

# Modeling Complex Robotic Assembly Process Using Gaussian Process Regression

Binbin Li\*, Hongtai Cheng<sup>†</sup>, Heping Chen\* and Tongdan Jin\*

\*Ingram School of Engineering

Texas State University, San Marcos, Texas 78666, USA

Email: {b\_l77, hc15, tj17}@txstate.edu

<sup>†</sup>Department of Mechanical Engineering and Automation

Northeastern University, Shenyang, Liaoning 110819, China

Email: chengt@me.neu.edu.cn

**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. For complex systems and processes operating in a stochastic environment such as the assembly processes, experiments and evaluations could be costly and low efficiency because of resource utilization, energy consumption, and dedicated labor. In order to improve the assembly process performance, we investigate the modeling problem for robotic assembly processes. Gaussian Process Regression, a non-parametric modeling technique, is chosen to model the relationship between the assembly process parameters and performance. The main challenge in implementing Gaussian Process Regression is to find suitable covariance functions which can minimize the modeling errors. Therefore we investigated different combinations of basic covariance functions and implemented them to explore the most suitable covariance function for an assembly process. The performance of the built models is compared and the covariance functions with the best performance are identified. An off-line modeling algorithm is appropriately developed using the identified covariance function. The effectiveness and accuracy of the proposed algorithm are further demonstrated by experiments, which were performed using a robotic valve body assembly process.

**Index Terms**—parameter optimization, Gaussian Process Regression, covariance function, hyperparameters.

## I. INTRODUCTION

In the modern manufacturing processes, the assembly environment is always changing and the 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] and sometimes a whole assembly station has to be stopped.

Several offline algorithms have been proposed to solve the assembly process parameter optimization problem [2]. To randomly search for optimal parameters, Genetic Algorithms (GA) are developed as well [3]. Moreover, 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 as to increase the efficiency of the GA based methods [4]. However, the lack of an effective model which describes the relation between the parameters and the performance still limits the overall efficiency.

Gaussian Process Regression (GPR) is a non-parametric tool that can easily handle the modeling problem with noisy observations and system uncertainties [5]. It is widely used to model geographical terrains [6], dynamical systems [7], nonlinear systems [8], complex environments [9] and sensor networks [10] etc. Given a certain amount of observed data set and a covariance function, it can be trained to make predictions at the new input.

The key challenge in implementing GPR is to find the suitable covariance function, which is the crucial ingredient in a Gaussian process predictor, as it encodes our assumptions about the function which we wish to learn. The covariance function determination is still an open problem. In this paper, different combinations of basic covariance functions are investigated and implemented to identify the most suitable one for an assembly process. A modeling algorithm is then appropriately developed to identify appropriate covariance function. The effectiveness and accuracy of the proposed algorithm are further tested by experiments using a robotic valve body assembly process.

The remaining paper is organized as follows. Section II briefly introduces covariance function selection process in robotic assembly process. Section III presents the covariance function determination algorithm. The experimental results are given in Section IV and the paper is concluded in Section V.

## II. PROBLEM FORMULATION

During the high precision robotic assembly process, the clearance and geometry of parts from different batches, sometimes different suppliers, involve large variations. Correspondingly, these variations will cause the increase of cycle time, which is unexpected during the assembly process; moreover, assembly failure rate will arise if the assembly process parameters do not change. The assembly processes also have many stages and different control strategies such as hopping and searching [11], it is almost impossible to construct a physical model using first principles [12] to optimize the process parameters. Sometimes, the whole assembly line has to be stopped to adjust to the assembly environment. For example, the valve body of assembly process system as shown in Figure 1.

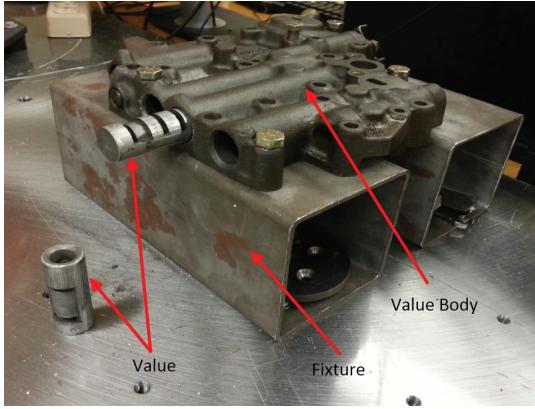


Fig. 1. The valve body assembly process.

It is not always so easy to install a valve into a valve body. The radius of the valve is about 24.96 mm but the radius of the hole in the valve body is only 25.00 mm with a tolerance about 40  $\mu\text{m}$ . Because of the fixture errors, the accumulator cannot be aligned with the holes on the valve body precisely. Therefore, the accumulator itself can be stuck on the surface of the valve body due to the position errors or jammed during the assembly process because of the orientation errors. Several parameters are involved in the 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.

To deal with this problem, the assembly process parameters should be tuned regularly 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 clearly understood. In this case, a model purely using the input data set, like GPR, which can predict the output data set with high precise, is extremely important and needed.

Since the key factor that controls the properties of GPR is the covariance function, to implement GPR with non-parametric modeling method is to find the suitable covariance function for the system. It is difficult to decide which one is better since there exit so many combinations of covariance functions. An assessment system is particularly vital to decide which kind of covariance function is the suitable one to meet the need of the assembly process without adjusting the whole assembly line. Meanwhile, a related testing method is crucial to identify the system and its correctness.

### III. PROPOSED SOLUTION

Because the assembly process is complex, it is almost impossible to derive a model using first principles. The only way is to identify the model indirectly using the observed system input and output and then judge the performance of the model through parameters.

#### A. Gaussian Process Regression

Gaussian Process Regression, which is a non-parametric tool, can handle the modeling problem with noisy observations and system uncertainties [13]. It is widely used to model

geographical terrains [14], dynamical systems [15], nonlinear systems [16], complex environments [17] and sensor networks [18] etc.

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

For a realistic modelling situations with noisy versions  $\mathbf{y} = f(\mathbf{x}) + w$ ,  $w$  is independent identically distributed Gaussian noise with variance  $\sigma_n^2$ ,  $w \sim N(0, \sigma_n^2)$ . The covariance function is  $\text{cov}(y_i, y_j) = k(\mathbf{x}_i, \mathbf{x}_j) + \sigma_n^2 \delta_{ij}$ , where the  $\delta_{ij}$  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

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

By deriving the conditional distribution, we can obtain the predictive function as

$$\mathbf{y}^* | \mathbf{X}, \mathbf{y}, \mathbf{X}_* \sim N(\mu(\mathbf{y}^*), V(\mathbf{y}^*)) \quad (2)$$

$$\begin{aligned} \mu(\mathbf{y}^*) &= E[\mathbf{y}^* | \mathbf{X}, \mathbf{y}, \mathbf{X}_*] = K(\mathbf{X}^*, \mathbf{X}) [K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I]^{-1} \mathbf{y} \\ V(\mathbf{y}^*) &= K(\mathbf{X}^*, \mathbf{X}^*) - K(\mathbf{X}^*, \mathbf{X}) [K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I]^{-1} K(\mathbf{X}, \mathbf{X}^*) \end{aligned} \quad (3)$$

where  $\mu(\mathbf{y}^*)$  is the predicted mean for  $\mathbf{y}^*$  and  $V(\mathbf{y}^*)$  is the predicted variance. The covariance function  $k(\mathbf{x}, \mathbf{x}')$  is a key for determining a GPR model. For each covariance function, there are some hyperparameters  $\theta$ .

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( $\mathbf{f}$  in short) for the given data set, the posterior probability is

$$p(\mathbf{f} | \mathbf{X}, \mathbf{y}, H, \theta) = \frac{p(\mathbf{y} | \mathbf{X}, \mathbf{f}, H, \theta) p(\mathbf{f} | \mathbf{X}, H, \theta)}{p(\mathbf{y} | \mathbf{X}, H, \theta)} \quad (4)$$

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

$$p(\mathbf{y} | \mathbf{X}, H, \theta) = \int p(\mathbf{y} | \mathbf{X}, \mathbf{f}, H, \theta) p(\mathbf{f} | \mathbf{X}, H, \theta) d\mathbf{f} \quad (5)$$

is the marginal likelihood that refers to the marginalization over the function  $f$ . Once  $k(\mathbf{x}, \mathbf{x}', \theta^*)$  is known, the equation (3) can be used to make prediction for new input  $\mathbf{x}^*$ .

#### B. Covariance Function and Diversity

A covariance function is the crucial ingredient in Gaussian Regression predictor, because it determines which function we want to learn. There are a number of covariance functions listed in the Table I. Here  $d = \mathbf{x} - \mathbf{x}'$  and the parameter  $l$  is the characteristic length-scale of the process (practically, “how far apart” two points  $\mathbf{x}$  and  $\mathbf{x}'$  have to be for  $X$  to change

TABLE I  
COMMONLY-USED COVARIANCE FUNCTION

Name	Equation
Constant	$K_c(\mathbf{x}, \mathbf{x}') = C$
Linear	$K_L(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T l^{-1} \mathbf{x}$
Gaussian Noise	$K_{GN}(\mathbf{x}, \mathbf{x}') = \sigma^2 \delta_{\mathbf{x}, \mathbf{x}'}$
Squared Exponential	$K_{SE}(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{ \mathbf{x}-\mathbf{x}' ^2}{2l^2}\right)$
Ornstein-Uhlenbeck	$K_{OU}(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{ \mathbf{x}-\mathbf{x}' }{l}\right)$
Matern	$K_{MA}(\mathbf{x}, \mathbf{x}') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} \mathbf{x}-\mathbf{x}' }{l}\right) K_\nu\left(\frac{\sqrt{2\nu} \mathbf{x}-\mathbf{x}' }{l}\right)$
Periodic	$K_P(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{2\sin^2 \mathbf{x}-\mathbf{x}' }{l^2}\right)$
Rational Quadratic	$K_{RQ}(\mathbf{x}, \mathbf{x}') = \sigma^2 \left(1 + \frac{ \mathbf{x}-\mathbf{x}' ^2}{2\alpha k^2}\right)^{-\alpha}$

significantly),  $\delta$  is the Kronecker delta and  $\sigma$  is the standard deviation of the noise fluctuations.

A complicated covariance function can be defined as a linear combination of other simpler covariance functions in order to incorporate different insights about the data-set at hand. Any function that is positive semidefinite can be a covariance function. Therefore it is a rich family.

The form of covariance function is so rich that it is impossible to traverse all the functions hence the model selection is still an open problem. On the other side, the covariance functions listed in Table I can only meet a particular class of problems. Therefore, it is improper to use this limited cases to solve all the problems. The following properties of the covariance function can be utilized to obtain a more general one.

A covariance function  $k(\mathbf{x}, \mathbf{x}')$  can also be constructed by combining different commonly-used covariance functions  $k_1(\mathbf{x}, \mathbf{x}')$  and  $k_2(\mathbf{x}, \mathbf{x}')$ :

1) *The sum of two covariance functions is a covariance function:* For two covariance functions  $k_1(\mathbf{x}, \mathbf{x}')$  and  $k_2(\mathbf{x}, \mathbf{x}')$  are independent, the random process  $k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}')$  is also a covariance function.

2) *The product of two covariance functions is a covariance function:* For two covariance functions  $k_1(\mathbf{x}, \mathbf{x}')$  and  $k_2(\mathbf{x}, \mathbf{x}')$  are independent, the random process  $k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}')k_2(\mathbf{x}, \mathbf{x}')$  is also a covariance function.

The first operation can be used to combine two covariance functions with different properties such as two with different length scale  $l$  to model the long term and short term variance, or periodic and non-periodic variance. The second operation can be used to modulate the amplitude of a covariance function with another covariance function. With these operations, a complex covariance can be approximated by the common-used ones as listed in Table I. It is a empirical task which should incorporate priori knowledge of the model structure.

### C. Covariance Function Determination Algorithm

The GPR modeling procedure aims to find an appropriate covariance function and hyperparameter which fit the observed data well. Both the model structure and the hyperparameters are obtained based on the data set. To test the performance of the generated model, the data set should be splitted into two

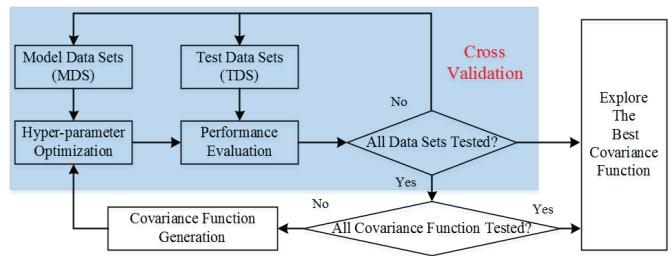


Fig. 2. Constructive Covariance Function Determination Algorithm for Gaussian Process Regression.

parts, one for modeling and the other for testing. Meanwhile, we investigate different combinations of basic covariance functions and implement them to identify the most suitable covariance function for an assembly process.

Given the data set and an appropriate covariance function set, the concept of cross validation can be used to verify the performance of each candidate covariance function  $k(\mathbf{x}, \mathbf{x}')$ . As a consequence, an offline model structure which is then developed using the identified covariance function is shown in Figure 2.

From the diagram, we can see that the process includes the following steps.

- 1) Split the data set. To test the stability of the covariance function and accuracy of the predictions, the original data set is split into two parts. Each time, one of them is picked out as the test data set ( $TDS : (\mathbf{x}^*, y^*)$ ) and the others are used as the modeling data set ( $MDS : (\mathbf{x}, y)$ ), where  $\mathbf{x}^*$  and  $\mathbf{x}$  are the input vectors and  $y^*$  and  $y$  are the corresponding target values.
- 2) Optimize the hyperparameter. For each covariance function, based on the MDS, the hyperparameters that can maximize the log likelihood can be found.
- 3) Predict with the TDS inputs. Substitute the inputs  $\mathbf{x}^*$  in TDS into the generated model and predict the mean and variance of  $y^*$ . Then calculate the root mean square error of TDS inputs.
- 4) Obtain the performance indices.
- 5) Repeat 2) to 5) until all the candidate covariance function is tested.
- 6) Verify and validate. Compare the performance of each covariance function and pick out the best one.

As we know, choosing a covariance function for applications comprises both setting of hyperparameters within a family, and comparing across different families. An appropriate covariance function should meet the following conditions.

- 1) Fit the data set well. The model is used to learn the latent function behind the data set, thus it should fit the data set well, which means the likelihood of the covariance function is big.
- 2) Small prediction error. The model imitate the behavior of the latent function  $f(\mathbf{x})$ . Therefore, the model should not only fit the modeling data, but also fit the new inputs and remain small prediction error.
- 3) Stable. When utilizing subset of the modeling data, it

- should generate similar results.
- 4) Simple form with less parameters. According to the Occam's Razor law [19]. If two covariance function behaves identical, it is preferred to use the simpler one which can also reduce the computation complexity.

As mentioned above, when given a data set  $(\mathbf{x}, y)$ , a covariance function, the following performance indices can be formulated to evaluate the performance of the model:

- 1) Marginal Log Likelihood. The hyperparameters  $\theta$  in a covariance function  $H$  can be optimized by maximizing the marginal log likelihood:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \log p(y|\mathbf{x}, H, \theta) \quad (6)$$

where Gaussian assumption makes it possible to derive the analytical solution of the log marginal likelihood,

$$\begin{aligned} \log p(y|\mathbf{x}, H, \theta) &= -\frac{1}{2}y^T(K + \sigma_n^2 I)^{-1}y \\ &- \frac{1}{2}\log|K + \sigma_n^2 I| - \frac{n}{2}\log 2\pi \end{aligned} \quad (7)$$

Hence the performance index is formulated as:  $L_{fit} = -\log p(y|\mathbf{x}, H, \theta)$

- 2) The Mean Predictive Log Likelihood (MPLL). As GPR produces a Gaussian predictive density, we can obtain

$$-\log p(y^*|\mathbf{x}^*) = \frac{1}{2}\log(2\pi\sigma_*^2) + \frac{(y^* - \bar{f}(\mathbf{x}^*))^2}{2\sigma_*^2} \quad (8)$$

as log loss to verify the method, i.e.,  $E_{MPLL}$ . Here  $\bar{f}(\mathbf{x}^*)$  is the target at each test point,  $\sigma_*^2$  is the predictive variance.

- 3) Root Mean Square Error (RMSE). it can be calculated by using the model data set  $(\mathbf{x}, y)$  and fitting data set  $(\mathbf{x}^*, y^*)$ , i.e.,  $E_{RMSE} = RMSE(y^* - y)$ .

The performance indices of different covariance functions can be used to explore the best covariance function by formulating the following multi-objective function problem [20]:

$$\min F = w_1 \frac{L_{fit}}{\bar{L}_{fit}} + w_2 \frac{E_{MPLL}}{\bar{E}_{MPLL}} + w_3 \frac{E_{RMSE}}{\bar{E}_{RMSE}} \quad (9)$$

where  $w = \{w_1, w_2, w_3\}$  is the weight vector,  $\bar{L}_{fit}$ ,  $\bar{E}_{MPLL}$  and  $\bar{E}_{RMSE}$  are the mean of performance indices.

By finding the minimum  $F$  value using all covariance functions, the best covariance function can be obtained and used for online parameter optimization. According to the algorithm, we can see that the generated models are different. Thus the proposed multi-objective optimization problem should be implemented to investigate the best one. Once a set of optimal parameters are found, the model can be used to predict the output  $y^*$  with the new input  $\mathbf{x}^*$ .

## IV. EXPERIMENTAL RESULTS

### A. Experimental System

To demonstrate the efficiency of the proposed method, experiments were performed using a high precision valve body assembly process. The experimental system as shown in Figure 3 consists of an ABB IRB140 robot with an IRC5 controller, a

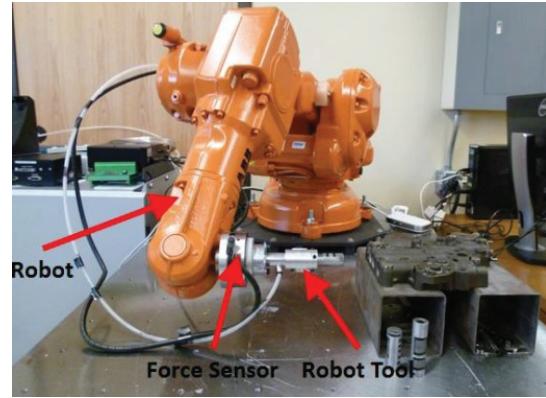


Fig. 3. Experimental system.

force sensor mounted on the robot end effector and a vacuum suction tool used to pick up the valve. The clearance between the peg and the hole is  $40\mu m$  which exceeds the motion accuracy of the industrial robot.

A force sensor and a search algorithm are needed to perform the assembly task and to find the real hole position. The robot tool starts from a random position and contacts with the workpiece and moves along a pattern. Once the tool moves to the right position, the tool can move forward. The movement is monitored and the search process terminates when the search process fails.

A computer is connected to the robot controller via ethernet. ABB force control package is used to perform the assembly process.

### B. Combination of Different Covariance Functions

The GPR has specific requirement for the form of covariance function. As we know, much of the machine learning work has used a very limited set of covariance function. And most of the researches use Squared Exponential Functions(SE) as the covariance function. In this paper, in addition to SE, Rational Quadratic function(RQ), Gaussian White Noise Function(GN), covariance function for a Constant Function(GC) and their combinations are considered and listed in Table II.

TABLE II  
DIFFERENT COMBINATION OF COVARIANCE FUNCTION

No.	Covariance Function	Equation
1	$SE + GN$	$K_{SE}(\mathbf{x}, \mathbf{x}') + K_{GN}(\mathbf{x}, \mathbf{x}')$
2	$RQ + GN$	$K_{RQ}(\mathbf{x}, \mathbf{x}') + K_{GN}(\mathbf{x}, \mathbf{x}')$
3	$GC + RQ + GN$	$K_{GC}(\mathbf{x}, \mathbf{x}') + K_{RQ}(\mathbf{x}, \mathbf{x}') + K_{GN}(\mathbf{x}, \mathbf{x}')$
4	$GC + SE + GN$	$K_{GC}(\mathbf{x}, \mathbf{x}') + K_{SE}(\mathbf{x}, \mathbf{x}') + K_{GN}(\mathbf{x}, \mathbf{x}')$
5	$SE \times (RQ + GN)$	$K_{SE}(\mathbf{x}, \mathbf{x}') (K_{RQ}(\mathbf{x}, \mathbf{x}') + K_{GN}(\mathbf{x}, \mathbf{x}'))$
6	$RQ + SE \times GN$	$K_{RQ}(\mathbf{x}, \mathbf{x}') + K_{GN}(\mathbf{x}, \mathbf{x}') K_{SE}(\mathbf{x}, \mathbf{x}')$

Changing the form of covariance function, we can identify the most suitable covariance function for an assembly process by training the GPR Model.

### C. Experimental Setup

The search process is complicated and a lot of noises and errors exist in the robot system. Therefore, it is difficult to

build the first principle model. Here the goal is to model the relations between the insertion force(IF), search parameters (search force and search speed) and the cycle time(time to find the assembly process). Search force(SF) and search speed(SS) are within the range (5-35) and (250-350) respectively as listed in Table III, where  $\Delta x$ ,  $\Delta y$ ,  $\Delta\phi_x$  and  $\Delta\phi_y$  are the noise of the system.

We use the design-of-experiment (DOE) method, of which the most sensitive parameters are chosen and tuned carefully after performing a series of experiments, to obtain the original data set. The DOE was performed offline for comparison and 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 4 times for each set of parameters, so totally 108 experiments were conducted. The data set is divided into 4 subsets and thus in each iteration, 81 samples are used to build the model and 27 randomly selected samples are used to test the result of the model.

For the robotic assembly process, it is desired that the cycle time and its variance are small. After collecting the needed data set using DOE method, we are able to train the GPR with different covariance functions to get the specific hyperparameters, which can be utilized to predict the output with totally new input. Calculating and comparing the performance indices of difference covariance functions, we can obtain the optimal model structure that fits the assembly process mostly.

TABLE III  
PARAMETER CONFIGURATIONS.

SS(mm/s)	SF(mm/s)	IF(mm/s)	$\Delta x$	$\Delta y$	$\Delta\phi_x(^{\circ})$	$\Delta\phi_y(^{\circ})$
250:50:350	5:15:35	50:25:100	-9±1	±1	±0.6	±0.2

#### D. Experimental Result

Through analysis of variance (ANOVA), insertion force is the least importance parameter that influences the performance of the assembly process. For easy visualization, only search force and search speed vs cycle time are plotted. The predicted 3D spatial curved surfaces with test data set using different combinations of covariance functions in Table II are showed in Figure 5.

If we plot the original data on the three-dimensional space as shown in Figure 4, we can see that it is not evenly distributed. That is why models with similar likelihood in Figure 5 may have great difference.

TABLE IV  
COMPARISON OF EXPERIMENTAL RESULTS IN FIGURE 5.

No.	SF(N)	SS(mm/s)	$-L_{fit}$	$E_{MPLL}$	$E_{RMSE}$	F
a	21	280	123.12	-1.887	1.22	2.96
b	22	252	140.07	-1.333	0.794	2.93
c	24	258	136.66	-1.332	0.794	3.01
d	24	258	136.55	-1.342	0.800	3.19
e	24	278	134.73	-1.338	0.810	3.23
f	22	250	138.19	-1.332	0.796	2.57

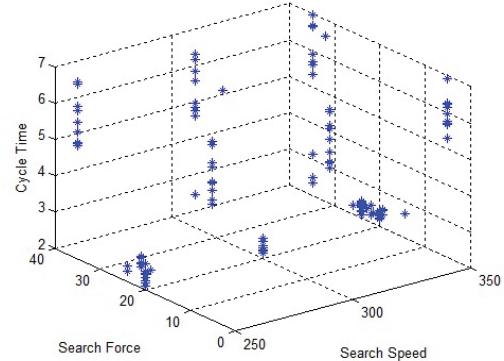


Fig. 4. 3D Data Points.

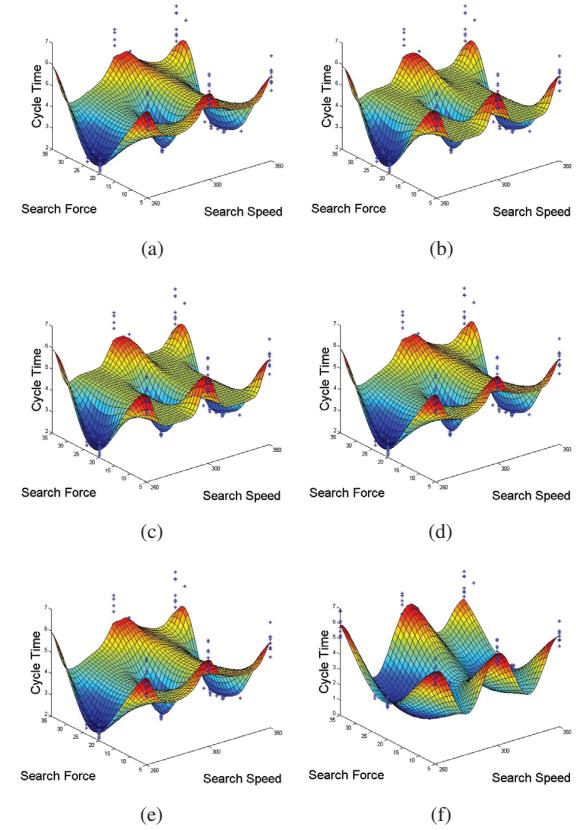


Fig. 5. Generated modeling using different covariance function in Table II  
a) 1, b) 2, c) 3, d) 4, e) 5, f) 6

The statistical results, such as the search parameters for each covariance function using MDS shown in Figure 5 are listed in Table IV. The calculated  $L_{fit}$ ,  $E_{MPLL}$ ,  $E_{RMSE}$  and F values are compared to find the most suitable covariance function for the assembly process.

If all of the weights are positive, as assumed in this study, weights are selected with equal importance(i.e.  $w_1 = w_2 = w_3 = 1$ ). Since the model using  $RQ + SE \times GN$  covariance function has the minimum F value,  $RQ + SE \times GN$  function is identified as the most suitable covariance function. The

TABLE V  
EXPERIMENTAL RESULTS

SS	SF	IF	Tr( $\mu$ )	Tc( $\mu$ )	$\sigma$	$\mu-\sigma$	$\mu+\sigma$
250	35	50	6.2	5.9	0.57	6.49	5.35
300	35	50	5.4	5.4	0.57	6.05	4.91
350	35	50	3.1	3.1	0.56	3.69	2.55
250	20	50	2.4	2.4	0.56	2.97	1.83
300	20	50	2.4	2.3	0.57	2.92	1.78
350	20	50	2.3	2.3	0.55	2.90	1.79
250	5	50	6.4	6.0	0.57	6.65	5.51
300	5	50	5.7	6.0	0.57	6.62	5.48
350	5	50	6.2	5.9	0.57	6.53	5.39
250	35	75	5.5	5.9	0.57	6.49	5.35
300	35	75	5.2	5.4	0.57	6.05	4.91
350	35	75	4.6	5.1	0.56	5.69	4.55
250	20	75	2.3	2.4	0.56	2.97	1.83
300	20	75	2.2	2.3	0.57	2.92	1.78
350	20	75	2.3	2.3	0.55	2.90	1.79
250	5	75	6.2	6.0	0.57	6.65	5.51
300	5	75	5.7	6.0	0.57	6.62	5.48
350	5	75	5.7	5.9	0.57	6.53	5.39
250	35	100	5.4	5.9	0.57	6.49	5.35
300	35	100	5.2	5.4	0.57	6.05	4.91
350	35	100	5.6	5.1	0.56	5.69	4.55
250	20	100	2.5	2.4	0.56	2.97	1.83
300	20	100	2.4	2.3	0.57	2.92	1.78
350	20	100	2.3	2.3	0.55	2.90	1.79
250	5	100	6.8	6.0	0.57	6.65	5.51
300	5	100	6.5	6.0	0.57	6.62	5.48
350	5	100	5.6	5.9	0.57	6.53	5.39

proposed optimization problem is implemented. A set of optimal parameters are selected with the best performance and the corresponding hyperparameters are calculated by the GPR method.

To test the performance of the model using  $RQ + SE \times GN$  covariance function, the experimental results utilizing the TDS are listed in the Table V.  $Tr(\mu)$  means the real assembly cycle time,  $Tc(\mu)$  means the calculated cycle time and  $\sigma$  is the calculated standard deviation using trained GPR. From Table V, we can see that the cycle time predicted by GPR model is correct at every input data set.

## V. CONCLUSION

The paper proposes a Gaussian Process Regression method to solve the covariance function selection problem in high precision robotic assembly processes. The original Gaussian Process Regression algorithm is improved to effectively identify the proper covariance functions to reduce the cycle time. The most suitable covariance function is found for an assembly process through combinations of some basic covariance functions. The proposed modeling method can be applied online without stopping an assembly station. This is our first attempt to model complex assemble process that will make online process optimization possible and generate huge economic impact. Our future work will focus on applying Gaussian Process Regression to torque converter assembly process utilizing online parameter optimization method, balancing the exploration and exploitation to minimize the cycle time and maximize the First Time Through rate.

## 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] 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," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3251–3256, 2005.
- [3] S. O. Tasan and S. Tunali, "A review of the current applications of genetic algorithms in assembly line balancing," *Journal of intelligent manufacturing*, vol. 19, no. 1, pp. 49–69, 2008.
- [4] J. A. Marvel and W. S. Newman, "Accelerating robotic assembly parameter optimization through the generation of internal models," *IEEE International Conference on Technologies for Practical Robot Applications*, pp. 42–47, 2009.
- [5] C. R. C. Williams, "Gaussian processes for regression," *Advances in Neural Information Processing Systems*, vol. 8, no. 2, 1996.
- [6] E. N. S. Vasudevan, F. Ramos and H. Durrant-Whyte, "Gaussian process modeling of large-scale terrain," *Journal of Field Robotics*, vol. 26, no. 10, pp. 812–840, 2009.
- [7] B. B. J. Kocijan, A. Girard and R. Murray-Smith, "Dynamic systems identification with gaussian processes," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 11, no. 4, pp. 411–424, 2005.
- [8] G. Gregoric and G. Lightbody, "Gaussian process approach for modelling of nonlinear systems," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 4-5, pp. 522–533, 2009.
- [9] B. Frank, C. Stachniss, N. Abdo, and W. Burgard, "Using gaussian process regression for efficient motion planning in environments with deformable objects," in *Automated Action Planning for Autonomous Mobile Robots*, 2011.
- [10] J. Kho, A. Rogers, and N. R. Jennings, "Decentralized control of adaptive sampling in wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 3, p. 19, 2009.
- [11] D. M. Morris, R. Hebbar, and W. S. Newman, "Force guided assemblies using a novel parallel manipulator," vol. 1, pp. 325–330, 2001.
- [12] N. Bulygina, "Model structure estimation and correction through data assimilation," Ph.D. dissertation, The University of Arizona, 2007.
- [13] C. E. Rasmussen, "Gaussian processes in machine learning," in *Advanced Lectures on Machine Learning*. Springer, 2004, pp. 63–71.
- [14] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, "Gaussian process modeling of large-scale terrain," *Journal of Field Robotics*, vol. 26, no. 10, pp. 812–840, 2009.
- [15] M. Schneider and W. Ertel, "Robot learning by demonstration with local gaussian process regression," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 255–260, 2010.
- [16] K. Yang, S. Keat Gan, and S. Sukkarieh, "A gaussian process-based rrt planner for the exploration of an unknown and cluttered environment with a uav," *Advanced Robotics*, vol. 27, no. 6, pp. 431–443, 2013.
- [17] W. S. N. Jing Wei, "Improving robotic assembly performance through autonomous exploration," *Robotics and Automation*, vol. 3, pp. 3303–3308, 2002.
- [18] M. Jadaliha, Y. Xu, J. Choi, N. Johnson, and W. Li, "Gaussian process regression for sensor networks under localization uncertainty," *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pp. 223–237, 2013.
- [19] M. C. Amy Courtney, "Comments regarding on the nature of science," *Physics in Canada*, vol. 64, no. 3, pp. 7–8, 2012.
- [20] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Reliability Engineering & System Safety*, vol. 91, no. 9, pp. 992–1007, 2006.