

# Anomaly Detection for Insertion Tasks in Robotic Assembly Using Gaussian Process Models

Romeres, Diego; Jha, Devesh K.; Dau, Hoang; Yerazunis, William S.; Nikovski, Daniel N.

TR2019-055 June 29, 2019

## Abstract

Component insertion is a common task in robotic assembly, and is widely used for manufacturing a variety of electronic devices. This task is generally characterized by low tolerances, thus requiring high precision during assembly. An early detection of a fault in the mating during the insertion process enables quality control of the end products, as well as safeguards the robotic equipment. We propose to use Gaussian Process Regression-based methods to learn the force profile during successful insertions, as well as quantify permissible deviations from this profile. The GPR model is then used to detect anomalies in case the observed force profile deviates significantly from the expected range. Apart from the standard GPR formulation, we consider two other variants – the Heteroscedastic GPR and the local GPR for better modeling accuracy and computational time efficiency, respectively. We report an accuracy of 100% in differentiating between normal and faulty insertions. The modeling and detection results indicate that our approach is accurate and robust to severe uncertainties due to process (e.g., force drift) and measurement noise.

*European Control Conference (ECC)*

© 2019 MERL. This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.



# Anomaly Detection for Insertion Tasks in Robotic Assembly Using Gaussian Process Models

Diego Romeres<sup>1</sup>, Devesh K. Jha<sup>1</sup>, Hoang Anh Dau<sup>2</sup>, William Yezunis<sup>1</sup>, Daniel Nikovski<sup>1</sup>

**Abstract**—Component insertion is a common task in robotic assembly, and is widely used for manufacturing a variety of electronic devices. This task is generally characterized by low tolerances, thus requiring high precision during assembly. An early detection of a fault in the mating during the insertion process enables quality control of the end products, as well as safeguards the robotic equipment. We propose to use Gaussian Process Regression-based methods to learn the force profile during successful insertions, as well as quantify permissible deviations from this profile. The GPR model is then used to detect anomalies in case the observed force profile deviates significantly from the expected range. Apart from the standard GPR formulation, we consider two other variants – the Heteroscedastic GPR and the local GPR for better modeling accuracy and computational time efficiency, respectively. We report an accuracy of 100% in differentiating between normal and faulty insertions. The modeling and detection results indicate that our approach is accurate and robust to severe uncertainties due to process (e.g., force drift) and measurement noise.

## I. INTRODUCTION

Improvements in actuation, sensing, and computing capabilities have allowed robots to play an increasingly important role in automated assembly [1], [2]. Robotic assembly is widely used in many industrial areas such as the manufacturing of automobiles, mobile phones, and even delicate components such as surface-mount circuit boards. The robot typically operates in a repetitive fashion, following a sequence of operations to assemble products from a set of parts. However, these predefined actions generally do not take into account the uncertainty and imprecision in robots behavior.

Even though industrial robots are highly reliable, and abnormal behaviors are rare, the consequences of abnormalities can be extremely costly. Most of the parts that are assembled in robotic assembly have small tolerances, and an abnormal operation can result in assembly line downtime, damage to the manufactured device or to the robot itself. Thus, timely detection and recovery from these errors is essential for assembly automation. In this paper, we present experiments on robotic insertion assemblies. However, our approach could be applied just as well to other trajectory-centric automation processes. Our particular assembly task is connector mating, and is ubiquitous in manufacturing. Fig. 1 shows an example setup for insertion task. The male part of the connector is held fixed on a surface. The female part of the connector

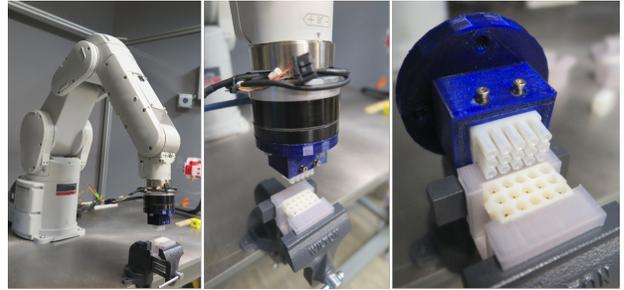


Fig. 1. Experiment setup. left) the Mitsubishi Electric RV-4FL model robot, middle) 15-pin Molex connector and its fixture, right) a close-up view of connector blocks

is attached to the end of a robot arm. Successful insertions, which consist in fully mating the two connectors, rely on precision in the connector approach path, the connector grasp, and the accuracy of the connector housing and electrical contacts. Unknown nonlinearities, such as tool and grip dynamics, complex friction inside the connector, and environmental factors make it challenging to model the force profile using rigid dynamics. An additional interesting phenomenon that we observed after an extended period of data collection is the presence of force drift in consecutive normal runs, which will be explained in Section V-B.

We propose to combine the expressiveness of the data-driven approach with a probabilistic technique to model the force profile along the trajectory of the robot during a normal mating process. We verified that the insertion force applied by the robot is a well-behaved function of the vertical distance between the parts being mated. We model the insertion force as a probabilistic function of the end-effector’s vertical positions, and then use the confidence interval obtained from the probabilistic map to detect anomalies.

**Contributions.** First, we propose a general-purpose anomaly detection algorithm for the problem of insertion-type assembly using Gaussian Process Regression-based models (GPR) [3]. Second, we test the proposed algorithm using a real robotic test bed with an industrial grade robot under the presence of several uncertainties, such as force drift due to repeated long-term usage of the robot. Third, the GPR-based detection method is compared against several competing machine learning techniques, such as random forests and gradient-boosting regression.

**Related Work.** Anomaly detection in assembly or robotic manipulation kind of problems has received active interest because of the ubiquity of robotic assembly [4]. Two main types of approaches can be distinguished: model-based and data-driven. The model-based approach attempts to create a model using an accurate physical knowledge of the pro-

<sup>1</sup>Diego Romeres, Devesh Jha, William Yezunis and Daniel Nikovski are with Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139 {romeres, jha, yezunis, nikovski}@merl.com

<sup>2</sup>Hoang Anh Dau is with University of California at Riverside hdau001@ucr.edu

cess [5], [6]. Furthermore, creating physics-based models requires substantial domain knowledge, which might not be feasible for many complex applications. Some examples for these approaches could be found in [7], [8], [9], which model the force profile by means of physical models, and then declare an anomaly based on any deviation from the model. However, it is not clear how difficult it would be to systematically model the noise and uncertainty that inevitably appear during deployment. For these reasons, many prefer a data-driven approach, either using a classification or regression approach to model the process. On the other hand, Gaussian processes have been used for anomaly detection in many different applications, e.g. videos [10], surveillance [11], prognostics [12], etc. for its non-parametric property and ability to model temporal correlations. However, they have never been used for insertion-type assembly problems.

Furthermore, for the best of our knowledge, none of the existing methods explicitly considers the uncertainties associated with the mating process, and thus these methods are not robust to process noise. The data-driven approach is based on data to model successful and faulty insertion runs. Perhaps the simplest and most widely used technique based on this approach is threshold-based anomaly detection, where a mating is classified as faulty if the instantaneous force or torque exceeds the maximum values observed in the past. Despite being a simple and effective in some cases, this method does not generalize well to different settings.

## II. BACKGROUND

In this section, we briefly describe Gaussian Process Regression (GPR) and two of its variations namely, Heteroscedastic GPR (HGPR) and Local GPR (LGPR).

### A. Gaussian Processes

A Gaussian process (GP) is completely specified by its mean and covariance function [13]. We define the mean function  $m(\mathbf{x})$  and covariance function  $k(\mathbf{x}, \mathbf{x}')$  of a real process  $f(\mathbf{x})$  as follows:

$$\begin{aligned} m(\mathbf{x}) &= E[f(\mathbf{x})] \\ k(\mathbf{x}, \mathbf{x}') &= E[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))] \end{aligned}$$

and the Gaussian process is written as follows:

$$f(\mathbf{x}) \sim \mathcal{N}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \quad (1)$$

The GP predictive distribution at a test point  $\mathbf{x}^*$  is given by

$$p_{GP}(f(\mathbf{x}^*)|\mathcal{D}, \mathbf{x}^*) = \mathcal{N}(\mu(\mathbf{x}^*), \Sigma(\mathbf{x}^*)) \quad (2)$$

$$\mu(\mathbf{x}^*) = \mathbf{k}_*^T (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{Y} \quad (3)$$

$$\Sigma(\mathbf{x}^*) = \mathbf{k}_{**} - \mathbf{k}_*^T (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_* \quad (4)$$

where  $\mathcal{D} = \{\mathbf{X}, \mathbf{Y}\}$  is the training data,  $\mathbf{K}$  is the kernel matrix, where the individual elements are given by  $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ ,  $\mathbf{k}_{**} = k(\mathbf{x}_*, \mathbf{x}_*)$ ,  $\mathbf{k}_* = k(\mathbf{X}, \mathbf{x}^*)$  and  $\sigma^2$  is the measurement noise covariance. There are several choices for the kernel function  $k$  which are commonly used in literature. The inference problem in GP is to learn the parameters of

the kernel function, called hyperparameters, maximizing the log marginal likelihood [13]. A Gaussian process is a non-parametric technique for modeling probabilistic relationships known for its flexibility and accuracy to infer data. One of its main drawbacks is that the computational cost scales as  $\mathcal{O}(N^3)$ , where  $N$  is the number of data points used to train the GP. Thus, it is relatively computationally intensive.

### B. Local Gaussian Process Regression

Local Gaussian Process Regression was introduced in [14] and it consists of a local approximation of GPR with the aim to speed up the training and prediction process.

In the training phase this method consists on clustering the training data set using for example a K-mean algorithm and then a GP is learned for each cluster with the procedure describe in Section II-A. Since each cluster has a reduced number of data the computational time is lower. In the test phase of the method, the proximity between any query point and all the cluster centroids is computed and used as the weight associated to the local model. A kernel function, e.g., the RBF kernel can be used to define this proximity metric. Finally, the LGP posterior distribution can be computed as the weighted average of local models' prediction.

$$p_{LGP}(f(\mathbf{x}^*)|\mathcal{D}, \mathbf{x}^*) = \mathcal{N}(\mu_{LGP}(\mathbf{x}^*), \Sigma_{LGP}(\mathbf{x}^*)) \quad (5)$$

$$\mu_{LGP}(\mathbf{x}^*) = \frac{\sum_{i=1}^M w_i \mu_i(\mathbf{x}^*)}{\sum_{i=1}^M w_i} \quad (6)$$

$$\Sigma_{LGP}(\mathbf{x}^*) = \frac{\sum_{i=1}^M w_i^2 \Sigma_i(\mathbf{x}^*)}{\sum_{i=1}^M w_i^2} \quad (7)$$

where  $M$  is the number of local models,  $w_i$  is the weight of model  $i$  given by the proximity metric and,  $\mu_i(\mathbf{x}^*)$ ,  $\Sigma_i(\mathbf{x}^*)$  are the posterior mean and covariance of the  $i^{th}$  model.

### C. Heteroscedastic Gaussian Processes

The Heteroscedastic Gaussian Process (HGP) is a GP in which the noise model is input data dependent [15], [16], [17]. This gives more degree of freedom to the model to characterize more accurately the data. In particular, the model of the noise becomes  $\epsilon(\mathbf{x}) \sim \mathcal{N}(0, \sigma_{HGP}^2(\mathbf{x}))$ .

Solving the full HGPR problem could be computationally expensive, since the number of hyperparameters to optimize might depend on the number of input data. For this reason, parametrizations of the covariance  $\sigma_{HGP}^2(\mathbf{x})$  are used. In this work, we consider that the noise levels are clustered in  $M \ll N$  clusters which corresponds to  $M$  noise levels  $\sigma_i^2$  with  $i \in \{1, \dots, M\}$  and  $N$  is the number of data points.

The noise variance of each input data point is the weighted average of the noise level of each cluster.

$$\sigma_{HGP}^2(\mathbf{x}) = \frac{\sum_{i=1}^M w_i(\mathbf{x}) \sigma_i^2}{\sum_{i=1}^M w_i(\mathbf{x})} \quad (8)$$

where  $M$  is the total number of clusters,  $w_i = \exp\{-0.5(\mathbf{x} - c_i)^T C(\mathbf{x} - c_i)\}$  is the weight of the  $i^{th}$  cluster,  $\sigma_i^2$  is the noise level of the  $i^{th}$  cluster. The noise levels of each cluster and the length scales  $C$  are hyperparameters

of the heteroscedastic kernel, and thus can be optimized by standard marginal log-likelihood maximization.

The posterior distribution obtained with heteroscedastic noise is like in the standard GPR (2) with noise covariance defined in (8). Which at a test point  $\mathbf{x}^*$  is given by

$$p_{HGPR}(f(\mathbf{x}^*)|\mathcal{D}, \mathbf{x}^*) = \mathcal{N}(\mu(\mathbf{x}^*), \Sigma(\mathbf{x}^*)) \quad (9)$$

$$\mu(\mathbf{x}^*) = \mathbf{k}_*^T (\mathbf{K} + \Sigma_{HGPR})^{-1} \mathbf{Y} \quad (10)$$

$$\Sigma(\mathbf{x}^*) = k_{**} - \mathbf{k}_*^T (\mathbf{K} + \Sigma_{HGPR})^{-1} \mathbf{k}_* \quad (11)$$

where  $\Sigma_{HGPR} := \text{diag}(\sigma_{HGPR}^2(\mathbf{x}_1), \dots, \sigma_{HGPR}^2(\mathbf{x}_N))$  is the noise covariance matrix.

### III. PROBLEM FORMULATION

The objective of this work is to detect normal and abnormal runs in insertion-type robotic assembly processes. Runs are executed multiple times during the process. A run is assumed to consist of one vertical insertion actuated by a robotic arm, starting from a predetermined position. Each run is described by a data set, i.e., the  $i$ -th run consists of  $\{X^i, Y^i\}$  where  $X^i := [x_1^i, \dots, x_r^i]$  are the vertical  $z$ -positions of the end effector of the robot,  $Y^i := [y_1^i, \dots, y_r^i]$  are the force sensor measurements and  $r \in \mathbb{R}$  is the number of sampled points in the run. Given a training data set of normal insertion runs  $\{X, Y\}$  defined by stacking the data sets  $X^i, Y^i$  of  $n$  runs in  $X$  and  $Y$ , respectively, we assume that there exists a probabilistic model such that

$$Y = f(X) + \epsilon \quad (12)$$

where  $\epsilon$  is an additive independent Gaussian noise with zero mean and covariance  $\sigma^2$ , and  $f$  is an unknown function. We want to determine a probabilistic estimator  $\hat{f}$  of  $f$ , such that an appropriate anomaly measure can be defined with the help of the posterior distribution  $p(f|\mathcal{D})$ , in order to judge any unseen runs to be either normal or abnormal.

### IV. ANOMALY DETECTION

In this section, we present the proposed approach to solve the anomaly detection problem described in Section III.

In order to estimate the probabilistic model in (12), we follow the three paradigms described in Section II which belong all to the GPR framework, and therefore for each of them, the posterior distribution is known and Gaussian (or locally Gaussian for LGP) when the hyperparameters are fixed to their marginal likelihood estimate. Hence, the confidence intervals of the final estimator are described by the estimate of the posterior density  $p(f|\mathcal{D})$ , and are ellipsoidal sets. In the case of GPR described in Section II-A, the confidence sets are defined for any input point  $x^*$  by

$$\mathcal{E}^\alpha(x^*) = \{h \in \mathcal{R} : (h - \mu(x^*))\Sigma^{-1}(x^*)(h - \mu(x^*)) < \chi_\alpha^2\} \quad (13)$$

where  $\mu(\cdot)$  and  $\Sigma(\cdot)$  are defined in (3) and (4), respectively. For a fixed probability level  $\alpha$ ,  $\chi_\alpha^2$  is the value for which  $Pr(\chi^2 < \chi_\alpha^2) = \alpha$ .  $\mathcal{E}^\alpha$  defines the region in which a sample from  $p(f|\mathcal{D})$  will end up with probability  $\alpha$ . Analogous expressions can be defined for the estimators in Sec. II-B, II-C.

After a posterior distribution  $p(f|\mathcal{D})$  is estimated in a training data set composed only of normal runs, the anomaly detection algorithm computes a score for each data point  $x_i^j$  of any new run  $j$  based on whether the prediction  $\hat{f}(x_i^j)$  falls within the expected confidence interval  $\mathcal{E}^\alpha(x_i^j)$ . The scores for each data point  $i \in \{1, \dots, r\}$  are then combined to assign one final anomaly score to the entire run. Finally a threshold decides whether the run is normal or abnormal as described in Algorithm 1. Algorithm 1 takes as inputs the

---

#### Algorithm 1 Algorithm for Anomaly Detection

---

**Input:** run =  $\{X^i, Y^i\}$ ,  $\mathcal{D}$  training dataset,  $p(f|\mathcal{D})$ ,  $\alpha$ , thr

**Output:** anomaly\_flag

*Initialisation :*

- 1: anomaly\_flag = 0, anomaly\_score = 0
  - 2: **for**  $j = 0$  to  $r$  **do**
  - 3:   Compute  $\hat{f}(x_j^i)$
  - 4:   Compute  $\mathcal{E}^\alpha(x_j^i)$
  - 5:   **if**  $(\hat{f}(x_j^i) \notin \mathcal{E}^\alpha(x_j^i))$  **then**
  - 6:     anomaly\_score += 1
  - 7:   **end if**
  - 8: **end for**
  - 9: anomaly\_score = anomaly\_score/ $r$ ,    $r = |X^i|$
  - 10: **if** (anomaly\_score  $\geq$  thr) **then**
  - 11:   anomaly\_flag = 1
  - 12: **end if**
  - 13: **return** anomaly\_flag
- 

data points of the run that has to be detected, the probability of inclusion of the confidence set  $\alpha$ , the threshold thr and the posterior distribution  $p(f|\mathcal{D})$  estimated on a training data set  $\mathcal{D}$  of all normal runs and computed with any of the methods described in Section II or any other probabilistic modeling technique. Step 3 might be computed using the posterior means (2), (5), (9) and Step 4 might be computed using Eq. (13) and one of the posterior means (3),(6),(10) and posterior covariances (4),(7),(11). The percentage of inclusion of the confidence interval,  $\alpha$ , is set at 99.9%. The anomaly score, ‘anomaly\_score’ and the choice of the threshold, ‘thr’ are described in the following section.

#### A. Binary Anomaly Score

Given a predictive posterior distribution it is possible to compute an anomaly score for each individual observation in a run, which means that an anomaly score will be associated to any force sensor measure at any given  $z$ -position location. This will give us an informative picture of the machine health during the mating process. To get the final anomaly score of an entire run, we average the anomaly scores of each individual observations within that run. A higher anomaly score means higher abnormality. With an appropriate threshold value, we can turn this anomaly score into an anomaly label (normal or abnormal). The anomaly score for each data point is based on whether the prediction falls within the expected confidence interval (13). Considering the properties of GPR and the uncertainty level  $\alpha = 99.9\%$  and given a

pair of points  $\{x^*, \hat{f}(x^*)\}$  the anomaly score for each data can be computed as:

$$\text{anomaly\_score}(x^*, \hat{f}(x^*)) = \begin{cases} 0, & \text{if } (\mu(x^*) - 3.29\sigma(x^*)) \leq y^* \leq (\mu(x^*) + 3.29\sigma(x^*)) \\ 1, & \text{otherwise} \end{cases}$$

After the anomaly score is defined we need to define a threshold in order to declare when a run is normal or abnormal. A standard and informative way to evaluate the performance of an anomaly detection method is to compute its Receiver Operating Characteristic (ROC) curve. The ROC curve is created by plotting the true positive rate vs. the false positive rate at various threshold values on the anomaly score. In this way it is possible to analyze and the evaluate the performance of several methods at a different values of threshold and choose a best threshold value.

## V. EXPERIMENTS AND DATA ANALYSIS

In order to verify the accuracy of Algorithm 1, a robotic arm was used to automatically insert a Molex 3x5 0.250" 15-pin connector repeatedly, while gathering precise force and position information, see Figure 1.

### A. Data Gathering

Force was measured by a six-axis load cell directly behind the connector mount on the robot, while position was measured by the robot's built-in encoders. The connectors were held fixed by 3D-printed fixtures; the moving (female) connector is secured by 3mm capscrews, while the male fixture is C-clamped to the steel robot platform.

Each data acquisition trial starts with loading a new connector set. Then the robot is manually steered to complete the connector mating insertion. By monitoring the real-time forces from the load cell, the connector is centered on the location which requires minimum force while mated, i.e., the location which has the least contacts between the walls of the connectors. This position is defined as the "fully mated" (FM) position. The robot and moving connector is then withdrawn vertically in the  $z^+$  direction, and this second position is defined as "prepare to mate" (PM) position.

A *normal run* is defined by the following sequence of movements: move the robot to the PM position, start logging  $z$ -positions and load cell forces measurements, wait a short amount of time, then move to the FM position, wait for a short amount of time, then stop logging. Then the robot moves upward back to the PM position.

An *abnormal run* is defined as a normal run except the PM position is shifted from its original position by an offset. The robot is then commanded to the FM position with a diagonal motion causing the two connectors surfaces to hit and slide against each other, then finally completing the insertion.

Two data sets obtained from two independent data collections performed on different days are considered, see Fig. 2. The first data set is composed of 4,000 normal insertions collected in an unbroken rapid sequence. In order to limit the computational cost in training we down-sampled this dataset

to 1 every 10 runs, this data set will be called  $\mathcal{D}_{tr}$ . The force profile is nearly flat until the end of the insertion where the two connectors enter in contact and the force rapidly increases. However, notice that there is a drift on the force measurements after each run given by the force sensor drift (this is a known phenomenon). The drift is higher at the end of the insertion with a delta in forces of 16[N] while at the beginning the delta is of 8[N]. The second data set is used

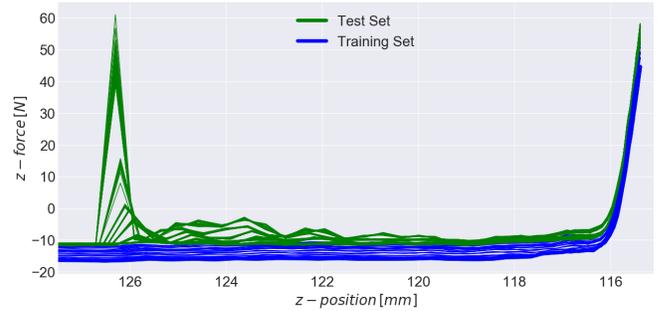


Fig. 2. Force profile of the insertion phase as a function of the end-effector's  $z$  coordinate. In blu there are the training trajectories, normal runs, and in green the test trajectories, a mix of normal and faulty runs. Data was sampled at 140 Hz. Note that the x-axis displays decreasing vertical distances as the robot moves downward to insert two components.

for testing,  $\mathcal{D}_{test}$ , and is composed of 100 normal runs and 600 abnormal runs, also collected in sequence. The abnormal runs are obtained by two type of faults, either repeatedly shifting the PM position in steps of 0.5mm up to 6mm along a direction perpendicular to the  $z$ -axes or repeatedly rotating the PM position in steps of 0.5 degrees up to 4 degree. The misalignment is an important anomaly to detect because it could either damage the part where the connector is hold or break one pin but still performing the full mating. This would cause the final product to fail without realizing it.

### B. Z-force drift

Several experiments have been conducted in different conditions such as changing the temperature of the environment and changing the control mode of the robot. In one condition the data set sequences in our collections presented an interesting challenge: Z-force drift. As can be seen in Fig. 3,

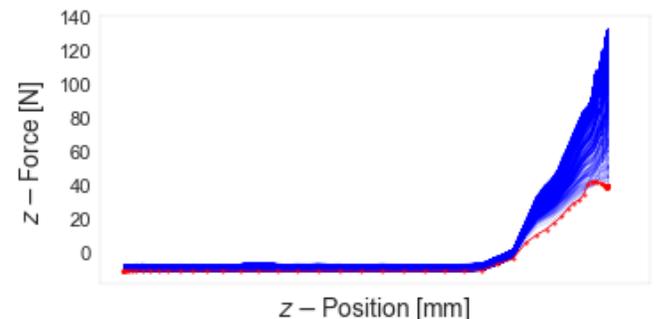


Fig. 3. Force profile of insertion phase as a function of the end-effector's  $z$ -coordinate. The blue lines show 1000 normal runs and the red line highlights the force profile of the first run. Apparent in the plot is force drift, appearing after the first contact of the two connectors.

on top of the known measurement drift given by the sensor, a systematic, although irregular, drift can be observed in the force profile. This causes each normal run in the data set to have a different force profile, and thus even normal runs could rightfully considered to be an abnormal run w.r.t. the prior runs of this sequence.

We investigated the causes of this force drift. Several hypotheses were tested and rejected by means of a set of experiments, including connector wear out, connector expansion due to frictional heating, connector slipping in the mounting clamps, and robot encoder drift during execution. Ultimately, the force drift was empirically proven to be caused by internal thermal expansion of the robot due to the repeated movement, which causes a minor shift in the position of the end effector. This phenomenon corresponds to a misalignment between the two connectors that creates an increase in the frictional forces between the walls of the two connectors during the insertion, and a corresponding increase in the measurements of  $z$ -force. Note in Fig. 3 how the force drift appears only at the  $z$ -position when the two connectors are in contact. We confirmed this with an independent laser positioning sensor ( resolution of  $10\mu\text{m}$ ) taking measurements of the PM position, while the robot is performing normal runs repeatedly. The measurements showed an intermittent and non-negligible drift in the PM position; thermal infrared imaging showed parts of the robot arm reaching  $39\text{ degC}$  versus  $20\text{ degC}$  ambient in the test cell. Also note that the effect of this drift in position on the measured  $z$ -force would be very difficult or impossible to model from physical principles. This finding further favors the use of a data-driven approach instead of physical modeling. For space reasoning we could not show all the analysis for also this case but will be summarized in the end.

### C. Results

In this section, we will analyze the performance in anomaly detection obtained by applying the three posterior distributions described in Section II namely  $p_{GP}(f|\mathcal{D}_{tr})$ ,  $p_{HGP}(f|\mathcal{D}_{tr})$  and  $p_{LGP}(f|\mathcal{D}_{tr})$  to Algorithm 1. For completeness, we will compare these methods with two other standard methods popular in the machine learning literature: Random Forests (RF) see [18] and Gradient Boosting Regression (GBR). The estimator of  $p_{LGP}(f|\mathcal{D}_{tr})$  is computed considering 5 clusters. All models have been trained on the data set  $\mathcal{D}_{tr}$  and tested on the data set  $\mathcal{D}_{test}$ .

The ROC curve described in Section IV-A is used as a first comparison among the methods using the binary anomaly score defined in Section IV-A. Fig. 4 provides a comparison between all tested methods as their threshold values vary. From Fig. 4, it is evident that HGPR and LGPR are the only methods able to correctly separate all the normal and abnormal runs, for some suitable threshold value. For GPR, GBR and RF, no such threshold exists – for these methods, there is always some classification error, either as a false positive, or as a false negative. Fig. 4 shows the same performance for two of the GP-based models, in order to determine which method appears to be most suitable for

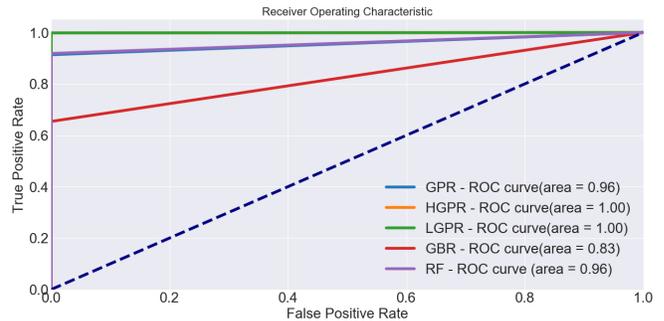


Fig. 4. ROC curve computed for all the proposed models GPR, HGPR and LGPR and compared with other machine learning techniques such as GBR and RF. The curves obtained with GPR, HGPR and LGPR overlap.

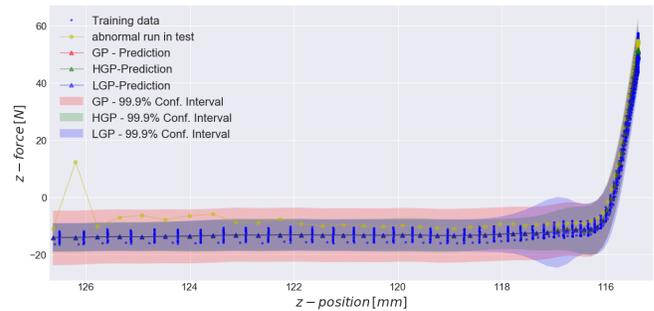


Fig. 5. Comparison of the mean and confidence interval predictors of GPR, HGPR and LGPR. Notice the force drift on the training data (blu) and how HGPR and LGPR estimates accurately the confidence set, while GPR confidence set is too high in the initial phase of the insertion. One faulty run from  $\mathcal{D}_{test}$  is also shown.

anomaly detection problems, we are interested in analyzing the models further by inspecting their confidence sets and the computational time.

In Fig. 5 we compare the prediction accuracy of  $p_{GP}(f|\mathcal{D}_{tr})$ ,  $p_{HGP}(f|\mathcal{D}_{tr})$  and  $p_{LGP}(f|\mathcal{D}_{tr})$  on one single faulty test run. While in terms of predictive mean there are not significant differences between the three methods,  $p_{HGP}(f|\mathcal{D}_{tr})$  and  $p_{LGP}(f|\mathcal{D}_{tr})$  outperform  $p_{GP}(f|\mathcal{D}_{tr})$  in estimating the confidence set. This is because the drift in the force measurements affects the data differently in different regions of the space (small variations in the initial phase of insertion and large variations towards the end) and this translates to different levels of noise from a modeling point of view, i.e., the data is generated from a non stationary process. However,  $p_{GP}(f|\mathcal{D}_{tr})$  does not have the capability of describing different levels of noise yielding a large confidence set (the thick red band in Fig. 5) in the initial part of the insertion. On the other hand,  $p_{HGP}(f|\mathcal{D}_{tr})$  is based on a non-stationary kernel where the noise level depends on the input space and the confidence sets accurately describe the variability of the training data for all  $z$ -positions of the insertion phase. LGPR can also describe accurately the variability given by the  $z$ -force drift because it can assign a different level of noise to each cluster. However, LGPR confidence sets loose accuracy in some area of the insertions e.g. around  $z = 117$  in Fig 5. Indeed, LGPR is an

approximation of HGPR.

Additionally, we tested a down-sampled version of  $\mathcal{D}_{tr}$  called,  $\mathcal{D}_{tr}^{DS}$ , where the runs are down-sampled to 1 run out of every 20 runs. We observed that also in this case the accuracy does not change and the ROC curves are as in Fig. 4. We believe this is because all the runs are a noisy representation of the same process and the sampled points of  $\mathcal{D}_{tr}^{DS}$  are enough to cover the  $z$ -position domain and the variance of the noise, guaranteeing good prediction performance. However, it is fundamental that  $\mathcal{D}_{tr}^{DS}$  includes at least one initial and one final run for the algorithm to work, so that all the possible force range generated by the drift is spanned. Finally, the computational time required to train all the models in  $\mathcal{D}_{tr}$  and in  $\mathcal{D}_{tr}^{DS}$  are compared in Table I. In  $\mathcal{D}_{tr}$ , GBR and RF are one order of magnitude

TABLE I  
COMPUTATIONAL TIME REQUIRED TO TRAIN ALL THE MODELS.

	GPR	HGPR	LGPR	GBR	RF
Time [s] - $\mathcal{D}_{tr}$	275.97	685.21	101.63	7.06	8.09
Time [s] - $\mathcal{D}_{tr}^{DS}$	13.58	59.71	2.28	1.59	5.4

faster than the GP-based models. However, GBR and RF are not able to correctly classify all the normal and abnormal runs, which may make them unsuitable for high-confidence manufacturing anomaly detection purposes. Among the GP-based methods, HGPR is the slowest one, because it has the most complicated model with the highest number of hyperparameters to estimate. LGPR is the fastest one because trains independent GPs in each cluster with a smaller number of data, thus minimizing the  $\mathcal{O}(N^3)$  complexity. Considering  $\mathcal{D}_{tr}^{DS}$  the computational time is reduced dramatically making the GP-based methods speed competitive with GBR and RF.

Regarding the case described in Section V-B with  $z$ -force drift the obtained results reflected the conclusions already derived: HGPR and LGPR are the only methods with 100% accuracy in anomaly detection and LGPR is the fastest GP-based method slightly slower than GBR.

## VI. CONCLUSIONS

We propose an anomaly detection algorithm based on probabilistic models belonging to the Gaussian Process Regression framework. We analyzed three varieties of GP models both in terms of prediction accuracy in detecting the anomalies and in terms of computational time. These methods are also compared with other two machine learning techniques, GBR and RF.

Our analysis showed that the most promising methods for anomaly detection are HGPR and LGPR. With advantages that the former is possibly the most accurate in estimating the confidence intervals and the latter is faster to train. However, both these methods are able to correctly detect all the normal and abnormal runs. Our approach solves the problem of undesired effects that can be found in real industrial applications such as the  $z$ -force drift. To further explore this kind of problems, in future works we want to estimate the confidence intervals within which the next run should

belong to knowing the last normal run. Suppose that the robot is performing in sequence multiple runs, after one normal run is executed instead then expecting the following run to be included in the whole admissible confidence interval we want to learn an appropriate probabilistic model that will infer where only the next run will be. This should reduce considerably the size of the expected confidence intervals.

Further extensions of the proposed approach might include solving the anomaly detection problem in real time (i.e., while the run is performing and not complete), consider faulty and uncertain sensors and classifying different faults.

## REFERENCES

- [1] W. Wan, F. Lu, Z. Wu, and K. Harada, "Teaching robots to do object assembly using multi-modal 3d vision," *Neurocomputing*, vol. 259, pp. 85–93, 2017.
- [2] P. Jiménez, "Survey on assembly sequencing: a combinatorial and geometrical perspective," *Journal of Intelligent Manufacturing*, vol. 24, no. 2, pp. 235–250, 2013.
- [3] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.
- [4] O. Pettersson, "Execution monitoring in robotics: A survey," *Robotics and Autonomous Systems*, vol. 53, no. 2, pp. 73–88, 2005.
- [5] R. Isermann, "Model-based fault-detection and diagnosis—status and applications," *Annual Reviews in control*, vol. 29, pp. 71–85, 2005.
- [6] F. Chen, K. Sekiyama, H. Sasaki, J. Huang, B. Sun, and T. Fukuda, "Assembly strategy modeling and selection for human and robot coordinated cell assembly," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 4670–4675.
- [7] J. Huang, P. Di, T. Fukuda, and T. Matsuno, "Robust model-based online fault detection for mating process of electric connectors in robotic wiring harness assembly systems," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 5, pp. 1207–1215, 2010.
- [8] J. Huang, T. Fukuda, and T. Matsuno, "Model-based intelligent fault detection and diagnosis for mating electric connectors in robotic wiring harness assembly systems," *IEEE/ASME Transactions on Mechatronics*, vol. 13, no. 1, pp. 86–94, 2008.
- [9] J. Huang, P. Di, T. Fukuda, and T. Matsuno, "Fault-tolerant mating process of electric connectors in robotic wiring harness assembly systems," in *7th World Congress on Intelligent Control and Automation*. IEEE, 2008, pp. 2339–2344.
- [10] K.-W. Cheng, Y.-T. Chen, and W.-H. Fang, "Video anomaly detection and localization using hierarchical feature representation and gaussian process regression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2909–2917.
- [11] K. Kowalska and L. Peel, "Maritime anomaly detection using gaussian process active learning," in *15th International Conference on Information Fusion (FUSION)*. IEEE, 2012, pp. 1164–1171.
- [12] D. K. Jha, A. Srivastav, and A. Ray, "Temporal learning in video data using deep learning and gaussian processes," in *Workshop on Machine Learning for Prognostics and Health Management at 2016 KDD, San Francisco, CA*, 2016.
- [13] C. E. Rasmussen, "Gaussian processes in machine learning," in *Advanced lectures on machine learning*. Springer, 2004, pp. 63–71.
- [14] D. Nguyen-Tuong, M. Seeger, and J. Peters, "Model learning with local gaussian process regression," *Advanced Robotics*, vol. 23, no. 15, pp. 2015–2034, 2009.
- [15] P. W. Goldberg, C. K. Williams, and C. M. Bishop, "Regression with input-dependent noise: A gaussian process treatment," in *Advances in neural information processing systems*, 1998, pp. 493–499.
- [16] Q. V. Le, A. J. Smola, and S. Canu, "Heteroscedastic gaussian process regression," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 489–496.
- [17] K. Kersting, C. Plagemann, P. Pfaff, and W. Burgard, "Most likely heteroscedastic gaussian process regression," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007.
- [18] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.