2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Online Parameter Optimization in Robotic Force Controlled Assembly Processes

Hongtai Cheng, *Member, IEEE,* and Heping Chen, *Senior Member, IEEE*

***Abstract*— In the high precision robotic assembly processes, the process parameters have to be tuned in order to adapt to variations and satisfy the performance requirements. However, because of the modeling difficulty and low efficiency of the existing solutions, this task is usually performed offline. In this paper, an online parameter optimization method is developed. Gaussian Process Regression(GPR) is utilized to model the relationship between the process parameters and system performance. The GPR surrogated Bayesian Optimization Algorithm(GPRBOA) is proposed to optimize the process parameters. To reduce the risk of converging to a local minimum, a random variation factor is added to the Lower Confidence Bound(LCB) acquisition function to balance the exploration and exploitation processes. To deal with the computational burden of GPR, a switching criterion is proposed to coordinate the optimization process and production process to reduce the computational complexity. Experiments were performed using a peg-in-hole process. The experimental results verify the effectiveness of the proposed algorithm and demonstrate its efficiency and accuracy compared to Design Of Experiment(DOE) methods. The proposed method is the first attempt of model-driven assembly process parameter optimization and will generate big economic impact.**

I. Introduction

In the manufacturing processes, the environment is constantly changing and parts to be processed could come from different batches and sometimes different suppliers. All of these variations will cause difficulty for conventional industrial robots to perform many manufacturing processes[1], [2], for example, the valve body assembly process as shown in Figure 1.

Installing a valve into a valve body is not always as easy as it looks. The radius of the valve is about 24.96 mm while the radius of the hole in the valve body is 25.00 mm with a clearance about 40 μm. Because of the fixture errors, the valve cannot be aligned with the holes on the valve body exactly. Therefore, the valve can be stuck at the surface of the valve body due to the positioning errors or jammed in the middle of the valve body due to the orientation errors. Thus several parameters are involved in this assembly process, such as search force, search speed, search radius and insertion force. The assembly process performance will decrease if these parameters are not tuned correctly to adapt to the variations.

Several offline algorithms have been proposed to solve the assembly process parameter optimization problem[3], [4]. The Genetic algorithms (GA) are developed to randomly search for optimal parameters[5], [6]. To increase the efficiency of the GA based methods, Artificial Neural Network (ANN) is utilized to model whether the parameters are “good” or “bad” to filter the candidate parameters first without performing any experiment[7]. The design-of-experiment (DOE) methods[8], [9] adopt a systematic way to optimize the parameters. After performing a series of experiments, the most sensitive parameters are chosen and tuned carefully[10], [11]. Even though these methods are effective in offline parameter optimization, it is unreasonable to use them online because of their low efficiency.

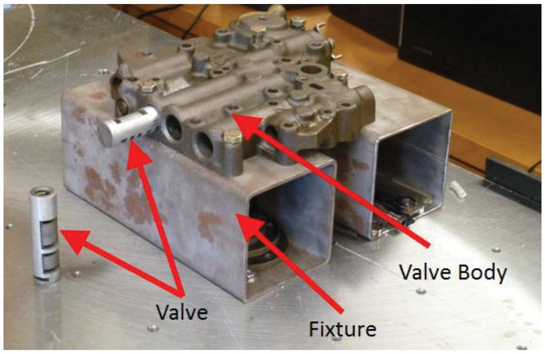


Fig. 1. The valve body assembly process.

To our best knowledge, no practical online parameter optimization algorithm has been proposed for complex assembly processes. Because the assembly processes typically have many stages and different control strategies such as hopping and searching, it is hardly possible to construct a physical model to optimize the process parameters. Therefore an online parameter optimization method without using any physical model has to be investigated. Gaussian Process Regression(GPR) is a non-parametric tool that can easily handle the modeling problem with noisy observations and system uncertainties. The Gaussian Process Regression surrogated Bayesian Optimization Algorithm (GPRBOA) can iteratively model a complex system and optimize the system performance. Hence it is an ideal online solution to solve the process performance optimization problem.

In this paper, GPRBOA is proposed to model the complex high precision assembly process and optimize the process parameters online. Experiments performed using the valve body assembly process demonstrate the effectiveness of the proposed algorithm. Compared to the DOE based methods, the efficiency and accuracy are greatly improved; moreover it can be applied online without stopping an assembly line to perform experiments for parameter optimization.

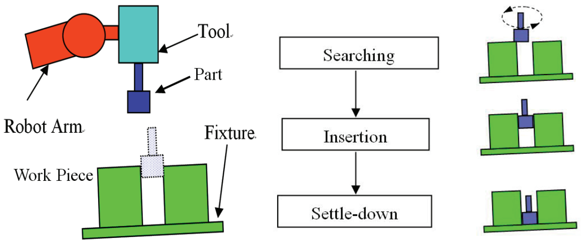
H. Cheng is with Department of Mechanical Engineering and Automation, North East University, China and the Ingram School of Engineering, Texas State University, USA, chenght@me.neu.edu.cn

H. Chen are with the Ingram School of Engineering, Texas State University, hc15@txstate.edu

978-1-4799-3684-7/14/$31.00 ©2014 IEEE

II. Problem Formulation(High Precision Assembly Process)

A high precision robotic assembly requires a robot to perform assemblies in which the assembly clearance is close to or better than the robots' repeatability. Figure 2(a) shows a robotic system to perform a high precision peg-in-hole assembly process. The robot tool picks up a part and assembles it into the workpiece.



(a) (b)

Fig. 2. a) A robotic system to perform a high precision assembly process. b) Steps to perform the high precision assembly process.

Figure 2(b) shows the steps to perform the assembly process. A searching method is used to find the exact location of the workpiece. After the part is engaged with the workpiece, an insertion force is applied to insert the part into the workpiece. During insertion, the tool orientation is changed according to the orientation of the workpiece to avoid jam.

The clearance and geometry of parts from different batches, sometime different suppliers, are different. These variations will cause the increase of cycle time; moreover, assembly failure if the assembly process parameters do not change. To deal with the problem, the assembly process parameters should be tuned to adapt to the variations. However, it is difficult to tune all these parameters since the relationship between the parameters and system performance is not clear.

III. Proposed Solution

As mentioned previously, the model-free optimization algorithms such as GA and DOE randomly search optimal parameters based on experimental results. Therefore, such low efficient algorithms can only be used offline. To increase the parameter optimization efficiency, a model-based method is proposed to solve the modeling and parameter optimization problems online.

A. Gaussian Process Regression

Gaussian Process Regression(GPR) is a non-parametric tool that can handle the modeling problem with noisy observations and system uncertainties[12]. It is widely used to model geographical terrains, dynamical systems, nonlinear systems, complex environments and sensor networks etc.

For a Gaussian Process $f(\mathbf{x})$, a set of multivariate Gaussian random variables $\mathscr{F} = \{f(\mathbf{x}_1), f(\mathbf{x}_2),,\ldots, f(\mathbf{x_N})\}$ can be defined over $\mathscr{X}$ with any finite set of N points $\{\mathbf{x}_i \in \mathscr{X}\}_{i=1}^{N}$ where $f(\mathbf{x_i})$ has the value of the latent function $f(\mathbf{x})$ at $\mathbf{x_i} \in \mathscr{X}$ and $\mathscr{X}$ is defined over $\mathbb{R}^D$. $f(\mathbf{x})$ is completely specified by its mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x},\mathbf{x}')$: $f(\mathbf{x}) \sim \mathscr{GP}(m(\mathbf{x}),k(\mathbf{x},\mathbf{x}'))$ where $\mathbf{x}$ and $\mathbf{x}'$ are two arbitrary variables in $\mathscr{X}$.

For a model $y = f(\mathbf{x}) + w$ and $w \sim N\left(0, \sigma_n^2\right)$, the covariance function is $\mathrm{cov}\left(y_i, y_j\right) = k\left(\mathbf{x}_i, \mathbf{x}_j\right) + \sigma_n^2 \delta_{ij}$, where the δ_{ij} is the Kronecker delta which is one iff $i = j$ and zero otherwise. The joint distribution of the observed data set $(\mathbf{X},\mathbf{y})$ and predicted data set $(\mathbf{X}_*,\mathbf{y}_*)$ is

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} K(\mathbf{X},\mathbf{X}) + \sigma_n^2 I & K(\mathbf{X},\mathbf{X}_*) \\ K(\mathbf{X}_*,\mathbf{X}) & K(\mathbf{X}_*,\mathbf{X}_*) \end{bmatrix}\right) \tag{1}$$

where $K(\mathbf{X}',\mathbf{X}'')$ ($\mathbf{X}'$ and $\mathbf{X}''$ refer to $\mathbf{X}$ and $\mathbf{X}_*$) is the covariance matrix whose element K_{ij} in i_{th} row and j_{th} column equals to $k(\mathbf{x_i'}, \mathbf{x_j''})$. By deriving the conditional distribution, we can obtain the predictive function as

$$\mathbf{y}_* | \mathbf{X}, \mathbf{y}, \mathbf{X}_* \sim N\left(\mu\left(\mathbf{y}_*\right), \mathrm{V}\left(\mathbf{y}_*\right)\right) \tag{2}$$

$$\begin{gathered} \mu\left(\mathbf{y}_*\right) = E\left[\mathbf{y}_* | \mathbf{X}, \mathbf{y}, \mathbf{X}_*\right] = K\left(\mathbf{X}_*, \mathbf{X}\right)\left[K(\mathbf{X},\mathbf{X}) + \sigma_n^2 I\right]^{-1} \mathbf{y} \\ \mathrm{V}\left(\mathbf{y}_*\right) = K\left(\mathbf{X}_*, \mathbf{X}_*\right) - K\left(\mathbf{X}_*, \mathbf{X}\right)\left[K(\mathbf{X},\mathbf{X}) + \sigma_n^2 I\right]^{-1} K\left(\mathbf{X}, \mathbf{X}_*\right) \end{gathered} \tag{3}$$

where $\mu(\mathbf{y}_*)$ is the predicted mean for $\mathbf{y}_*$ and $\mathrm{V}(\mathbf{y}_*)$ is the predicted variance. The covariance function $k(\mathbf{x},\mathbf{x}')$ is a key for determining a GPR model. Among various forms of covariance functions, the commonly-used one is the Squared Exponential $k_{\mathrm{SE}}(\mathbf{x},\mathbf{x}')$ which is [12]

$$k_{\mathrm{SE}}(\mathbf{x},\mathbf{x}') = \sigma^2 \exp\left(-\frac{|\mathbf{x}-\mathbf{x}'|^2}{2l^2}\right) \tag{4}$$

where l is a characteristic length-scale factor controlling how close $\mathbf{x}$ and $\mathbf{x}'$ are and σ^2 is the amplitude of the variance. For each covariance function, there are some hyperparameters θ such as l and σ for k_{SE}. The goal of model construction is to find a covariance function $k(\mathbf{x},\mathbf{x}',\theta)$ which fits the data set $(\mathbf{X},\mathbf{y})$ best. Suppose $f(\mathbf{x})$ is a candidate latent function(f in short) for the given data set, the posterior probability is

$$p(f | \mathbf{X}, \mathbf{y}, H, \theta) = \frac{p(\mathbf{y} | \mathbf{X}, f, H, \theta)\, p(f | \mathbf{X}, H, \theta)}{p(\mathbf{y} | \mathbf{X}, H, \theta)} \tag{5}$$

where H is the hypothesis on the structure of the covariance function, θ is the hyperparameters, $\mathbf{X},\mathbf{y}$ are the sample data sets and

$$p(\mathbf{y} | \mathbf{X}, H, \theta) = \int p(\mathbf{y} | \mathbf{X}, f, H, \theta)\, p(f | \mathbf{X}, H, \theta)\, df \tag{6}$$

is the marginal likelihood that refers to the marginalization over the function f.

As mentioned above, in the Gaussian Process Regression, the function $f(\mathbf{x})$ is not given explicitly. Thus, $p(\mathbf{y} | \mathbf{X}, H, \theta)$ actually refers to the likelihood of H and θ given the data set $(\mathbf{X},\mathbf{y})$. The Gaussian assumption makes it possible to derive the analytical solution of the log marginal likelihood

$$\begin{aligned} \log p(\mathbf{y} | \mathbf{X}, H, \theta) &= -\tfrac{1}{2}\mathbf{y}^{\mathrm{T}}\left(K + \sigma_n^2 I\right)^{-1}\mathbf{y} \\ &- \tfrac{1}{2}\log\left|K + \sigma_n^2 I\right| - \tfrac{n}{2}\log 2\pi \end{aligned} \tag{7}$$

Therefore the hyperparameters θ in a covariance function H can be optimized by maximizing the marginal log likelihood:

$$\theta^* = \underset{\theta}{\mathrm{argmax}} \log p(\mathbf{y} | \mathbf{X}, H, \theta) \tag{8}$$

This optimization problem can be solved using different techniques such as Conjuncted Gradient Algorithm[12] or evolutionary algorithms[13]. In this paper, Nelder-Mead(Simplex) method is utilized which is effective and uses only the value of $\log p(\mathbf{y}|\mathbf{X},H,\theta)$. Once $k(\mathbf{x},\mathbf{x}',\theta^*)$ is known, equation (3) can be used to make prediction for new input $\mathbf{x}^*$.

B. Bayesian Optimization Algorithm

The Bayesian Optimization Algorithm(BOA)[14] estimates a probability distribution of promising solutions using Bayesian Network(BN)[15] in order to generate new candidate solutions. The BN model is updated at each iteration using new samples. The BOA achieves a good balance between the modeling difficulty and parameter optimization efficiency and has been applied to solve many optimization problems. However, in each iteration both the BN structure and parameters have to be optimized and this optimization process will increase the computational complexity and difficulty. Therefore, in this paper, the BN is replaced with the GPR model.

The GPR surrogated BOA (GPRBOA) [16], [17] has been implemented in several applications such as environmental monitoring and machine learning. The algorithm can iteratively model a complex system and optimize the system performance. In each iteration, new samples are added into the existing data set and used to update the GP model. The new GP model will then be used to search for the optimal solutions by maximizing a performance index $s(x)$(also known as the acquisition function) over the input domain. Because the acquisition function controls the new sample points, it directly affects the quality of the built model and the optimal solution. Therefore, it plays a key role in the GPRBOA algorithm. Typically the acquisition function defined using $\mu(\mathbf{x})$ and $\sigma(\mathbf{x})$ is deployed to acquire a new candidate using different techniques such as Probability of Improvement, Expected Improvement and Lower Confidence Bound(LCB).

LCB is recently developed based on the idea of exploiting lower confidence bounds (upper, when considering maximization) to increase the optimization efficiency. It has the form

$$LCB(\mathbf{x}) \triangleq \mu(\mathbf{x}) - \kappa \cdot \sigma(\mathbf{x}) \tag{9}$$

where κ is a scaling factor. Instead of only sampling the points with minimum mean $\mu(\mathbf{x})$ or maximum variance $\sigma(\mathbf{x})$ predicted using the current model to improve the model uncertainty, LCB reduces the search space by combining the mean with variance, which can ignore points that has no possibility of being optimal.

C. GPRBOA-VR Algorithm

The difficulties of deploying GPRBOA online come from the following two aspects: the variations of the assembly process[18] and computational burden of GPR[19].

1) Variations of the assembly process: GPRBOA performs online modeling and optimization simultaneously using the exploration and exploitation processes. If the exploration and exploitation processes are not properly balanced, the optimization process can be trapped in local minima. Hence the proposed algorithm must be able to balance the two processes. LCB method can explore a system by sampling $\mathbf{x}$ with large $\mu(\mathbf{x})$ and exploit the model by sampling $\mathbf{x}$ with large $\sigma(\mathbf{x})$. Hence it requires prior information about the variance of cycle time. However, for different batches or assembly processes, such prior information is not available. To deal with such a problem, we propose a new algorithm called GPRBOA-VR to balance the exploration and exploitation processes:

$$\mathbf{x}^* = \begin{cases} \arg\min LCB(\mathbf{x}) & \text{rand}(1) > \varepsilon \\ \arg\max VR(\mathbf{x}) & otherwise \end{cases} \tag{10}$$

where ε is the performance index which is updated online according to the hyperparameters and system performance as shown in equation (12); $VR(\mathbf{x})$(Variation Random) is a random acquisition function used to explore the unsampled area to improve the model quality by investigating the farthest unsampled points:

$$VR(\mathbf{x}) = \min_{i\in 1\cdots N}[d(\mathbf{x}-\mathbf{x_i})] \tag{11}$$

where $d(\mathbf{x}-\mathbf{x}')$ is the distance between two sets of parameters $\mathbf{x}$ and $\mathbf{x}'$. At each iteration, if rand(1) $> \varepsilon$, the new candidate is optimized by exploiting the current model; otherwise, it is calculated by exploring the unsampled parameter space. Hence the exploitation process optimizes the process parameters according to the constructed GPR model while the exploration process refines the model according to random variation.

2) Computational Complexity of GPR: The computational complexity of GPR is proportional to $\mathbb{O}(N^3)$ where N is the number of samples. When N becomes bigger, the computational complexity will increase greatly. Therefore, the optimization process should be terminated once the model becomes stable and the optimal parameters are identified. Meanwhile, if the assembly performance decreases, the optimization process should be restarted to re-optimize the assembly process parameters. Hence a new switching method is proposed to control the parameter optimization process:

$$\varepsilon = \begin{cases} 1 & \Delta\theta > 1 \\ 0 & \Delta\theta < k_\theta, C_t < C_t^*/k_u \\ 0.5 & \Delta\theta < k_\theta, C_t > C_t^*/k_l \\ \Delta\theta & otherwise \end{cases} \tag{12}$$

where k_u, k_l are two constants controlling the optimization process; k_θ is the threshold to determine if a model is converged; C_t is the current cycle time and C_t^* is the best cycle time so far; $\Delta\theta = |\theta_k - \theta_{k-1}| / |\theta_k - \theta_0|$ is the normalized change of hyperparameters, where θ_k is the hyperparameters at iteration k. When a model is converged, $\Delta\theta \to 0$. When $\Delta\theta > 1$, ε is set to 1 to encourage exploring the unknown parameter space; When both $\Delta\theta$ and the cycle time satisfy the given conditions, ε is set to 0 to exploit the model to optimize the parameters and the GPR modeling process stops; When the performance degrades (the cycle time increases), ε is set to 0.5 to restart the optimization process; Otherwise ε is set to $\Delta\theta$ to balance the exploration and exploitation processes.

3) Implementation of GPRBOA-VR: The implementation of the GPRBOA-VR algorithm to optimize the system parameters is shown in Figure 3. There are four blocks within the diagram: GPR modeling, new candidate parameter generation, production/evaluation and switching criterions determination.

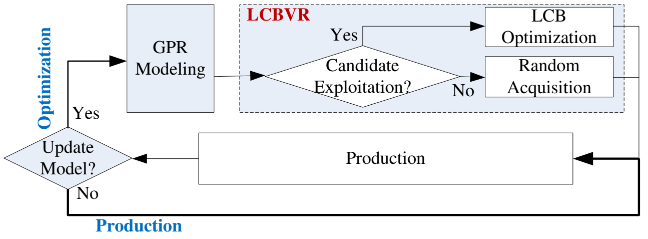


Fig. 3. The GPRBOA-VR assembly parameter optimization algorithm. LCBVR represents Lower Confidence Bound Variation Random.

From the diagram, we can see that the optimization and production loops are integrated to adapt to the process variations and dynamically optimize the assembly process parameters. The algorithm starts from an initial set of parameter candidates and searches for the optimal parameters iteratively. Once a set of optimal parameters is found, the system switches to the production loop and perform the assembly task repeatedly. Meanwhile the system performance is monitored continuously. And once it decreases, the optimization process is restarted and the above processes repeat.

IV. Experimental Results

A. Experimental Setup

To demonstrate the efficiency of the proposed method, experiments were performed using a high precision valve body assembly process as shown in Figure 1. The experimental system as shown in Figure 4 consists of an ABB IRB140 robot with an IRC5 controller, a force sensor mounted on the robot endeffector and a vacuum suction tool used to pick up the valve.

A computer is used for offline and online parameter optimization. The computer is connected to the robot controller via ethernet. ABB force control package is used to perform the assembly process.

The assembly process consists of force guided spiral search and force controlled insertion. The following three parameters are considered: Search Speed(SS), Search Force(SF) and Insertion Force(IF). As listed in TABLE I, three groups of experiments were performed for comparison, where 3P3V refers to three parameters and each parameter has three values; 3PFV refers to three parameters and each parameter has more than three values. In TABLE I, the parameters are defined using the format (Minimum Value: Interval: Maximum Value). Hence for the DOE 3P3V configuration, the search force parameter has 3 possible values:250,300,350; while for the GPRBOA-VR 3PFV configuration, the search force parameter has 11 possible values from 250 to 350 with step size 10.

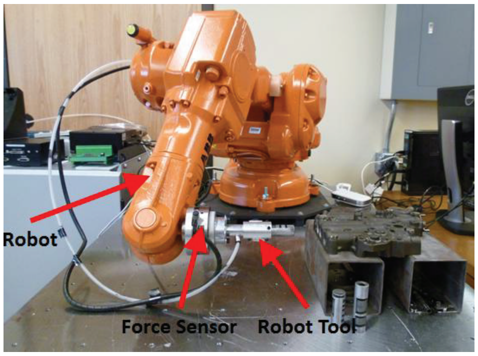


Fig. 4. The experimental system.

The parameters in the algorithm are chosen as $k_\theta = 0.05$, $k_u = 0.95$, $k_l = 0.75$. From equation (12), we know that the model is converged if k_θ is close to 0 and the variation of the cycle time is close to ($k_u = 0.95$) the current best cycle time. If the variation of the cycle time is larger than one-third of the current best cycle time ($k_l = 0.75$), the process parameter optimization should be restarted.

B. DOE Results

The DOE experiments were performed offline. For the 3P3V configuration, there are $3^3 = 27$ sets of parameters. Because the cycle time is affected by several random factors, the performance of each set of parameters has to be statistically calculated. Therefore, experiments were performed 10 times for each set of parameters, i.e., totally 270 experiments were conducted. For the robotic assembly process, it is desired that the cycle time and its variance are small. Using the DOE method, the mean cycle time is 2.3s and variance is 0.09s.

C. GPRBOA-VR Results

The GPRBOA-VR experiments were performed online. Each GPRBOA-VR configuration is repeated twice which are denoted as 3P3V #1, 3P3V #2, 3PFV #1 and 3PFV #2 in short. 3P3V #1 and 3P3V #2 have different initial sample points([350,5,100] and [250,35,50]). After about 10 iterations, the model converges and a set of optimal parameters [350,20,50] is identified.

Compared to the 3P3V configuration, 3PFV GPRBOA-VR splits each parameter more precisely. Thus the underlying relationship between the parameters and the cycle time can be described more accurately. Due to the variation of the assembly process, the derived GPR models in each experiments are not all the same. That is why 3PFV #1 and #2 experiments converge to two sets of parameters [350,24,50] and [350,23.5,50]. However, because the derived models are similar, the optimal parameters are very close.

TABLE I

PARAMETER CONFIGURATIONS. SS, SF AND IF REPRESENT SEARCH SPEED, SEARCH FORCE AND INSERTION FORCE RESPECTIVELY.

EXPERIMENTS	SS	SF	IF	Δx	Δy	$\Delta\phi_x(^\circ)$	$\Delta\phi_y(^\circ)$	REPEAT TIMES
DOE 3P3V	250:50:350	5:15:35	50:25:100	-9±1	±1	±0.6	±0.2	10
GPRBOA-VR 3P3V	250:50:350	5:15:35	50:25:100	-9±1	±1	±0.6	±0.2	2
GPRBOA-VR 3PFV	250:10:350	5:0.5:35	50:5:100	-9±1	±1	±0.6	±0.2	2

The four GPRBOA-VR experiments are plotted in Figure 5. Each experiment can be divided into two stages: optimization and production. The optimization stage took about 8 to 11 assemblies. The algorithm explores the parameter space, builds the underlying GPR model and optimizes the parameters online. Once the optimal parameters are found, the system switches to the production stage to perform the assembly process using the identified optimal parameters.

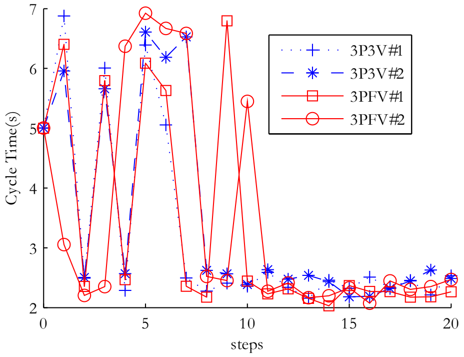


Fig. 5. Experimental results of the GPRBOA algorithm.

D. Discussion

1) Efficiency: The DOE method chooses several values for each parameter, tests the parameter combinations by experiments and finds the optimal one among them. Thus the result is not globally optimal. To overcome the variations of the assembly process and obtain stable result, each parameter set has to be tested several times. Therefore using the full factorial experiments, the number of required experiments can be expressed as $K\prod_{i=1}^{M} N_i$ where M is number of parameters, N_i is the number of values of the i_{th} parameter and K is the repeat times for each parameter set. As the number of parameters and the number of values of each parameter increase, the number of experiments to be performed grows rapidly.

For the GPRBOA-VR method, the number of parameters and the number of values of each parameter do not increase the complexity of the optimization process. From the experimental results shown in Figure 5, we can find that although the parameter combinations increase from $3\times3\times3=27$ to $11\times61\times11=7381$, the number of experiments to identify the optimal parameters does not increase.

By comparing the DOE results to the GPRBOA-VR 3P3V results, it is noted that the GPRBOA-VR method achieves same optimal parameters using 8 experiments instead of 270 experiments using the DOE method. Therefore, the GPRBOA-VR is more efficient than the DOE method.

The optimal parameters, the cycle time (mean and variance) and the corresponding experimental time are listed in TABLE II. The experimental time is obtained by accumulating the cycle time of each experiment(The data processing time for the DOE method is not considered because it is done offline).

TABLE II

COMPARISON OF EXPERIMENTAL RESULTS.

Methods	Optimal	$C_t\mu$(s)	$C_t\sigma$(s)	Number of Experiments	Total Time(s)
DOE	[350,20,50]	2.32	0.09	270	1254
3P3V #1	[350,20,50]	2.39	0.13	8	36.61
3P3V #2	[350,20,50]	2.44	0.14	13	53.7
3PFV #1	[350,24,50]	2.31	0.14	13	54.26
3PFV #2	[350,23.5,50]	2.23	0.1	13	52.05

From TABLE II, similar optimal parameters and cycle time are obtained; however, the parameter optimization time is greatly reduced using the GPRBOA-VR method.

2) Accuracy: From TABLE II we can find that the mean cycle time obtained using GPRBOA-VR 3PFV is better than that using DOE method. This is because more values of each parameter can be explored. Figure 6 shows the modeling process of GPRBOA 3PFV #1 while Search Speed is fixed to be 350 and Insertion Force 50. Hence Figure 6 shows the relationship between average cycle time and Search Force.

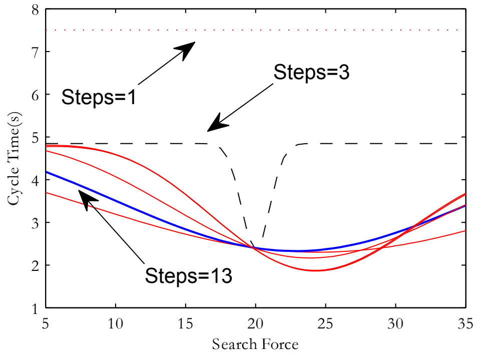
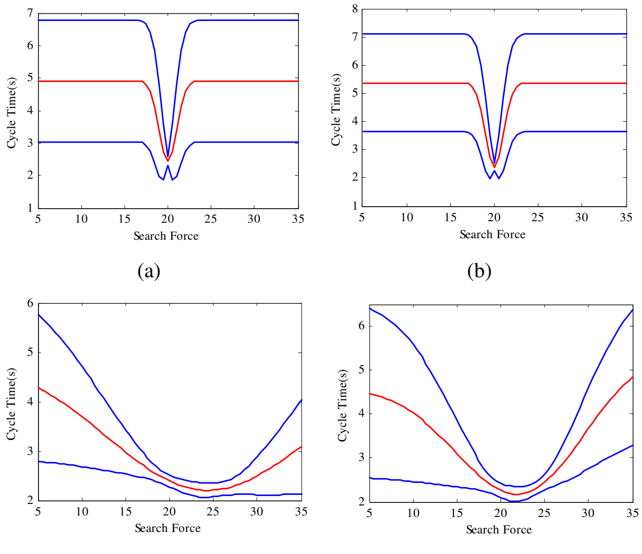


Fig. 6. The relation between average cycle time and search force parameter in 3PFV #1. The model is updated during each step with more and more data sets are considered.

From Figure 6 we can see that the initial model is rough with little useful information. After several iterations, the model converges with a minimal point. From the figure we can see that the minimal point lies between 20 and 25 which cannot be identified using DOE 3P3V. This is why

GPRBOA-VR 3PFV is able to find better parameters than DOE method.

Figure 7 gives the final models for the four GPRBOA-VR configurations.



(a)

(b)

(c)

(d)

Fig. 7. Final model of each GPRBOA-VR experiment. a) 3P3V #1, a) 3P3V #2, a) 3PFV #1, a) 3PFV #2

The blue lines refer to the upper and lower bounds($\mu \pm \sigma$) of the cycle time and red lines refer to the mean cycle time values. Compared to the models built for the 3PFV configurations, those for the 3P3V configurations are rough.

In summary, for the DOE method, because the parameter values cannot be chosen arbitrarily, the real optimal parameters may not be found. Because the GPRBOA-VR method can explore the interesting area in detail without worrying about the complexity, it can identify the optimal parameters to achieve better cycle time.

V. Conclusion

The paper proposes a GPRBOA-VR method to efficiently solve the parameter optimization problem online in high precision robotic assembly processes. The original GPRBOA algorithm is improved to effectively optimize the process parameters to reduce the cycle time by balancing the exploration and exploitation processes. Compared to the existing model-free offline methods, such as DOE and GA, the GPRBOA-VR method can greatly improve the parameter optimization efficiency and accuracy without stopping the production line for experiments. Experimental results demonstrate the effectiveness of the GPRBOA-VR method. The proposed method is the first attempt of model-driven assembly parameter optimization and will generate big economic impact. It can also be used to model complex processes, to optimize the process parameters and improve the system performance.

Acknowledgment

The research is partially sponsored by the Research Enhance Program(REP). Grant No.9000000936, Texas State University, San Marcos. And it is also partially sponsored by the Startup Research Fund. Grant No.02090021233043, Northeastern University, China.

References

[1] F. Dietrich, D. Buchholz, F. Wobbe, F. Sowinski, A. Raatz, W. Schumacher, and F. M. Wahl, “On contact models for assembly tasks : Experimental investigation beyond the peg-in-hole problem on the example of force-torque maps,” *German Research*, pp. 2313–2318, 2010.

[2] W. Jing and W. S. Newman, “Improving robotic assembly performance through autonomous exploration,” in *Robotics and Automation, 2002. Proceedings. ICRA ’02. IEEE International Conference on*, vol. 3, 2002, pp. 3303–3308.

[3] N. Yamanobe, H. Fujii, Y. Maeda, T. Arai, A. Watanabe, T. Kato, T. Sato, and K. Hatanaka, “Optimization of damping control parameters for cycle time reduction in clutch assembly,” in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, 2005, pp. 3251–3256.

[4] J. Wei and W. Newman, “Improving robotic assembly performance through autonomous exploration,” in *Robotics and Automation, 2002. Proceedings. ICRA ’02. IEEE International Conference on*, vol. 3, 2002, pp. 3303–3308.

[5] G. Levitin, J. Rubinovitz, and B. Shnits, “A genetic algorithm for robotic assembly line balancing,” *European Journal of Operational Research*, vol. 168, no. 3, pp. 811 – 825, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221704004874

[6] J. Marvel, W. Newman, D. Gravel, G. Zhang, and T. Fuhlbrigge, “Automated learning for parameter optimization of robotic assembly tasks utilizing genetic algorithms,” *2008 IEEE International Conference on Robotics and Biomimetics*, pp. 179–184, Feb. 2009.

[7] J. Marvel and W. Newman, “Model-assisted stochastic learning for robotic applications,” *Automation Science and Engineering, IEEE Transactions on*, vol. 8, no. 4, pp. 835–845, 2011.

[8] T. Gruendling, M. Guilhaus, and C. Barner-Kowollik, “Design of experiment (doe) as a tool for the optimization of source conditions in sec-esi-ms of functional synthetic polymers synthesized via atrp,” *Macromolecular Rapid Communications*, vol. 30, no. 8, pp. 589–597, 2009. [Online]. Available: http://dx.doi.org/10.1002/marc.200800738

[9] J. Antony, *Design of Experiments for Engineers and Scientists*, 1st ed. Elsevier, 2003.

[10] D. Gravel, G. Zhang, A. Bell, and B. Zhang, “Objective metric study for DOE-based parameter optimization in robotic torque converter assembly,” *The IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3832–3837, Oct. 2009.

[11] ——, “Robot learning and self optimization of process parameters,” in *ASME/ISCIE International Symposium on Flexible Automation*, St. Louis, MO, 2012, pp. 1–7.

[12] C. E. Rasmussen, “Gaussian processes for machine learning.” MIT Press, 2006.

[13] D. Petelin, B. Filipic, and J. Kocijan, “Optimization of gaussian process models with evolutionary algorithms,” in *Adaptive and Natural Computing Algorithms*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, vol. 6593, pp. 420–429.

[14] M. Pelikan, D. E. Goldberg, and E. Cantu-Paz, “Boa: The bayesian optimization algorithm.” Morgan Kaufmann, 1999, pp. 525–532.

[15] C. Lima, F. Lobo, M. Pelikan, and D. Goldberg, “Model accuracy in the bayesian optimization algorithm,” *Soft Computing*, vol. 15, no. 7, pp. 1351–1371, 2011. [Online]. Available: http://dx.doi.org/10.1007/s00500-010-0675-y

[16] Y. Jin, “Surrogate-assisted evolutionary computation: Recent advances and future challenges,” *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61 – 70, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2210650211000198

[17] A. Jalali, J. Azimi, and X. Fern, “Exploration vs exploitation in bayesian optimization,” *CoRR*, vol. abs/1204.0047, 2012.

[18] J. Marvel and J. Falco, “Best practices and performance metrics using force control for robotic assembly.” 2012. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/ir/2012/NIST.IR.7901.pdf

[19] C. Park, J. Z. Huang, and Y. Ding, “Gplp: a local and parallel computation toolbox for gaussian process regression,” *J. Mach. Learn. Res.*, vol. 13, pp. 775–779, Mar. 2012. [Online]. Available: http://dl.acm.org/citation.cfm?id=2188385.2188411