Simultaneous Model Order and Parameter Estimation (SMOPE) based on Random Asynchronous Particle Swarm Optimization

Nor Azlina Ab. Aziz¹, Nor Hidayati Abd Aziz¹, Tasiransurini Ab Rahman², Norrima Mokhtar³, and Marizan Mubin³

¹Faculty of Engineering and Technology, Multimedia University, Melaka, Malaysia.
²Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Johor, Malaysia.
³Faculty of Engineering, University of Malaya, Kuala Lumpur, Malaysia.

ABSTRACT –Simultaneous model order and parameter estimation (SMOPE) is a metaheuristic based system identification method. SMOPE was introduced using particle swarm optimization (PSO). There are several iteration strategies for PSO. The original work on SMOPE is based on synchronous PSO (S-PSO). However, in some works PSO using other iteration strategy is found to give better results. In this work, based on six system identification problems random asynchronous (RA-PSO) based SMOPE is found to have slight advantage over S-PSO.

Introduction

Simultaneous model order and parameter estimation (SMOPE) was proposed for solving autoregressive exogenous system identification problem effectively using metaheuristics algorithms [1-2]. The method enabled a system's order and parameters values to be searched simultaneously. This is possible through the way the problem is encoded in the search agents. Even though SMOPE was introduced based on particle swarm optimization (PSO) [1-2], it can easily be adapted to suit other metaheuristic algorithm such as, gravitational search algorithm (GSA) [3-4].

The PSO is a population-based optimization algorithm. The search agents of PSO, known as particles mimics how living organism such as birds and fishes look for food by exploring the search area using their own experience and information from neighborhood as guidance. The search in PSO is done iteratively. PSO’s iteration strategy can be classified as synchronous (S-PSO) and asynchronous (APSO) update [5]. S-PSO is more popular approach than A-PSO, where in S-PSO the movement of the whole particles in the swarm is done at once, after their performance is evaluated. In A-PSO a particle moves as soon as its own performance is evaluated, without the need to wait for others to complete their evaluation. The direction of the movement in A-PSO is made based on whatever information available. This is a more accurate replication of nature.

Random asynchronous PSO (RA-PSO) was introduced in [6]. In the original APSO the particles are evaluated and move according to the particle number. However, in RA-PSO the particle to be evaluated and move is chosen randomly, hence, in an iteration a particle can move more than once or none at all. It is found that RA-PSO is better than A-PSO.

In this work the implementation of SMOPE using RA-PSO is studied and compared with SMOPE based on S-PSO. In several works, implementation of PSO with a particular iteration strategy is found to give a better result compare to other strategy. For example, Wu and Gao had reported that their adaptive inertia weight PSO implemented using asynchronous update has a better performance than the same approach implemented using synchronous update [7]. In [8], A-PSO with discrete crossover is found to perform better than S-PSO with the crossover operator.

However, Engelbrecht in his work concluded that there is no definite winner of S-PSO vs A-PSO but rather it is a function dependent option [5]. The same observation is made in [9].

Therefore, in this work the performance of RA-PSO based SMOPE is compared with the S-PSO based SMOPE. Six ARX system identification problems are used. The results show that RA-PSO on average has a slightly better performance.

Autoregressive Exogenous Model (ARX)

System identification is a task of finding an accurate mathematical model of a control system based on the available input and output data [10]. In [11], the ARX model was introduced by Ljung among many other models for system identification.
SMOPE is implemented using PSO based algorithm which has gained popularity due to its simplicity and low computational cost. It has been successfully adopted in various fields, such as robotics [12], power distribution planning [13], and financial planning [14].

Each of the agents in SMOPE represents system order and parameters values. Assuming maximum system order under consideration is \(D\), the agents dimension should be \(2D+1\). The first dimension of each agent’s represents the system order; \(n\), while dimension 2 to \(D+1\) represents the possible values of poles parameters, \(a_1, a_2, ..., a_m\), and dimension \(D + 2\) to \(2D + 1\) are reserved for the zeros parameters, \(b_1, b_2, ..., b_m\). Both \(m\) and \(n\) can be lesser than \(D\). If \(m < D\), then only the values in dimension 2 to \(m_a + 1\) are used, while the values in dimension \(m_a + 2\) to \(D + 1\) are ignored. Similarly, if \(m_b < D\), then only the values in dimension \(D + 2\) to \(D + m_b + 1\) are used, while the values in dimension \(D + m_b + 2\) to \(2D + 1\) are ignored. In this work the maximum order considered is 9 with \(m_a \leq m_b\).

Particle Swarm Optimization

Particle swarm optimization (PSO) is a population based algorithm which has gained popularity due to its simplicity and low computational cost. It has been successfully adopted in various fields, such as robotics [12], power distribution planning [13], and financial planning [14].

Each of the particles in PSO acts as the search agents. The particles have velocity, and position . The search for optimal solution is conducted in PSO by iteratively evaluating and updating particles performance, velocity and position. The velocity and position are updated according to equation (5) and (6), accordingly. The particles’ search direction is influenced by the previous search, their own best performance, \(p\text{Best}\), and neighbourhood best, \(g\text{Best}\). The performance of the particles’ can be measured using equation (7). In the equation, \(\hat{y}_{\text{estimation}}\) is the output signal based on the mathematical model found by a particle, whereas \(y\) is the actual data and \(\bar{y}\) is its mean value.

In this paper, SMOPE is implemented using PSO of two different update strategies, synchronous PSO (S-PSO) and random asynchronous PSO (RA-PSO).
\begin{align*}
v_i(t) &= \omega v_i(t - 1) + c_1 r_1 (p_{Best} - x_i(t - 1)) + c_2 r_2 (g_{Best} - x_i(t - 1)) \\
x_i(t) &= v_i(t - 1) + x_i(t - 1)
\end{align*}

\text{best fit} = 100 \left[ 1 - \frac{\text{norm}(\chi_{estimation} - \chi)}{\text{norm}(\chi - \chi)} \right] \%

Synchronous update is the more famous iteration strategy for PSO. In S-PSO, the whole population is updated first before their velocities and positions are updated. Hence, the particles have overview of the whole swarm’s performance before the next move is made. The pseudocode for S-PSO is shown in Figure 2. There are two loops per iteration for S-PSO. In the first loop the performance of the whole population is evaluated, whereas the particles velocities and positions are updated in the second loop.

Random asynchronous update is a new iteration strategy for PSO [5]. In RA-PSO, a particle is chosen randomly to be evaluated. Immediately after this particle is evaluated, its velocity and position are updated using the available information. There is no restriction on repetition, hence a particle can be chosen more than once or none at all in an iteration. The chosen particles in RA-PSO are updated based on various neighborhood information. The pseudocode for RA-PSO is shown in Figure 3. There is only one loop per iteration in RA-PSO. In the loop, first a particle to be evaluated is randomly chosen, then its performance is evaluated, followed by its velocity and position update.

**Experiments**

Six system identification problems found in database for the identification of system (DaISy) were used. Four of the systems chosen are mechanical systems, which are ball-beam, hairdryer, wing flutter and robot arm. The data for ball-beam, hairdryer and robot arm systems are obtained from laboratory works while the wing flutter data is obtained from industry. A thermic system namely SISO heating system is also chosen for the experiment. The heating system’s output is measured using thermocouple taken from the back of a steel plate. The last experiment is using data from process industry, which is a liquid-saturated steam heat exchanger system.

The first half of the data from each of the systems, is used for training purpose, which is to select the best order and parameters values using SMOPE, while the other half is used for testing.

For example, as shown in Figure 4. The first half of the data for the hair dryer system (in the box) is used for training while the remaining is used to test the quality of the solution found by SMOPE.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Variable in ARX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Order, n</td>
</tr>
<tr>
<td>2</td>
<td>a_1</td>
</tr>
<tr>
<td>3</td>
<td>a_2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>D+1</td>
<td>a_D</td>
</tr>
<tr>
<td>D+2</td>
<td>b_1</td>
</tr>
<tr>
<td>D+3</td>
<td>b_2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>2D+1</td>
<td>b_2D</td>
</tr>
</tbody>
</table>

**Table 1. Agent’s encoding.**

![Figure 2. S-PSO’s pseudocode.](image2)

![Figure 3. RA-PSO’s pseudocode.](image3)

The SMOPE method is implemented using both S-PSO and RA-PSO here. The algorithms are using population of 100 particles which are randomly initialized. The algorithms are repeated until either 100% training fitness is achieved or the iteration count exceeds 2000. Each of the experiment is repeated 50 times and the results found are averaged.

**Results and Discussions**

The results obtained from the experiment are tabulated in Figure 5 and Figure 6 shows the average training fitness in every iteration for each system.
Figure 4. Input and output data of the hair dryer system.

Figure 5. Performance of S-PSO based SMOPE vs RA-PSO based SMOPE.

On average RA-PSO has a slight advantage over the original implementation which is based on S-PSO. Out of the six systems used, RA-PSO performs better in four systems, which are the heating system, exchanger system, hair dryer system and wing flutter system. RA-PSO has a better performance for these systems in training phase as well as in the testing stage. However, the differences between the two algorithms are marginal.

The marginal difference can be seen in Figure 5. It can be seen that in all iteration the fitness of S-PSO based SMOPE and RA-PSO based SMOPE is close to each other. The mathematical models for each system found by both algorithms are presented in Figure 7.

Both algorithms found their own model with their own parameters values and system order.

Conclusion

SMOPE is a metaheuristic based system identification method. The method is able to determine the system order and the parameters simultaneously. This work investigates the difference between S-PSO based and RA-PSO based SMOPE. The implementation of SMOPE using RA-PSO is found to have a slight advantage over its implementation using S-PSO.
Figure 6. Convergence curves.

<table>
<thead>
<tr>
<th>System</th>
<th>S-PSO based SMOPE</th>
<th>RA-PSO based SMOPE/PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Heating System</strong></td>
<td>$G_h(Z) = \frac{1.1728E-01Z^{-1} + 1.2558E-01Z^{-2}}{1-1.5447E+00Z^{-1} + 5.1517E-02Z^{-2} + 4.0888E-02Z^{-3}}$</td>
<td>$G_s(Z) = \frac{2.0558E-01Z^{-1} + 4.2518E-02Z^{-2}}{1-1.6480E+00Z^{-1} + 4.2518E-02Z^{-2} + 4.7193E-02Z^{-3}}$</td>
</tr>
<tr>
<td><strong>Hair Dryer System</strong></td>
<td>$G_{hd}(Z) = \frac{1.9358E-03Z^{-1} + 3.4780E-03Z^{-2}}{1+ 6.3246E-02Z^{-1} + 2.4032E-02Z^{-2} + 4.24258E-01Z^{-3} + 5.6596E-02Z^{-4}}$</td>
<td>$G_{hs}(Z) = \frac{-4.321E-04Z^{-1} + 5.0368E-03Z^{-2} + 5.303E-02Z^{-3}}{1+ 1.1112E+00Z^{-1} + 7.0820E-02Z^{-2} + 2.3728E-01Z^{-3} + 5.1538E-02Z^{-4}}$</td>
</tr>
<tr>
<td><strong>Robot Arm System</strong></td>
<td>$G_r(Z) = \frac{-6.3515E-02Z^{-1} + 7.3661E-02Z^{-2}}{1-2.8196E+00Z^{-1} + 6.3586E+00Z^{-2} - 2.3624E+02Z^{-3} + 6.7568E-01Z^{-4}}$</td>
<td>$G_{rs}(Z) = \frac{-5.8551E-01Z^{-1} + 1.3019E+00Z^{-2} - 1.2507E+00Z^{-3}}{1+ 6.2992E-01Z^{-1} + 2.3527E-02Z^{-2} - 5.496E-02Z^{-3} + 1.1418E-01Z^{-4}}$</td>
</tr>
<tr>
<td><strong>Wing Flutter System</strong></td>
<td>$G_w(Z) = \frac{-3.3964E-02Z^{-1}}{1-2.6113E+00Z^{-1} + 2.4575E+00Z^{-2} - 8.3518E-01Z^{-3}}$</td>
<td>$G_{ws}(Z) = \frac{-4.6148E-02Z^{-1}}{1-2.2871E+00Z^{-1} + 2.3704E+00Z^{-2} + 3.3797E-01Z^{-3} - 6.9798E-01Z^{-4} + 5.0341E-02Z^{-5} + 4.1748E-02Z^{-6}}$</td>
</tr>
<tr>
<td><strong>Ball Beam System</strong></td>
<td>$G_{bh}(Z) = \frac{2.7357E-01Z^{-1} + 1.5266E-01Z^{-2}}{1-9.1088E-02Z^{-1} - 2.8977E-02Z^{-2} + 4.4946E-02Z^{-3} - 1.0381E-02Z^{-4} + 2.8406E-02Z^{-5}}$</td>
<td>$G_{bs}(Z) = \frac{-1.9295E-01Z^{-1} + 1.3219E-01Z^{-2} + 1.2011E-01Z^{-3} + 9.303E-02Z^{-4}}{1-9.9073E-01Z^{-1} + 2.1849E-01Z^{-2} + 1.0525E-01Z^{-3} - 1.3991E-01Z^{-4} + 5.4615E-02Z^{-5} - 1.2018E-02Z^{-6} + 1.9851E-01Z^{-7}}$</td>
</tr>
<tr>
<td><strong>Exchanger System</strong></td>
<td>$G_e(Z) = \frac{2.2076E-02Z^{-1}}{1-1.2569E+00Z^{-1} + 2.5761E-01Z^{-2}}$</td>
<td>$G_{es}(Z) = \frac{2.3131E-01Z^{-1}}{1-1.2661E+00Z^{-1} + 2.6692E01Z^{-2}}$</td>
</tr>
</tbody>
</table>

*CORRESPONDING AUTHOR | Nor Azlina Ab. Aziz | azlina.aziz@mmu.edu.my

Figure 7. Mathematical models.
References


