1.3630 \times 10^4$ (100%)</td>
</tr>
<tr>
<td>(3) BF</td>
<td>$3.6153 \times 10^5 \pm 3.4431 \times 10^5$ (100%)</td>
<td>$5.0351 \times 10^4 \pm 3.8630 \times 10^3$ (100%)</td>
</tr>
<tr>
<td>(4) DWF</td>
<td>$4.9805 \times 10^5 \pm 4.1998 \times 10^5$ (100%)</td>
<td>$5.1151 \times 10^4 \pm 7.6710 \times 10^3$ (100%)</td>
</tr>
<tr>
<td>(5) ECF</td>
<td>$2.1490 \times 10^5 \pm 3.2485 \times 10^5$ (99.70%)</td>
<td>$5.0051 \times 10^4 \pm 0.00$ (100%)</td>
</tr>
<tr>
<td>(6) PHF</td>
<td>$6.9583 \times 10^4 \pm 1.1281 \times 10^4$ (100%)</td>
<td>$5.0051 \times 10^4 \pm 0.00$ (100%)</td>
</tr>
<tr>
<td>(7) RF</td>
<td>$1.9771 \times 10^6 \pm 1.2190 \times 10^6$ (71.90%)</td>
<td>$6.1951 \times 10^4 \pm 2.7370 \times 10^4$ (100%)</td>
</tr>
<tr>
<td>(8) STF</td>
<td>$7.3347 \times 10^5 \pm 8.7492 \times 10^5$ (96.60%)</td>
<td>$5.0401 \times 10^4 \pm 4.1710 \times 10^3$ (100%)</td>
</tr>
<tr>
<td>(9) Y2F</td>
<td>$2.5562 \times 10^5 \pm 2.3652 \times 10^5$ (100%)</td>
<td>$5.0351 \times 10^4 \pm 3.8630 \times 10^3$ (100%)</td>
</tr>
<tr>
<td>(10) Y4F</td>
<td>$7.8195 \times 10^5 \pm 6.6517 \times 10^5$ (97.60%)</td>
<td>$5.0301 \times 10^4 \pm 3.5284 \times 10^3$ (100%)</td>
</tr>
</tbody>
</table>
solution with all research directions. This shows the superior search performance of the proposed LFICuS to that of the ICuS for global optimization.

4. Problem Formulation of LFICuS-Based PID Controller Design. The active suspension system model and the design problem formulation of the PID controller by the LFICuS algorithm are described in this section as follows.

4.1. Active vehicle suspension system model. Suspension systems have been widely applied to modern vehicles with complex control algorithms for improving the ride comfort and road handling [18,19]. Normally, the two-degree-of-freedom (2DOF) suspension systems can be classified into three groups, i.e., passive, semi-active and active suspension systems. Among them, the active suspension system algorithms perform highest performance for vehicle ride comfort and road handling [18-20]. The active quarter car suspension model can be simplified by the 2DOF mass-spring-damper system as represented in Figure 7, where $m_b$ is sprung mass, $m_u$ is unsprung mass, $k_s$ is spring constant,
$b_s$ is damper’s friction coefficient, $f_s$ is actuator force, $k_t$ is tire stiffness constant, $x_b$ is sprung mass displacement, $x_w$ is unsprung mass displacement and $r$ is vertical displacements of road profile. By applying the Newton’s laws, the equations of motion for the active quarter car suspension system can be formulated as expressed in (5) and (6) for sprung and unsprung masses, respectively.

$$m_b \ddot{x}_b = -k_s(x_b - x_w) - b_s(\dot{x}_b - \dot{x}_w) + f_s$$

$$m_w \ddot{x}_w = k_s(x_b - x_w) + b_s(\dot{x}_b - \dot{x}_w) - k_t(x_w - r) - f_s$$

From (5) and (6), the mathematical model of the active quarter car suspension system can be performed in the state-variable model as expressed in (7), where $x_1 = x_b$, $x_2 = \dot{x}_b$, $x_3 = x_w$ and $x_4 = \dot{x}_w$ are the state variables, $u = [f_s \ r]^T$ are the input variables and $y = x_1 = x_b$ is the output variable. The parameters $m_b = 300$ kg, $m_w = 60$ kg, $k_s = 16,000$ N/m, $b_s = 1,000$ N-s/m and $k_t = 190,000$ N/m, are utilized from [21]. In this work, these parameters are assumed to be fixed. The parameter variations can occur as the model uncertainty due to aging and environmental effects. System sensitivity due to such the model uncertainty can be analyzed by the robust control approach as appeared in [22]. Thus, the state-variable model of the considered active quarter car suspension system becomes the model in (8).
4.2. Problem formulation of LFICuS-based PID controller design optimization. The LFICuS-based PID controller design optimization framework for the active suspension system can be represented by the block diagram shown in Figure 8. The design optimization framework is adapted from the controller design optimization [22-27]. From Figure 8, $G_c(s)$ is the PID controller model stated in (9), where $K_p$ is the proportional gain, $K_i$ is the integral gain and $K_d$ is the derivative gain. Based on the modern optimization, the objective function \( f(\cdot) \) is set as the sum-squared error (SSE) between the input reference $r_{ref}$ and output $y = x_b$ as stated in (10).

\[
G_c(s) = K_p + \frac{K_i}{s} + K_ds
\]

\[
\text{Minimize } f(K_p, K_i, K_d) = \sum_{i=1}^{N} [r_{ref,i} - y_i]^2
\]

Subject to \[
M_p \leq M_{p,\text{max}}, \quad t_{re} \leq t_{re,\text{max}}, \quad e_{ss} \leq e_{ss,\text{max}}, \quad K_{p,\text{min}} \leq K_p \leq K_{p,\text{max}}, \quad K_{i,\text{min}} \leq K_i \leq K_{i,\text{max}}, \quad K_{d,\text{min}} \leq K_d \leq K_{d,\text{max}}
\]

The objective function $f(\cdot)$ will be fed to LFICuS algorithm to be minimized by searching for the optimal values of $K_p$, $K_i$ and $K_d$ of the PID controller within their search spaces and the design specification set as the inequality constrained functions as expressed in (11), where $K_{p,\text{min}}$ and $K_{p,\text{max}}$ are search bounds of $K_p$, $K_{i,\text{min}}$ and $K_{i,\text{max}}$ are search bounds of $K_i$, $K_{d,\text{min}}$ and $K_{d,\text{max}}$ are search bounds of $K_d$, $M_p$ is the maximum percent overshoot, $M_{p,\text{max}}$ is maximum allowance of $M_p$, $t_{re}$ is recovering time, $t_{re,\text{max}}$ is maximum allowance of $t_{re}$, $e_{ss}$ is steady-state error and $e_{ss,\text{max}}$ is maximum allowance of $e_{ss}$.

5. Results and Discussions. In order to design the PID controller for the active suspension system, the proposed LFICuS algorithm was coded by MATLAB version 2017b (License No.#40637337) run on Intel(R) Core(TM) i7-10510U CPU@1.80 GHz, 2.30 GHz, 16.0 GB RAM. Referring to Figure 8, the active suspension system with the PID controller is coded by the Simulink of MATLAB, while the LFICuS algorithm is coded by the script m-file of MATLAB. The search parameters of the LFICuS algorithm are set from the preliminary study, i.e., $N = 50$, $n = 20$ and $R = 100\%$ of search spaces. Maximum iteration (Max_Iter) $= 100$ is set for terminating each search direction. For the AR and AN mechanisms, Stage-I: at the 25th iteration, $R = 10\%$ of search spaces and $n = 10$, Stage-II: at the 50th iteration, $R = 1.0\%$ of search spaces and $n = 5$. 50-trial runs are conducted to find the optimal PID controller parameters ($K_p$, $K_i$ and $K_d$). The input
reference \( r_{ref} \) in Figure 8 and in (10) is set as zero. Thus, the objective function \( f(\cdot) \) in (10) can be stated in (12). The inequality constrained functions and search bounds in (11) are defined from the preliminary study as expressed in (13).

A single-bumpy road is utilized as the road profile \( r \) as stated in (14) [28]. The road disturbance \( \delta = 0.02 \) is set to achieve a bump height of 4 cm. With 50-trial runs, the LFICuS can successfully provide the optimal PID controller for the active suspension system as stated in (15). The convergent rates over 50-trial runs are plotted in Figure 9.

\[
\text{Minimize } f(K_p, K_i, K_d) = \sum_{i=1}^{N} [y_i]^2 = \sum_{i=1}^{N} [x_{b,i}]^2 
\]

\[
\text{Subject to } \begin{cases} 
M_p \leq 30.00\% , \\
t_{re} \leq 2.50 \text{ sec} , \\
e_{ss} \leq 0.01\% , \\
0 \leq K_p \leq 10.00 , \\
0 \leq K_i \leq 10.00 , \\
0 \leq K_d \leq 5.00 
\end{cases} 
\]

\[
r = \begin{cases} 
\delta(1 - \cos \omega t) , & 0 \leq t \leq 0.25 \text{ sec} \\
0 , & \text{otherwise} 
\end{cases} 
\]

\[
G_c(s) = 9.9967 + \frac{1.5762}{s} + 4.9985s 
\]

The simulation results against a single bumpy road profile are shown in Figure 10 by comparison between the passive suspension system (without PID controller and \( f_s = 0 \)) and the active suspension system with the PID controller designed by the LFICuS. From Figure 10, the passive suspension system yields slower response and greater oscillation of the sprung mass displacement (\( M_p = 70.52\% , t_{re} = 2.54 \text{ sec and } e_{ss} = 0.00\% \)). In contrast, the active suspension system provides faster response and smaller oscillation of the sprung mass displacement (\( M_p = 28.01\% , t_{re} = 0.72 \text{ sec and } e_{ss} = 0.00\% \)). This can be noticed
that the considered active suspension system controlled by the PID controller designed by the LFICuS yields very satisfactory response superior to the passive suspension system, significantly.

6. Conclusions. The LFICuS algorithm and its application to the PID controller design for the active vehicle suspension system have been presented in this paper. The LFICuS has been developed from the ICuS by using the random number with the Lévy-flight distribution. The AR and AN mechanisms have been also conducted in the proposed LFICuS. By testing against ten selected standard multimodal benchmark functions, the proposed LFICuS performed much more efficient than the ICuS. For the design application based on the modern optimization, the LFICuS could provide the optimal PID controller for the active suspension system. By comparison with the passive suspension system, it was found that the PID controller designed by the LFICuS for the active vehicle suspension system could provide very satisfactory response superior to the passive suspension system.
with faster response and smaller oscillation. For future research, applications of the proposed LFICuS will be extended to other control design optimization problems including the PIDA, FOPID and FOPIDA controllers for some specific real-world systems.

REFERENCES


