An improved particle swarm optimization using long short-term memory model for positioning control of a coplanar XXY stage

Yi-Cheng Huang¹ and Ming-You Ma²

Abstract
This paper presents XXY stage alignment with an image feedback system consisting of two charge-coupled devices (CCDs) and a proportional–integral–derivative (PID) image servo system tuned by particle swarm optimization (PSO). The initial stop values for the PSO algorithm often cause problems in calculation. A long short-term memory (LSTM) deep learning model can identify long-term dependences and sequential model data. Using LSTM to improve the PSO algorithm for searching best fitness value. LSTM predicts the fitness value of PSO, eliminating the need to preassess fitness, value, and uses the predicted fitness value to adjust the inertia weights of PSO adaptively. This allows the PSO search to be terminated in an early stage and reduces the time required for the search. Proposed method was applied to a visual servo system consisting of two CCD cameras and a personal computer–based PID controller for XXY stage motion. The experimental results indicate that LSTM can reduce the time required for PSO fitness search for controlling XXY stage motion under different conditions successfully. Through the training of the LSTM model, the stage positioning error and time of finding optimal control parameters for a coplanar XXY stage can reduced significantly for in-line inspection processes.

Keywords
Automated optical inspection, long short-term memory (LSTM), particle swarm optimization (PSO), proportional–integral–derivative (PID) controller, XXY stage

Introduction
With the rapid development of industrial science and technology, positioning systems have been widely applied in the field of precision manufacturing.¹ With the development of computers, communications, and consumer electronics, high-precision alignment systems have become increasingly crucial to component manufacturing processes, such as the lamination and electrical testing of touch panels – which involve resistance, capacitance, and linearity parameters – electrical testing for open circuits or shorts, the testing of printed circuit boards (PCB), and the manufacture of light-emitting diodes from wafers.² When wafer masks are applied, a precision position platform is used to correctly position and orient the panel. Accordingly, accuracy and quality are crucial to the manufacturing process.¹ Many alignment systems now use coplanar stages for high-precision alignment because such stages are faster than stacked ones. This is because coplanar stages have a lower center of gravity and less inertia.³ Coplanar stages also have higher angular resolution and cost less than direct-drive motor stages. According to kinematic equations, the motors on a coplanar stage can be operated simultaneously to move the stage to the target position faster than a stacked stage can move.² Computer vision has been widely used in image-based technology.⁴–⁶ Vision systems function like the human eye, which can automatically detect and recognize images. Therefore, this study’s proposed platform control uses computer vision for image detection and the processing of image signals. Image-processing technology
can improve the accuracy of stage positioning and the effectiveness of stage control methods.\(^1\)

Kim used two CCD cameras, an \(XY\theta\) stage, and a feed-forward neural network controller for alignment in wafer manufacturing. The positioning precision was approximately 1 pixel (approximately 12 \(\mu\)m). The alignment performance was approximately 6\% superior to that achieved with conventional algorithms. Kim et al.\(^7\) employed a cross mark for alignment. Cross marks are commonly used in many visual alignment systems because they have clear features and can be easily identified. Huang and Lin\(^8\) used a CCD camera, a stacked \(XY\) stage, and cross marks for alignment. A fuzzy logic controller was used to control the motion of the alignment stage, and the imaging software eVision was used for image processing. The steady-state error was less than 1 \(\mu\)m.

Alignment using stacked \(XY\theta\) stages usually causes cumulative errors, such as parallelism errors, orthogonal errors between axes, and flatness errors. Tarng and Lin\(^9\) used Coplanar \(XY\theta\) stages have been used to avoid such errors, and they have exhibited superior positioning accuracy to that of stacked stages. Lee and Liu\(^14\) presented an image alignment system with visual servo control that used a specially designed coplanar \(XY\) stage, and each alignment motion was completed in less than 1 s, with a precision of \(\pm 1 \mu\)m and \(\pm 5\)\(^\circ\). Lee et al.\(^12\) proposed a system consisting of four CCDs, two alignment objects, and a specially designed parallel stage with three degrees of freedom. The alignment displacements for the the \(X\)-axis, the \(Y\)-axis, and the angle of the rotation, were analyzed through forward kinematic analysis. The linear and angular positioning repeatability of the \(XY\) stage was better than 1 \(\mu\)m and 20\(^\circ\). The alignment error was approximately \(\pm 1\) pixel. Yang et al.\(^13\) proposed automatic locating and image servo alignment for the touch panels of an automatic laminating machine; the design employs four CCD cameras integrated with a coplanar \(XY\) stage. Lin et al.\(^14\) designed an optical alignment system using a coplanar \(XY\) stage integrated with dual CCD cameras and used a neural network–based method to improve the accuracy of their image servo system. However, the time required for image pattern matching and image processing should be reduced for real-world applications. Lin et al.\(^15\) used a microcontroller for an image-based \(XY\) positioning platform. By incorporating image recognition technology, they could detect the position error between \(XY\) and a detected object. Such error information can be used for positioning control.\(^1\)

Conventional proportional–integral–derivative (PID) controllers are widely applied in control systems for their practicality and robustness. PID gains must be properly tuned to ensure superior dynamic performance, security, and the sustainability of equipment and plants.\(^16\) Several approaches have been used to determine the parameters for PID controllers, including the Ziegler–Nichols tuning formula (for time response), frequency domain shaping, genetic algorithms (GAs),\(^17\) and Moth Fly Optimization (MFO).\(^18\) The performance of a PID controller designed using these techniques is satisfactory but may be suboptimal because any constraints on settling time, overshoot, or undershoot and disturbances are system dependent. Kennedy and Eberhart\(^19\) introduced particle swarm optimization (PSO) in 1995. PSO simulates simplified social behavior and can efficiently solve continuous nonlinear optimization problems. As a swarm intelligence technique, PSO is one of the population-based optimization algorithms. PSO is attractive for several reasons. First, the PSO algorithm requires only a few lines of computer code. Second, its search technique is simple and uses the fitness value of the objective function instead of gradient information. Third, it is computationally inexpensive because its memory and processing speed requirements are low. Fourth, unlike conventional deterministic methods, PSO does not require strong assumptions about linearity, differentiability, convexity, separability, and can work under constraints to solve problems efficiently. Finally, its solutions do not depend on the initial states of particles, which represents an advantage for design optimization problems in engineering.\(^20\)–\(^22\)

Therefore, accurate real-time historical mapping of the relationship between the parameter fields of the controller and the integral of the absolute magnitude of error (IAE) is necessary. One deep learning model, the long short-term memory (LSTM) network, is highly proficient in time series forecasting in various fields because it can dynamically learn new information while retaining a consistent memory of historical information. Sagheer and Kotb\(^23\) performed time series forecasting for petroleum production by using recurrent LSTM networks optimized by a GA. The LSTM model outperformed other standard approaches. Zhang et al.\(^24\) employed an LSTM network to predict the remaining useful life of lithium-ion batteries. Zhang et al.\(^25\) developed an LSTM model to predict water table depth in agricultural areas and evaluated the abilities of the proposed model. The LSTM model achieved satisfactory performance. Qin et al.\(^26\) employed an LSTM model to predict the remaining life of gears. LSTM networks can capture both short-term correlation and long-term dependence. Gao et al.\(^27\) employed the thermal error model by using LSTM networks which was optimized by PSO, the PSO-LSTM model is established to precisely predict the thermal error of ball screws, and then provide a foundation for thermal error compensation. Liu et al.\(^28\) performed a railway freight volume forecasting model by using LSTM networks optimized by PSO, and then the PSO-LSTM model has lower prediction error and higher prediction accuracy than the traditional LSTM and model GA-LSTM. However, few studies have investigated using LSTM networks to predict the IAE of machine controllers, especially for \(XY\) stages.

This study attempted to overcome three main challenges of image servo mask alignment system using an
XYY stage through intelligent machine tuning. The first challenge is solving the problem for the lack of linear scale for displacement measurement and the nonlinear positioning error due to mechanical accuracy of the nonlinear motion of the stage. Although coplanar XYY stages have some advantages, such as lower cumulative error and quicker responses, than stacked XXY stages, motion planning for XYY stages is difficult because the individual movement along the X1-axis, X2-axis, and Y-axis is coupled with the others, and their kinematic relationship is nonlinear. Therefore, this study developed an image-based method to solve this problem. The second challenge is that the lack precision caused by the semi-closed loop control of the servo motor. This study proposes a PID control for visual servo control system based on PSO tuning controller parameters. The third challenge is predicting the fast convergence fitness value for PSO algorithm. Since the stop bettering fitness value for stop PSO computation must be obtained experimentally, but this may cause time consuming; therefore, this study uses the characteristics of LSTM to predict the fitness value for PSO and stop the PSO algorithm after fewer iterations. The fourth future challenge is the assembly of a precision stage demands time for fine tune of individual control parameters. Knowing the machine positioning accuracy is a key priority. Since the remaining useful life of motion stage could decay after long time operation. As for prognostic health maintenance issue, PSO-LSTM model can be deployed in shop floor system.

This paper is organized as follows: in Section 2, experimental setup including hardware of XYY stage, vision system and controller; in section 3, fundamental of PSO; in section 4, LSTM network; in section 5, the XYY stage motion with PSO–LSTM model; in section 6, experimental analysis; and in section 7, contributions are concluded.

**XYY system principle and structure**

**XYY stage hardware**

The XYY stage is characterized by three motors on the same plane with the merits of low gravity center. In other words, the moving speed of XYY stage can be faster than the traditional stacked XXY stage. It is small and light. Thus, the main advantage of XYY stage is being smaller cumulative error of stage composition than the traditional stacked stage. Therefore, the coplanar XYY stage are very popular for the applications of precision motion, such as AOI and lithography.

As shown in Figure 1, the system hardware consists of the upper mask chip device, which carries the upper cross mask for CCD’s imaging, the lower coplanar XYY stage (XYY-25-7, CHUAN YAN Ltd., Changhua, Taiwan), which carries the lower part to align the upper device with the image servo control, and two CCD camera lens, which are mounted on the top of the system as the image servo sensors for positioning. A motion card (PCI-8143, ADLINK TECHNOLOGY INC, Taiwan) control the XYY stage, and ADLINK’s Domino Alpha2 image card was used for XYY stage image position feedback. Picture of the XYY experimental stage was shown in Figure 2.

Traditional XXY stages use the stacked design, which consists of an X-axis translation stage, a Y-axis translation stage, and a θ-axis rotational stage. The controller design for traditional stacked XXY stage is simple because the movement of each axis is independent. However, the XXY stage produces cumulative flatness errors due to the stacked assembly and the large size of the stage. Therefore, coplanar XXY stage was developed because the coplanar design produces lower cumulative error and can move faster than the traditional XXY stage. Figure 3 displays the structure of the coplanar XXY stage, which is driven by three servo motors, the X1-axis motor, X2-axis motor, and Y-axis motor. The working stage is supported by four sub stages; each substage consists of X translation, Y translation, and θ rotation stages. Therefore, the motion of the XYY stage has three degrees of freedom: the
translation along the $X$-axis and $Y$-axis and rotation around the $\theta$-axis.\textsuperscript{11,15}

The $XXY$ stage can move up to $\pm 5\text{mm}$, and the maximal angle is $2^\circ$. When the motors of the $X_1$- and $X_2$-axes rotate clockwise or counter-clockwise and the motor of the $Y$-axis is static, the stage can move along the $X$-axis. By contrast, the motor of the $Y$-axis is used to control the movement of the stage along the $Y$-axis.

The $XXY$ stage has motion in three dimensions, $X$, $Y$, and $\theta$ (Figure 4). If the linear displacement of the stage is represented as $\mathbf{s} = \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix}$ (mm) and the angular displacement of the stage is $\delta_\theta$ (radians), the basic linearized relation between the stage displacement and the motor displacement is

$$
\begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} = \frac{R_m}{l_p} \begin{bmatrix} 1 & 0 & k_a \\ 1 & 0 & k_b \\ 0 & 1 & k_c \end{bmatrix} \begin{bmatrix} \delta_x \\ \delta_y \\ \delta_\theta \end{bmatrix},
$$

(1)

where $R_m$ is the motor resolution (pulses per revolution); $l_p$ is the lead of the ball screw (mm/revolution); and $k_a$, $k_b$, and $k_c$ are the parameters for angle rotation. The parameters $k_a$, $k_b$, and $k_c$ are related to the distance between the center of the stage and the center of the $XXX$ stage in the $\{m_1\}$, $\{m_2\}$, and $\{m_3\}$ coordinate systems. To achieve high-precision motion control, the kinematic formula must consider the deviation of the center of the $XXX$ stage during motion.\textsuperscript{11}

**Vision for $XXX$ stage**

The purpose of the proposed image processing method is to determine the position of the alignment symbol through the center of gravity method.\textsuperscript{14,32} Then, the coordinates of the $XXX$ stage are calculated on the basis of the coordinates of the mask target. Locating the target mark and determining the corresponding relationship between stage movements is crucial tasks. Our method employs the image preprocessing tools.\textsuperscript{6,7} Binarization and morphology are used to simplify the image and reduce the noise. The preprocessing reduces the residual noise in the image, allowing the features to be separated. Next, the feature coordinates are detected through a simple center of gravity method, and the position of the feature center is acquired.

The center of gravity method is used to obtain the target position in the image coordinate system. Two gray pictures are acquired from the two CCDs. The noise in the images is processed using a filter, and the binary threshold of the grayscale histograms is used to separate the two targets. Expansion and subtraction are used to remove the remaining noise, allowing the optimal image to be identified through a morphological process. Then, feature targets are identified through the findContours function of OpenCV. The center of gravity method is thus used to obtain the coordinates of the center of the image for the positioning mark. Figure 5 displays a flowchart of the image identification procedure, Figure 6 displays the cross mask position results by center of gravity method.

**Controller for $XXX$ stage**

The time domain of the PID controller for the $XXX$ stage is given as follows:

$$
u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} + K_{ff} r(t),$$

(2)

in which $r(t)$, $e(t)$, $u(t)$, $K_p$, $K_i$, $K_d$, and $K_{ff}$ represent input command, system error, control variable, proportional gain, integral gain, derivative gain, and velocity feed-forward gain, respectively. Figure 7 presents the
control architecture of PCI-8143 motion card. Each coefficient of the PID controller affects the system’s output response. Therefore, selecting the appropriate gains is crucial for the practical application of this controller.30

Particle swarm optimization

The PSO algorithm is based on a flock of birds and simulates their behavior in a simplified social system. The individual elements in PSO evolve through cooperation and competition through generations instead of through the application of a genetic operator. Each individual is referred to as a “particle” and treated as a point in a D-dimensional space, representing a potential solution to a problem. Each particle adjusts its velocity, position, and best previous position on the basis of its own experience and the experiences of its neighbors.17,18,20 Inertia weight is used to balance the local and global search functions. Because each particle remembers its worst experience, it can explore the search space effectively to identify the most promising solution region. The PSO algorithm can be presented as follows:

\[
V_{id}(t) = w(t) \times V_{id}(t - 1) + c_1 \times \text{Rand}(t) \times (P_{id} - X_{id}) + c_2 \times \text{Rand}(t) \times (P_{gd} - X_{id}),
\]

(3)

\[
X_{id}(t) = X_{id}(t - 1) + V_{id}(t),
\]

(4)

where \(V_{id}(t)\) is the current velocity of \(i\)th particle \(i = 1, \ldots, n\), where \(n\) is the population size; \(\text{Rand}(t)\) represents a uniform random number between 0 and 1; subscript \(d\) is the dimension of the particle; \(P_{id}\) is the best previous position of the \(i\)th particle; \(P_{gd}\) is the best previous position among all particles in the swarm; and \(X_{id}\) is the current position of the \(i\)th particle. The constants \(c_1\) and \(c_2\) represent the weighting of the stochastic acceleration terms pulling each particle toward the \(P_{id}\) and \(P_{gd}\) positions. Low values allow particles to roam far from the target regions before being tugged back. High values result in abrupt movement toward or past the target regions. Inertia weight \(w(t)\) is calculated as follows:22

\[
w(t) = w_{\text{max}} - \frac{w_{\text{max}} - w_{\text{min}}}{\text{Iter}_{\text{max}}} \times N_{\text{iter}}.
\]

(5)

![Figure 5. Image identification procedure.](image)

![Figure 6. Picture of the cross mask position results by center of gravity method.](image)

![Figure 7. Architecture of PCI-8143 motion card control.](image)
error (ISE) are very similar. Each particle constituted with four parameters that are assigned real values for the PID’s proportional gain, integral gain, derivative gain, and velocity feed forward gain, respectively. For \( n \) individuals in a population, the dimensions of the population are \( n \times 4 \). The matrix for a population with a total of 20 particles is as follows: \( \mathbf{K}_n = [K_p, K_i, K_d, K_f] \), \( n = 1, 2, \ldots, 20 \).  

\[
\text{IAE} = \int_0^\infty |e(t)|dt.
\]

Figure 8 presents a rough summary of the PSO algorithm.

**LSTM network**

LSTM is a deep learning model used to solve the problem of gradient vanishing in recurrent neural networks (RNNs). LSTM networks handle nonlinear data effectively, and they are often used for long-term dependencies. Compared with RNNs, LSTM networks use more complex gate structures to process information and storage units to determine which data to forget or retain. Figure 9 presents the internal structure of LSTM. A neuron consisting of an input gate \( i_t \), forget gate \( f_t \), and output gate \( o_t \) conducts selective learning and stores information. The formulas are as follows:

\[
f_t = \sigma(W_f[h_{t−1},x_t] + b_f),
\]

\[
i_t = \sigma(W_i[h_{t−1},x_t] + b_i),
\]

\[
\tilde{c}_t = \tanh(W_c[h_{t−1},x_t] + b_c),
\]

\[
c_t = f_t \odot c_{t−1} + i_t \odot \tilde{c}_t,
\]

\[
o_t = \sigma(W_o[h_{t−1},x_t] + b_o),
\]

\[
h_t = o_t \odot \tanh(c_t),
\]

where \( x_t \) is the input of the cell at time \( t \), \( h_t \) is the output at time \( t \), \( \sigma \) is the sigmoid function, and \( \tanh \) is the hyperbolic tangent function. The terms \( W_f, W_i, W_c, \) and \( W_o \) are the weight matrices. The terms \( b_f, b_i, b_c, \) and \( b_o \) are the corresponding offset vectors.

**XXY stage motion based on IAE prediction with PSO–LSTM model**

This study uses PSO to determine the XXY stage control parameters. PSO algorithm employs two common stop conditions. The first is a limit on the number of iterations of the search. However, this limit does not result in efficient convergence, but the search time is constant. The second is a target fitness value, but this requires pretesting to obtain a stable convergence of fitness values. Because fitness value depends on
the stage operating conditions, therefore the dynamic fitness value must be adjusted accordingly.

This study proposes predicting fitness values or trends on the basis of updating controller’s parameters and trained data to determine whether PSO search should be continued or not. This method combines LSTM with the time series structure of neural networks to predict bettering fitness value. The training data input to this LSTM structure consist of the particle position errors from the PSO iteration. The position data consist of the three parameters of the PID controller for the \( XX \) stage, and the output training data was derived from the results of the PSO search for the global optimal particle fitness, then the current PSO particle data is used to predict the current best fitness value through the trained LSTM model, as shown in Figure 10.

PSO search can be continued by predicting fitness value and trends to achieve convergence early to stop the search iteration and avoid being unable to obtain the optimal parameters. When the PSO search is terminated early, computation time is saved.

Figure 11 presents the architecture of the PSO–LSTM model. First, the PSO search data are initialized to generate the position and velocity parameters for the particles. The control parameters of each particle are set in the controller, and then the position of the \( XX \) stage is compensated. Furthermore, the dynamic response displacement of the stage is obtained by the CCDs. The fitness \( f_i \) of the motion response is then calculated, and each iteration yields a global fitness value. Figure 12 presents a block diagram of the control with automatic optical inspection system. After several PSO searches, the data from each PSO iteration are submitted into the LSTM architecture to train the model. The training input data are \([K_{p1}, K_{i1}, K_{d1}, K_{ff1}, \ldots, K_{pi}, K_{di}, K_{ffi}, \ldots] \), and the training output data is the fitness \( f_i \) value. After training, the current PSO particle swarm parameters are fed into the LSTM to predict the fitness \( f_{ip} \), and the predicted data are evaluated to determine whether to stop the PSO search. Through this method, the search can be stopped without evaluating the global fitness \( f_{gd} \).

Experiments

This study used PSO and IPSO to test three types of motion of the \( XX \) stage. The first was single-axis motion in the \( Y \) direction. The second was synchronous two-axis motion both in the same \( X \) direction, the \( XX \) stage was controlled by two different motors to control the same \( X \)-axis direction of the stage motion. The third was three-axis circular rotational motion. The onset of stop PSO’s searching of three experiments was as follows. The particle data of PSO in each iteration were used to predict the current fitness value based on position error through the established LSTM network model, then continue the PSO search, and evaluate or call for stop in advance in comparison with the predicted fitness value from LSTM.

\( Y \)-axis experiments

PSO was used to search for the optimal motion control parameters for \( Y \)-axis movement of the \( XX \) stage. The motion command was to move 1 mm at a velocity of 1 mm/s. The parameters of the PSO algorithm were as follows: The dimensions were set to 4, representing the four control parameters on the \( Y \)-axis, \( K_{p1} \), \( K_{i1} \), \( K_{d1} \), and \( K_{ff1} \); the range for the parameters was 0–10,000; the maximal number of iterations was 20; the number of particles was 20; the learning factors \((c_1 \text{ and } c_2)\) were set to 2; and the inertia weight \( w \) was set to 0.6. For the IPSO experiment, the maximal inertia weight \( w_{\text{max}} \) was set to 1, and the minimal inertia weight \( w_{\text{min}} \) was set to 0.2. The experiment was terminated when PSO or IPSO reached the maximal number of iterations. Table 1 presents the parameters.

PSO achieved a dynamic motion response that did not reach the target of 1 mm (Figure 13). This may have been caused by mechanism or image error, since the
mechanical error of the stage is 20 μm, while the image error is about 1–2 pixels with the resolution of the CCD for 10 μm per pixel. Therefore, theoretically, the total error is about 30–40 μm. With the merit of changing inertia weight at each searching iteration, the IPSO was adaptively to obtain a lower fitness value by reducing \( w \) and then strengthen the local search capabilities (Figure 14).

Through the adaptive inertial parameter \( w \), IPSO has better adjustment ability than PSO in global search and local search. After 20 iterations, the fitness values obtained through PSO and IPSO were 3.363 and 3.186, respectively. The PSO dynamic response data were as follows: the peak overshoot \( M_p \) was 4.88%, the steady-state error \( E_{ss} \) was 1.6%, and the settling time \( T_s \) and rise time \( T_r \) were 0.87 and 0.78 s, respectively. IPSO obtained superior fitness values to those obtained by PSO. For IPSO, \( M_p \) was 4.09%, \( E_{ss} \) was 2.3%, and \( T_s \) and \( T_r \) were 0.99 and 1.05 s, respectively. Table 2 presents the data. Although the control parameters were different, the resulting positioning error and dynamics response on the actual platform were pretty similar.

Figure 15 presents a diagram of the particle parameter distributions for PSO and IPSO. PSO exhibited apparently convergence by the seventh iteration (number of iterative particles = 160), and \( K_i \) did not converge apparently. This may be because \( K_i \) had little effect on the system. IPSO exhibited obvious convergence by the fifth iteration (number of iterative particles = 120) when deploying high \( w \) than PSO. Therefore, IPSO exhibits a higher searching ambitious ability at the beginning of the iteration and find lower fitness parameters and fewer iterations in early searching stage. After few iterations, the fitness value in the middle searching iterations is more likely constant.
With decreasing \( w \), IPSO can be used to search for optimal control parameters in a conservative way and reach to bettering fitness value more effectively. As a result, the overall dynamic step responses in Figure 13 and the fitness value history in the end of Figure 15(a) and (b) found by the two methods of PSO and IPSO were quite similar.

Since the trend of fitness values of the PSO and IPSO differs and depends on initial parameters. The characteristics of time-dependence LSTM was used to find the history of fitness value obtained by PSO and IPSO as repetitive learning (as shown in Figure 11). First, the baseline configuration of the LSTM network was determined by using input and output data, which consisted of particle parameters and fitness values, respectively. Then, fitness was predicted before each iteration of the PSO search. The LSTM model is implemented by using Keras library and contains one LSTM layer and a dense layer.

To train the LSTM model, the weights and biases were updated through Adam optimization. The mean absolute error and accuracy were used as the metrics to evaluate performance. Number of time steps was 6, the input data dimension was 80 (20 particles × 4 control parameters), the number of epochs was 50. The hyper-parameters of the LSTM were adjusted using an optimization Adam algorithm because of its ability to quickly find the optimal solution.

As shown in Figure 16, for the first seven PSO searches, the data from the previous PSO search history were input into the LSTM model for training, and then the parameters of the 20 particles were used to predict the optimal fitness value for next iterations. After LSTM’s prediction with new coming input data, the predicted fitness value curve was highly similar to the actual curve. In Figure 17, the loss function of LSTM model is depicted by the initial model training (blue), current training (orange), initial model validation (red), and current validation (purple) in \( Y \)-axis by PSO (left) and IPSO (right). The initial LSTM model training line followed by validation line preserves the confidence convergence for PSO or IPSO learning. Besides, the loss value of the initial model training model is higher than the current training of PSO, the tells the LSTM model can transfer the learning ability for the next PSO or IPSO searching ability. The loss of model and current validation by PSO is reduced from 0.021 to 0.012, and the IPSO is reduced from 0.027 to 0.01, the train- ing loss of PSO and IPSO are both close to 0. This concludes through the training of the LSTM model, the stage positioning error can reduced based on is accurate LSTM model by in-line processes. The initial model behaves in great accuracy. Still the current model can be improved after importing new data. Thus, in Figure 17, the loss value of validation lines from initial training model (red) to current model (purple) can be reduced. Experimental results provide a significant solution for future machine in mass production line via initial PSO tuning followed by LSTM model.

Based on retraining and fitness prediction by LSTM,

| Type | \( K_p \) | \( K_i \) | \( K_d \) | \( K_{ff} \) | Fitness (IAE) | \( M_p \) (%) | \( T_s \) (s) | \( T_r \) (s) | \( E_m \) (%) |
|------|--------|--------|--------|--------|-------------|-------------|------------|------------|-------------|
| PSO  | 7502   | 1139   | 1469   | 0      | 3.363       | 4.88        | 0.87       | 0.78       | 1.6         |
| IPSO | 10,000 | 1332   | 1586   | 5907   | 3.186       | 4.09        | 1.05       | 0.99       | 2.3         |
proposed method resolve machine positioning accuracy for individual stage with different machine’s health status. Application with stage’s prognostic diagnosis can also deploy in shop floor system.

**X-axis experiments**

Next, PSO was used to search for the optimal control parameters for XXY stage motion along the X-axis. The motion command was to move 1 mm at 1 mm/s; the stage’s movement in the X direction was driven by two motors, X1-axis and X2-axis, simultaneously. Therefore, the gantry parameters of the two motors, $GK_p$ and $GK_v$, required adjustment for the X-axis. Since, the XXY stage is coplanar, and the structure of the two motors is similar. Neglect all the mechanical error, the $K_p, K_i, K_d,$ and $K_{ff}$ were the same for the two motors. The parameters of the PSO algorithm were as follows: The dimensions were set to 6 (in Table 2), representing the six control parameters, $K_p$, $K_i$, $K_d$, $K_{ff}$, $GK_p$, and $GK_v$, for the two motors; the range of the parameters was set to 0–10,000; the maximal number of iterations was set to 20; the number of particles was set to 20; the learning factors ($c_1$ and $c_2$) were set to 2; and the inertia weight $w$ was set to 0.6. In the IPSO experiment, maximal inertia weight $w_{max}$ was set to 1.
and the minimal inertia weight $w_{\text{min}}$ was set to 0.2. The experiment was terminated when PSO or IPSO reached the maximal number of iterations.

The control parameters for the $X$-axis motion of the $XY$ stage that were obtained through PSO did not reach the target within the error of 1 mm. This may have been caused by mechanism or image error (Figure 18). The fitness values obtained through PSO and IPSO were 6.165 and 5.933, respectively. IPSO was able to search lower fitness. When $GK_p$ and $GK_v$ were added, the search conditions became more complex. The control parameters identified during the experiment were dissimilar, but the dynamic response of the stage with the PSO and IPSO optimization was still similar, which means that there were many solutions for the optimal control parameters of this experiment. Resulting PSO dynamic response data are as follows: $M_p$ was 7.34%, $E_{\text{ss}}$ was 4.6%, and $T_s$ and $T_r$ were 1.11 and 0.84 s, respectively. The IPSO obtained superior fitness values: $M_p$ was 7.33%, $E_{\text{ss}}$ was 4.5%, and $T_s$ and $T_r$ were 1.08 and 0.84 s, respectively. Table 3 presents the details.

Figure 19 presents the particle parameter distribution diagram for PSO and IPSO. PSO exhibited apparent convergence by the seventh iteration (number of iterative particles = 160), and IPSO exhibited apparent convergence by the fourth iteration (number of iterative particles = 100), which was caused by initial higher $w$ effect. IPSO started with a higher $w$ at the beginning of the iteration and obtained lower fitness and exhibited fewer iterations. Still, the fitness values obtained through these two methods and their step responses were quite similar in Figures 18 and 20.

During the seventh PSO search, the data from the previous PSO search were input to the LSTM model for training, and the parameters of the 20 particles were used to predict the optimal fitness. After LSTM’s model prediction, the curve for predicted fitness was highly similar to the curve in actual fitness value. When the PSO search reached a stable level of fitness value during the sixth and seventh iterations, the prediction

![Figure 17. Loss of LSTM model for initial model training (blue), current training (orange), initial model validation (red), and current validation (purple) for $Y$-axis with PSO (left) and IPSO (right): (a) PSO and (b) IPSO.](image)

![Figure 18. X-axis step response by PSO and IPSO.](image)

Table 3. Indices of $X$-axis performance for controller tuned by PSO and IPSO.

| Type | $K_p$ | $K_i$ | $K_d$ | $K_{ff}$ | $GK_p$ | $GK_v$ | Fitness (IAE) | $M_p$ (%) | $T_s$ (s) | $T_r$ (s) | $E_{ss}$ (%) |
|------|-------|-------|-------|--------|--------|--------|--------------|----------|----------|----------|----------|
| PSO  | 7965  | 0     | 9897  | 7833   | 7922   | 893    | 6.165        | 7.34     | 1.11     | 0.84     | 4.6       |
| IPSO | 9828  | 0     | 10,000| 4780   | 4564   | 776    | 5.933        | 7.33     | 1.08     | 0.84     | 4.5       |
curve did not noticeably change. This stable level of fitness was used to determine whether to stop the PSO search (Figure 21). As shown in Figure 22, the loss of LSTM model and current validation by PSO is reduced from 0.05 to 0.005, and the IPSO is reduced from 0.022 to 0.003, while the training loss of PSO and IPSO are both close to 0 (Figure 22). This concludes through the training of the LSTM model, the stage positioning error and time of optimal control parameters in X-axis can reduced significantly with the assistance of LSTM model’s prediction for in-line processes.

**Circular motion experiments**

Next, PSO was used to search for the optimal control parameters for the XXY stage to move a circle by three axes. The radius of the circle was set to 1 mm, the speed was set to 1 mm/s, and the stage’s movement along the X-axis was driven by the two motors, X1-axis and X2-axis, simultaneously. The stage’s movement along the Y-axis was driven by the motor Y-axis. The two gantry control parameters, GKP and GKn, required adjustment for the X-axis movement. Because single-axis control was used for the Y-axis, the gantry parameter for the Y-axis was 0. Because the XXY stage is coplanar, the structure of the motors is similar. KP, KI, KD, and KFF were the same for the three axes. The parameters of the PSO algorithm are as follows: The dimensions were set to 6, representing the six control parameters, KP, KI, KD, KFF, GKp, and GKv, on the three axes. Table 1 presents the parameters. The fitness value for the circular motion was circularity. The results for control of the three-axis circular motion of the XXY stage are presented as circularity data. Once the farthest and closest distances have been calculated, then the circularity value results from a simple subtraction between them. The experiment was terminated when PSO or IPSO reached the maximal number of iterations.

With PSO used to obtain the control parameters for circular motion, the radius of the circle did not reach 1 mm. This may have been caused by mechanism or image error (Figure 23) The fitness values obtained through PSO and IPSO were 0.034 and 0.024, respectively. IPSO identified lower fitness value by using a lower w. For PSO, the farthest distance Dmax was 0.978, and the closest distance Dmin was 0.944. For IPSO, the farthest distance Dmax was 0.955, and the closest distance Dmin was 0.931. The two sets of data were compared. IPSO was able to approximate the circle control parameters. Table 4 presents the data. Figure 24 presents the particle parameter distribution diagram for PSO and IPSO. PSO exhibited clear convergence by the
second iteration (number of iterative particles = 60), and IPSO exhibited clear convergence by the fifth iteration (number of iterative particles = 120). Although IPSO did not search the optimal parameters as quickly, its fitness value was superior to that of PSO in each iteration because $w$ decreases at each iteration for convergence (Figure 25).

With the same seven PSO searches, the fitness curve LSTM prediction was highly similar to the actual fitness curve. In Figure 26, the LSTM predicted the downward trend for the fitness curves by the PSO and IPSO optimization processes. This allowed whether the PSO or IPSO carries the clear-cut decision to continue searching the best optimal parameters or not. Figure 27 shows the loss of model for current validation via PSO is reduced from 0.004 to 0.002, and the IPSO is reduced from 0.01 to 0.008, the training loss of PSO and IPSO are both close to 0.

**Conclusion**

This paper used PSO and IPSO to obtain the six controller parameters for the precision motion control of the $XXY$ stage. The bettering movement accuracy by the PSO optimization was calculated using image feedback automatic optical inspection system. The dynamic positioning performance was evaluated thereof. PSO control system associated with established long short-term memory (LSTM) prediction models with the functions of learning, storing, and transmitting memory for monitoring the value of

---

**Figure 21.** The initial training followed by prediction with real fitness values (IAE of the position error) by LSTM model for X-axis by (a) PSO and (b) IPSO methods.

**Figure 22.** Loss of LSTM model for initial model training (blue), current training (orange), initial model validation (red), and current validation (purple) for X-axis with PSO (left) and IPSO (right): (a) PSO and (b) IPSO.
fitness function of the controller parameters was implemented successfully. This can reduce the search of control parameters time with a prognostic-diagnosis-like manner for the multi-dimensional parameters demand.

Experimental results showed the PSO-LSTM and IPSO–LSTM based models improved tracking accuracy both for linear and circular trajectories. Therefore, replacing linear optical scales with CCD cameras to capture the stage positioning by PSO-LSTM based controller is feasible. Through the training of the LSTM model based on the PSO’s or IPSO’s information, the fitness function of PSO and IPSO can

![Figure 23. Trajectory of circular motion by PSO and IPSO tuning: (a) PSO and (b) IPSO.](image)

![Figure 24. (a) Variation of PID controller gains, feedforward gain and fitness value with number of iterative PSO’s particles in circular motion and (b) variation of PID controller gains, feedforward gain, and fitness value with number of iterative IPSO’s particles in circular motion.](image)

| Table 4. Indices of performance of controller tuned by PSO and IPSO for circular motion. |
| Type | $K_p$ | $K_i$ | $K_d$ | $K_{ff}$ | $G_K_p$ | $G_K_v$ | Fitness (Circularity) | $D_{max}$ (mm) | $D_{min}$ (mm) |
|------|------|------|------|--------|--------|--------|---------------------|----------------|----------------|
| PSO  | 10,000 | 9743 | 4239 | 10,000 | 4211   | 4119   | 0.034              | 0.978          | 0.944          |
| IPSO | 9914  | 0    | 447  | 5380   | 0      | 0      | 0.024              | 0.955          | 0.931          |
be predicted effectively. Therefore, the time of optimal control parameters searching in single translation movement (Y-axis), synchronous moving by two motor (two X-axis motor), and circular trajectory motion can predict the effectively with the characteristics of LSTM model’s prediction for real time application. Proposed method can resolve the consuming time for searching the bettering fitness value in optimization. Especially when the assembly of a precision stage demands time for fine tune of the controller parameters. In practice, knowing the machine positioning accuracy is a key priority when the remaining useful life of motion stage could change after long time operation. As for prognostic health maintenance, proposed method with PSO-LSTM model can deploy with precision equipment in shop floor.

Figure 25. PSO and IPSO fitness value with number of iteration as circular motion.

Figure 26. The initial training followed by prediction with real fitness values (IAE of the position error) by LSTM model for circular motion by (a) PSO and (b) IPSO methods.
Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The authors thanks the Ministry of Science and Technology for financially supporting this research under Grant MOST 109-2221-E-018-001-MY2 in part.

ORCID ID
Yi-Cheng Huang https://orcid.org/0000-0002-4065-7731

References
1. Chen HM, Lee TE, Su JP, et al. Realization of an image-based XXY positioning platform control. In: 2014 International symposium on computer, consumer and control, Taichung, Taiwan, 10–12 June 2014, pp.1187–1190. New York, NY: IEEE.
2. Lee HW, Liu CH and Chen JS. Development of a high precision edge alignment system for touch-panel glass substrates. Adv Mech Eng 2014; 6: 904061.
3. Yim HJ, Lee YH, Lee SH, et al. Multi-layer alignment control of a large area nano-imprinting stage. J Mech Sci Technol 2009; 23: 1094–1097.
4. Yesin KB and Nelson BJ. A CAD model based tracking system for visually guided microassembly. Robotica 2005; 23: 409–418.
5. Shieh TM. Automatic image positioning system and its application for mark positioning, master thesis of Institute of Manufacturing Information and Systems. Master Thesis, National Cheng Kung University, TW, 2005.
6. Wang L, Ren L, Mills JK, et al. Automatic 3D joining in microassembly. In: 2007 international conference on information acquisition, Seogwipo, Korea, 8–11 July 2007, pp.292–297. New York, NY: IEEE.
7. Kim HT, Lee KW, Yang HJ, et al. A self-learning method for automatic alignment in wafer processing. Int J Precis Eng Manuf 2013; 14: 215–221.
8. Huang SJ and Lin SY. Application of visual servocontrol X–Y table in color filter cross mark alignment. Sens Actuator A Phys 2009; 152: 53–62.
9. Kuo WM, Chuang SF, Nian CY, et al. Precision nanoalignment system using machine vision with motion controlled by piezoelectric motor. Mechatronics 2008; 18: 21–34.
10. Lin CJ, Chen GZ, Huang YX, et al. Computer-integrated micro-assembling with image-servo system for a microdroplet ejector. J Mater Process Technol 2008; 201: 689–694.
11. Lee HW and Liu CH. Vision servo motion control and error analysis of a Coplanar XXY stage for image alignment motion. Math Prob Eng 2013; 2013: 1–12.
12. Lee HW, Liu CH, Chiu YY, et al. Design and control of an optical alignment system using a parallel XXY stage and four CCDs for micro pattern alignment. In: 2012 symposium on design, test, integration and packaging of MEMS/MOEMS, Cannes, France, 25–27 April 2012, pp.13–17. New York, NY: IEEE.
13. Yang CM, Wen CC, Lin SW, et al. Application of image servo alignment module design to automatic laminating machine for touch panel. Smart Sci 2013; 1: 75–81.
14. Lin CJ, Yu SK, Li YC, et al. Design and control of an optical alignment system using a XXY stage integrated with dual CCDs. Smart Sci 2014; 2: 160–167.
15. Lin CJ, Hsu HH, Cheng CH, et al. Design of an imageservo mask alignment system using dual CCDs with an XXY stage. Appl Sci 2016; 6: 42.
16. Fang H and Chen L. Application of an enhanced PSO algorithm to optimal tuning of PID gains. In: 2009 Chinese control and decision conference, GuiLin, China, 17–19 June 2009, pp.35–39. New York, NY: IEEE.
17. El-Gammal AAA and El-Samahy AA. A modified design of PID controller for DC motor drives using particle swarm optimization PSO. In: 2009 international conference on power engineering, energy and electrical drives, Lisbon, Portugal, 18–20 March 2009, pp.419–424. New York, NY: IEEE.
18. Rout B and Pati BB. MFO ptimized fractional order based controller on power system stability. *Proc Eng Technol Innovation* 2018; 8: 46–59.
19. Kennedy J and Eberhart R. Particle swarm optimization. In: *Proceedings of ICNN95 - international conference on neural networks*, Perth, Australia, 27 November–1 December 1995, pp.1942–1948, New York, NY: IEEE.
20. Solhiin MI and Wahyudi AR. Self-erecting inverted pendulum employing PSO for stabilizing and tracking controller. In: *2009 5th international colloquium on signal processing & its applications*, Kuala Lumpur, Malaysia, 6–8 March 2009, pp.63–68. New York, NY: IEEE.
21. Huang YC, Su YW and Chuo PC. Iterative learning control bandwidth tuning using the particle swarm optimization technique for high precision motion. *Microsyst Technol* 2017; 23: 361–370.
22. Huang YC and Li YH. Experiments of iterative learning control system using particle swarm optimization by new bounded constraints on velocity and positioning. *Eng Comput* 2014; 31: 250–266.
23. Sagheer A and Kobl M. Time series forecasting of petroleum production using deep LSTM recurrent networks. *Neurocomputing* 2019; 323: 203–213.
24. Zhang YZ, Xiong R, He HW, et al. Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries. *IEEE Trans Veh Technol* 2018; 67: 5695–5705.
25. Zhang J, Zhu Y, Zhang X, et al. Developing a long short-term memory (LSTM) based model for predicting water table depth in agricultural areas. *J Hydrology* 2018; 561: 918–929.
26. Qin Y, Xiang S, Chai Y, et al. Macroscopic-microscopic attention in LSTM networks based on fusion features for gear remaining life prediction. *IEEE Trans Ind Electron* 2020; 67: 10865–10875.
27. Gao X, Guo Y, Hanson DA, et al. Thermal error prediction of ball screws based on PSO-LSTM. *The international journal of advanced manufacturing technology* 2021; 116: 1721–1735.
28. Qiu YY, Zhang Q and Lei M. Forecasting the railway freight volume in China based on combined PSO-LSTM model. *J Phys Conf Ser* 2020; 1651: 012029.
29. Chiauan Yan Technology Co., Ltd. XXY-25-7 model description, https://www.aafteck.com/en/product-609758/XXY25-07-XXY25-07.html (2015, accessed 13 May 2021).
30. ADLINK TECHNOLOGY INC. Motion control card user’s manual, PCI-8253/8256 DSP-based 3/6-axis analog motion control card user’s manual, January 2009.
31. Hochreiter S and Schmidhuber J. Long short-term memory. *Neural Comput* 1997; 9: 1735–1780.
32. Jun C and Tao W. Tracking of mobile robot system. In: *2011 international conference on consumer electronics, communications and networks (CECNet)*. Xianning, China, 16–18 April 2011, pp.4600–4602. New York: IEEE.