

Machine learning approaches to environmental disturbance rejection in multi-axis optoelectronic force sensors



J. Gafford^{a,*}, F. Doshi-Velez^a, R. Wood^{a,b}, C. Walsh^{a,b}

^a Harvard University, John A. Paulsen School of Engineering and Applied Sciences, 60 Oxford Street, Cambridge, MA 02138, United States

^b Wyss Institute for Biologically-Inspired Engineering, 3 Blackfan Circle, Boston, MA 02115, United States

ARTICLE INFO

Article history:

Received 25 March 2016

Received in revised form 15 June 2016

Accepted 26 June 2016

Available online 20 July 2016

MSC:

00-01

99-00

Keywords:

Light-intensity-modulation

Force sensors

Surgical robotics

Machine learning

Nonlinear regression

ABSTRACT

Light-intensity modulated (LIM) force sensors are seeing increasing interest in the field of surgical robotics and flexible systems in particular. However, such sensing modalities are notoriously susceptible to ambient effects such as temperature and environmental irradiance which can register as false force readings. We explore machine learning techniques to dynamically compensate for environmental biases that plague multi-axis optoelectronic force sensors. In this work, we fabricate a multisensor: three-axis LIM force sensor with integrated temperature and ambient irradiance sensing manufactured via a monolithic, origami-inspired fabrication process called printed-circuit MEMS. We explore machine learning regression techniques to compensate for temperature and ambient light sensitivity using on-board environmental sensor data. We compare batch-based ridge regression, kernelized regression and support vector techniques to baseline ordinary least-squares estimates to show that on-board environmental monitoring can substantially improve sensor force tracking performance and output stability under variable lighting and large ($>100^\circ\text{C}$) thermal gradients. By augmenting the least-squares estimate with nonlinear functions describing both environmental disturbances and cross-axis coupling effects, we can reduce the error in F_x , F_y and F_z by 10%, 33%, and 73%, respectively. We assess viability of each algorithm tested in terms of both prediction accuracy and computational overhead, and analyze kernel-based regression for prediction in the context of online force feedback and haptics applications in surgical robotics. Finally, we suggest future work for fast approximation and prediction using stochastic, sparse kernel techniques.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Light-intensity-modulated (LIM) force sensors comprise a subclass of force sensors that rely on optoelectronic emitter–detector pairs as transduction mechanisms to measure force by detecting relative motion within a compliant structure, as illustrated schematically in Fig. 1. Such a methodology is seeing increasing interest in the surgical robotics community over other force sensing technologies (i.e. strain gage, capacitive) due to MRI compatibility (for fiber-based LIM sensors) and the potential to be implemented on non-metallic (flexible) robotic platforms that undergo large strains. Optoelectronic sensors have been used to sense forces [1–3] and enable closed-loop shape estimation [4,5] of flexible surgical robots. A drawback of using optoelectronic transduction mechanisms is an inherent sensitivity to ambient conditions such as

environmental irradiance and temperature, which introduce errors that arise in the form of ‘false’ force measurements. Especially in electrosurgical interventions which generate high temperature gradients ($>50^\circ\text{C}$) and electromagnetic interference (EMI), distal sensing methodologies must be robust to these effects if closed-loop force control is to be achieved [6]. In order to simplify the often costly and complex hardware and signal conditioning requirements typically implemented to compensate for these disturbances, as well as to relax optical isolation and isothermal requirements, it is beneficial to explore algorithmic methods which can reject these errors in real time based on actual environmental feedback integrated into the sensor itself.

Fig. 2 shows typical behavior of a fully-integrated optoelectronic multi-axis force sensor under null force, but variable lighting and temperature profiles induced by adjusting the ambient light and applying heat with a heat gun. The bottom plot shows the emitter–detector pair outputs as they correlate to environmental changes which are being measured simultaneously by temperature and irradiance sensors as shown in the top plot. Observe how each

* Corresponding author.

E-mail address: jgafford@seas.harvard.edu (J. Gafford).

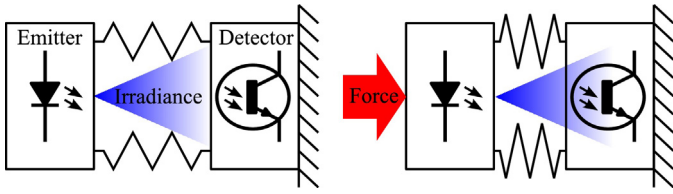


Fig. 1. Simplified schematic of discrete-component light-intensity modulation sensing principle.

emitter/detector pair has a different response to relative changes in light (denoted by light green shading), whereas temperature dependencies are somewhat uniform amongst all sensors (denoted by light red shading). We see how these effects could be construed as force readings despite the absence of applied force; thus, the need for environmental compensation is clear.

There has been some prior work in employing machine learning techniques to improve sensor performance in the face of nonlinearities, disturbances or other detrimental effects. Neural networks have been used to approximate nonlinearities in multi-axis force sensors [7–9]. Artificial neural networks (ANN) have been employed to linearize the behavior of capacitive humidity sensors [10]. We chose not to pursue NNs as they have a tendency to overfit the data, require numerous hyperparameters to be optimized, and are not guaranteed to converge to local minima. Newer techniques, such as Locally-Weighted Projection Regression (LWPR) have been used in force control applications [11]; however, like NNs, LWPR typically requires optimal tuning of numerous hyperparameters. Kernel-based support vector machines (SVM) have been employed to fit nonlinear functions to cross-axis coupling terms in multi-axis strain gage-based force sensors [12] and to compensate for nonlinear and environmental effects in photoelectric displacement sensors [13] and capacitive pressure sensors [14]. To the author's knowledge, there exists no prior literature in applying kernel-based machine learning techniques to actively reject thermal and ambient light disturbances while approximating cross-axis coupling relationships for optoelectronic multi-axis force sensing methodologies.

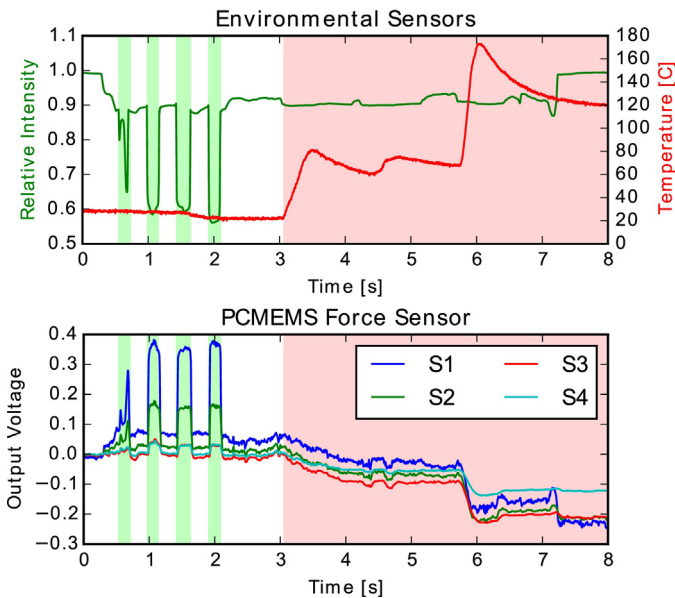


Fig. 2. PCMEMS multisensor response to environmental disturbance with null force input: (top) environmental sensor readings (where green shading denotes a change in ambient light and red shading denotes a temperature change), (bottom) corresponding emitter/detector output changes. (For interpretation of reference to color in this figure legend, the reader is referred to the web version of this article.)

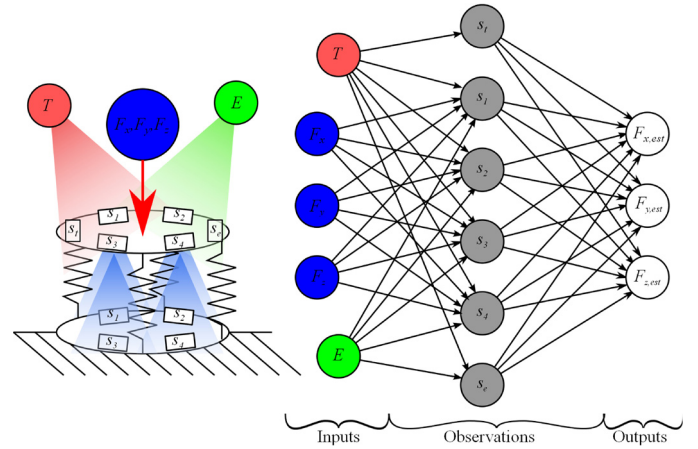


Fig. 3. (left) Operational schematic, (right) graphical model demonstrating the complex inter-relationships between inputs, observations and outputs, illustrating how knowledge of $s_1, \dots, s_4$ alone is insufficient to accurately reconstruct the input force vector.

Herein we describe the design and fabrication of a multi-axis sensor with on-board temperature and irradiance sensing capabilities using a novel monolithic fabrication process called printed-circuit MEMS (PCMEMS) [15]. Building on prior work in monolithic sensor fabrication [16] and discrete-component LIM force sensors [17], we improve performance by developing a model of the system that includes both environmental and cross-axis error terms and attempt to approximate these effects using simple linear techniques (ridge regression), kernel techniques (kernelized ridge regression, ϵ -sensitive support vector regression), and stochastic methods (sub-gradient support vector regression). We assess each approach in terms accuracy and computational complexity in both training and prediction. Overall, we demonstrate that the performance of multi-axis optoelectronic sensors can be drastically improved by including on-board estimates of environmental data. We also demonstrate that nonlinear kernel-based methods result in substantial performance improvements, and discuss ways in which these techniques can be applied to online estimation for deployable implementations. In summary, the contribution of this work is twofold: (1) implementation of monolithic fabrication methods to build high-quality LIM multi-axis force sensors, and (2) exploration of machine learning techniques to cancel disturbances and approximate (and compensate for) cross-axis terms. The result is a highly-accurate optoelectronic multi-axis force sensor with a range of ± 2.5 N, ± 2.5 N, ± 5 N and errors of 2.6%, 1.8% and 2.6% full-scale in F_x , F_y and F_z , respectively. In addition, the sensor's behavior is stable over the tested thermal gradient which exceeds 100 °C.

2. Sensor design and modeling

A discrete-component, multi-axis, light-intensity-modulated force sensor consists of l_s emitter–detector (E/D) pairs (IR LEDs and phototransistors) separated from each-other by an elastic structure, as illustrated in Fig. 3(left). As a force is applied to the structure, causing it to deflect in m degrees-of-freedom, the relative distance separating each emitter–detector pair is changed, and using the principle of light intensity modulation, we can combine the outputs of each pair to reconstruct the input force vector so long as $l_s \geq m$. However, as shown in Fig. 3(right), the influence of temperature and environmental irradiance affect the signals generated by each emitter–detector pair, and these effects can be interpreted as ‘false’ forces. We observe the information flow from component-wise input forces (blue) which influence E/D pairs ($s_1 - s_4$) through mechanical coupling. However we also see how temperature (red)

and irradiance (green) also influence $s_1 - s_4$ through thermal and optical coupling. When forces are reconstructed from available sensor data, it is impossible to discern actual input forces from thermal and irradiance effects by $s_1 - s_4$ alone. If we can observe these environmental effects separately through s_t and s_e , we can approximate their effects on the system as a whole and cancel them out of the final reconstruction.

2.1. Analytical and numerical modeling

In developing a simple mechanical model of the system, consider a three-dimensional input force vector $\mathbf{f} = [f_x, f_y, f_z]^T$ and the effect it has on the spacing of an arbitrary emitter–detector pair j . We consider a stiffness matrix $\mathbf{K}_j$ that contains the stiffness terms of the bulk structure as well as any homogeneous transformations between the location of the applied force and the location of the emitter–detector pair j in the structure. We can write this as follows:

$$\begin{bmatrix} \delta_{jx} \\ \delta_{jy} \\ \delta_{jz} \end{bmatrix} = \begin{bmatrix} k_{jx,x} & k_{jx,y} & k_{jx,z} \\ k_{jy,x} & k_{jy,y} & k_{jy,z} \\ k_{jz,x} & k_{jz,y} & k_{jz,z} \end{bmatrix}^{-1} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \quad (1)$$

where $\Delta_j = [\delta_{jx}, \delta_{jy}, \delta_{jz}]^T$ is a vector containing the relative displacement between the emitter and the detector given the applied force. Note that $\mathbf{K}_j$ is positive definite and symmetric and therefore always invertible. Assuming we know the elements of $\mathbf{K}_j$ and, therefore, Δ_j given an applied force, previous work [17] has shown that we can model the voltage output of emitter/detector pair j as:

$$s_j = g(\mathbf{f}) \quad (2)$$

$$= GR_G \left(\cos(\theta_j) \frac{I_{LED}}{(h - \delta_{jz} \sin(\theta_j))^2} \right)^\gamma + \beta \quad (3)$$

where G is the amplifier gain, R_g is the resistance value at the emitter-side of the phototransistor, I_{LED} is the emitter forward-current, h is the initial (undeformed) spacing between emitter and detector, γ and β are manufacturer-supplied scaling parameters, and θ_j is the misalignment angle between emitter and detector, given by the following simple model:

$$\theta_j = \tan^{-1} \left(\frac{\sqrt{\delta_{jx}^2 + \delta_{jy}^2}}{h} \right) \quad (4)$$

Finite element analyses (Solidworks Simulation, Dassault systems) were performed on the sensor's proposed mechanical structure to understand how three-dimensional input forces affect the relative spacing of each emitter/detector pair (i.e. given $\mathbf{f}$, compute Δ_j). These results, combined with the analytical model, generate the theoretical sensitivity curves given in Fig. 4(b), (d), and (f). We observe the highly nonlinear behavior of the output signal as a function of applied force. This will affect the fidelity of linear base models as we will demonstrate in subsequent sections. Finally, the amount of collector current generated over the desired force range gives us an idea of how much gain is required to convert this current into an appropriate voltage level that can be processed by a data acquisition unit. The stress levels indicated by the finite element models informed a design range of ± 2.5 N in x , y and ± 5 N in z . These were chosen to achieve a mechanical factor of safety of at least 2.5 for each axis.

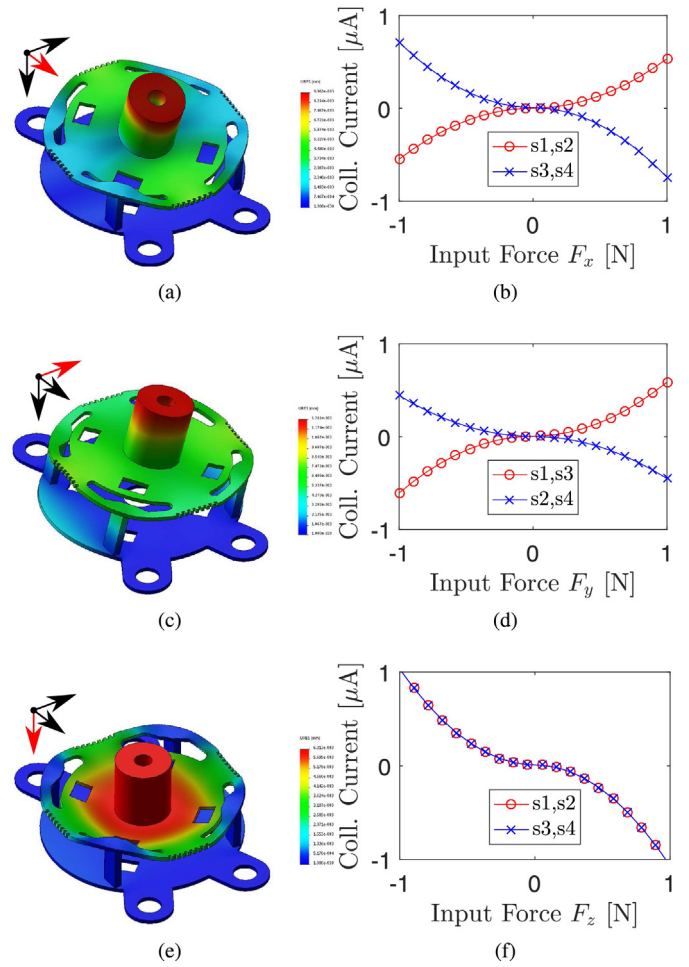


Fig. 4. Component-wise finite element simulation results (a),(c),(e) and associated (relative) collector current sensitivities (b),(d),(f) based on FEA results combined with analytical modeling.

2.2. Fabrication

The sensor was fabricated using printed-circuit MEMS [15,18]. The composite laminate consists of four layers of 75 μm 304 stainless steel, two layers of 25 μm Kapton polyimide, two layers of 25 μm Kapton polyimide with 18 μm copper cladding, and seven layers of DuPont Pyralux F0100 adhesive. Individual layers of material are machined using a diode-pumped solid-state laser and laminated together using heat and pressure. The laminate is then re-machined in the laser to release the sensor structure from the surrounding assembly scaffold. The purely two-dimensional laser machining and lamination process results in the composite structure shown in Fig. 5(left). Assembly is guided by four axisymmetric Sarrus linkage assemblies which constrain assembly kinematics to the z -axis (Fig. 5(middle), (right)), enabling trivial assembly into the 3D structure. These linkages also help to prevent torsional and



Fig. 5. PCMEMS multisensor structural assembly: (left) post-fabrication after release cuts are made, (middle) axisymmetric Sarrus linkages guide assembly of sensor along z -axis, (right) interlocking stiffeners are folded into corresponding slots to transmit load to flexural members.

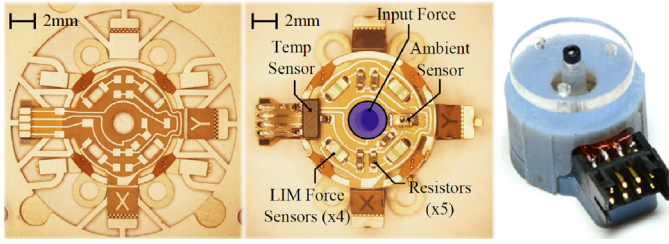


Fig. 6. Fabricated PCMEMS multisensor: (left) post-fabrication prior to assembly, still attached to assembly scaffold, (middle) assembled with components placed, (right) integrated into 3D-printed casing.

transverse deformations in the structure. The final assembly step includes manually folding four stiffener beams into corresponding slots, thereby transmitting input forces to the flexural members. After assembly is complete, electrical components are reflow soldered into place onto the top and bottom flexible circuit layers. The fabricated sensor, measuring roughly 10 mm in diameter and 3.5 mm in height, is then integrated into a 3D-printed casing for encapsulation. Fig. 6 shows the fabricated sensor in various stages of integration. The assembly sarrus linkages ‘jam’ against the inner wall of the casing if sufficient load is applied, thereby providing a mechanical ‘stop’ to prevent overloading.

A custom excitation and filtering circuit, described more thoroughly in [17], converts the small collector current from each emitter–detector pair (consisting of a HIR19-21C IR LED and PT19-21C IR phototransistor, both from Everlight) into a DC-level voltage that can be processed by a data acquisition unit. The temperature sensor (MCP9701T-E/TT from Microchip Technology) is an integrated circuit that contains all necessary conditioning circuitry. The ambient sensor (PT19-21C) was low-pass filtered with a cutoff of 1 kHz.

2.3. Experimental setup

The PCMEMS multisensor was rigidly fastened to an ATI-Nano17 6-axis commercial load cell which serves as ground truth. Data were collected using LabView at a rate of 2 kHz for roughly 135 s (using USB-6002 DAQ for the PCMEMS multisensor and PCI6259 DAQ for the ATI Nano17, both from National Instruments). Variable loading, lighting, and temperature conditions were applied through the duration of data capture. A heat gun was used to apply excessive thermal gradients above ambient. The experimental setup is shown in Fig. 7. The resulting data set was randomly shuffled and split into

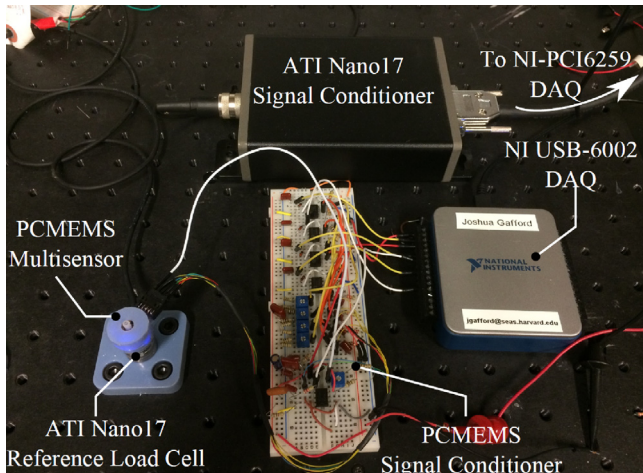


Fig. 7. Experimental setup, showing the encapsulated PCMEMS sensor mounted directly onto an ATI Nano16 reference load cell.

a training and testing set with a ratio of 75%/25%. The result is a training set of roughly $n_{train} \sim 200,000$ data points and a testing set of $n_{test} \sim 70,000$ data points.

3. Inference

In this section, we consider a simple linear inference model of the sensor to serve as a baseline for comparison. We then develop a model to describe the effects of temperature, irradiance, and cross-axis coupling to augment the baseline model. We explore linear and kernel-based techniques for optimizing the weighting parameters of the augmented model, and discuss trade-offs in terms of accuracy and computational complexity.

3.1. Baseline model (Moore–Penrose)

The most common means of calibrating multi-axis force sensors is to apply a calibration profile (i.e. some static or dynamic combination of known forces), and record the sensor outputs to formulate an inverse calibration matrix $\mathbf{W}_{MP}^{-1} \in \mathbb{R}^{m \times l}$, and perform Moore–Penrose pseudoinversion to compute the forward calibration matrix $\mathbf{W}_{MP} \in \mathbb{R}^{l \times m}$ [19]. Moore–Penrose theory states that, for any rectangular matrix $\mathbf{C}$, there exists a unique matrix $\mathbf{C}^+$ that satisfies the four Penrose conditions. A corollary of Moore–Penrose theory says that, if $\mathbf{C}$ has full row rank, the matrix $\mathbf{C}^+$ reduces to:

$$\mathbf{C}^+ = (\mathbf{C}^T \mathbf{C})^{-1} \mathbf{C} \quad (5)$$

As shown, the calibration matrix $\mathbf{C}^+ \equiv \mathbf{W}_{MP}$ amounts to the ordinary least squares (OLS) solution without regularization. Performing least-squares regression on a time-series of $s_1, \dots, s_4$ with N observations (i.e. $\mathbf{X}_s \in \mathbb{R}^{N \times 4}$) without accounting for temperature and ambient light yields the following calibration matrix:

$$\hat{\mathbf{Y}}_{MP} = \mathbf{X}_s \mathbf{W}_{MP} \quad (6)$$

$$= [\mathbf{s}_1 \quad \mathbf{s}_2 \quad \mathbf{s}_3 \quad \mathbf{s}_4] \begin{bmatrix} 1.11 & 3.75 & -6.36 \\ 3.96 & 6.27 & -6.50 \\ -4.00 & -2.89 & 2.40 \\ -0.44 & -4.44 & 3.58 \end{bmatrix} \quad (7)$$

where $\mathbf{s}_0 = [s_0(1), \dots, s_0(i), \dots, s_0(N)]^T \in \mathbb{R}^N$ is a matrix containing all observations from sensor o , and $\hat{\mathbf{Y}}_{MP} \in \mathbb{R}^{N \times m}$ is a matrix containing the estimated force components for N samples. For three-axis force sensing, $\hat{\mathbf{Y}}_{MP} = [\hat{\mathbf{f}}_{x,MP}, \hat{\mathbf{f}}_{y,MP}, \hat{\mathbf{f}}_{z,MP}]$ where $\hat{\mathbf{f}}_{p,MP} = [\hat{f}_{p,MP}(1), \dots, \hat{f}_{p,MP}(i), \dots, \hat{f}_{p,MP}(N)]^T \in \mathbb{R}^N$ is the series of Moore–Penrose estimates for force component p .

Timeplots of the component-wise PCMEMS sensor estimates, compared with the reference ATI Nano17 readings (denoted $\mathbf{Y} = [\mathbf{f}_x, \mathbf{f}_y, \mathbf{f}_z] \in \mathbb{R}^{N \times m}$), are shown in Fig. 8. The shaded area represents the 95% credible interval. We observe that regression on F_x and F_y performs well even using a simple regression model ignoring environmental effects, while prediction for F_z is markedly poor. This is a construct of the sensor morphology, where forces applied in x and y ultimately generate differential signals in which common-mode errors get canceled out (i.e. F_x causes s_1 and s_3 to increase while causing s_2 and s_4 to decrease). Conversely, the performance of F_z is poor as this component does not generate differential signals, and as a result, common-mode errors still exist in the final measurement. To correct this common-mode contamination of F_z , we can improve over the baseline by integrating on-board environmental measurements and performing complete-data ridge regression, kernel regression and SVR-augmented OLS as discussed below.

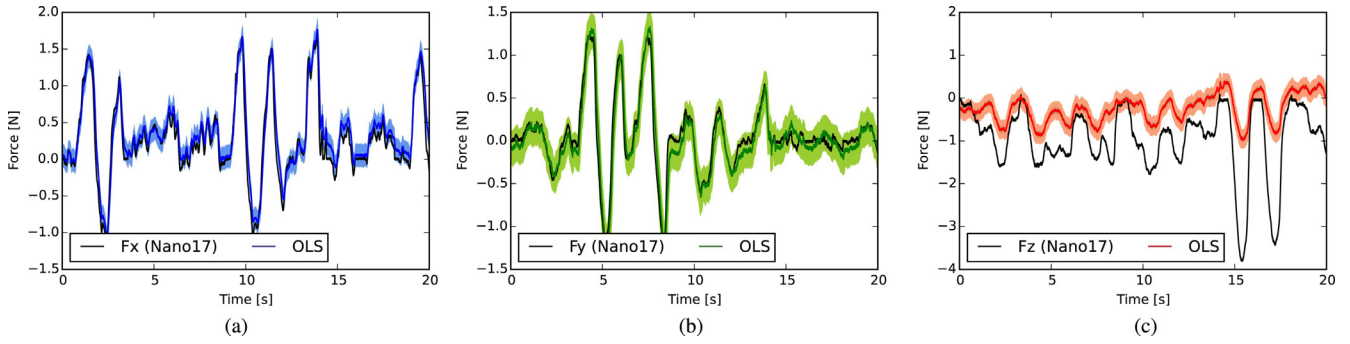


Fig. 8. Comparison of ground-truth force data with estimates generated via component-wise ordinary least-squares regression (Moore–Penrose pseudoinversion) using a linear combination of $\mathbf{s}_1, \dots, \mathbf{s}_4$: (a) $\hat{\mathbf{f}}_{x,MP}$ and $\mathbf{f}_x$, (b) $\hat{\mathbf{f}}_{y,MP}$ and $\mathbf{f}_y$, (c) $\hat{\mathbf{f}}_{z,MP}$ and $\mathbf{f}_z$. In general, we observe how the differential nature of F_x and F_y results in stable measurements, but the common-mode nature of F_z results in poor performance.

3.2. Ridge regression

Improving over the baseline, we combine outputs from $l_s = 4$ E/D pairs ($s_1, \dots, s_4$) with outputs from $l_e = 2$ environmental sensors, denoted s_t and s_e , respectively. For convenience, we define feature space sets $\mathbf{X}_s = \{\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \mathbf{s}_4\} \in \mathbb{R}^{N \times l_s}$ and $\mathbf{X}_e = \{\mathbf{s}_t, \mathbf{s}_e\} \in \mathbb{R}^{N \times l_e}$, and note that $\mathbf{X}_s \cap \mathbf{X}_e = \mathbf{X} \in \mathbb{R}^{N \times l}$ (the union of the two sets comprises a complete data set) but $\mathbf{X}_s \cup \mathbf{X}_e = \emptyset$ (E/D data and environmental data are distinct and non-interfering). Ultimately, we seek to find some combinatorial formulation $\mathbf{W} \in \mathbb{R}^{l \times m}$ of all on-board sensor data $\mathbf{X}$ to reproduce the most accurate estimate of the input force $\mathbf{Y} \in \mathbb{R}^{N \times m}$. A simple linear model is proposed to treat the actual force input as some weighted, linear combination of individual sensor voltages $\mathbf{X}$ corrupted by some component-wise noise $\boldsymbol{\varepsilon} \in \mathbb{R}^{N \times m}$ with each row entry $\varepsilon(i)$ drawn from the normal distribution $\mathcal{N}(\mu = \mathbf{0}, \Sigma \in \mathbb{R}^{m \times m})$.

$$\hat{\mathbf{Y}} = \mathbf{X}\mathbf{W} + \boldsymbol{\varepsilon} \quad (8)$$

To further simplify the problem, we re-write $\Sigma = l\sigma_{wc}^2$ where σ_{wc}^2 is the worst-case noise variance of all the sensors. We can use l_2 regularized regression techniques to fit a linear model to the data by computing the maximum a posteriori (MAP) estimate of the data given a prior distribution ($\sim \mathcal{N}(\mathbf{0}, (1/\tau^2)\mathbf{I})$ where $\mathbf{I} \in \mathbb{R}^{l \times l}$ is the identity matrix) on the regression weights to prevent overfitting as we explore higher-dimensional feature spaces. We no longer constrain ourselves to linear inputs, and are free to build a mapping $\mathbf{X} \mapsto (\Phi \in \mathbb{R}^{N \times l_\Phi})$ where Φ is some nonlinear function of $\mathbf{X}$ and l_Φ is the dimensionality of the new feature space. Assuming a Gaussian prior with precision τ^2 on the regression weights, and a sensor noise with variance σ^2 , we can compute a numerically-stable regularized estimate of the feature weights $\hat{\mathbf{w}}_{RR}$ which involves constructing an ‘augmented’ feature space matrix $\tilde{\Phi}$ and output matrix $\tilde{\mathbf{y}}$ as below:

$$\tilde{\Phi} = \begin{bmatrix} \Phi/\sigma \\ \sqrt{\Lambda} \end{bmatrix} \quad (9)$$

$$\tilde{\mathbf{y}} = \begin{bmatrix} \mathbf{y}/\sigma \\ \mathbf{0} \end{bmatrix} \quad (10)$$

where $\sqrt{\Lambda}$ is the Cholesky decomposition of $\Lambda = (1/\tau^2)\mathbf{I}$. We compute the Cholesky decomposition of the augmented feature space, and compute the optimum regressor estimate as follows:

$$\tilde{\Phi} = \mathbf{Q}\mathbf{R} \quad (11)$$

$$\hat{\mathbf{w}}_{RR} = \mathbf{R}^{-1}\mathbf{Q}^T\tilde{\mathbf{y}} \quad (12)$$

The output hypothesis is given by:

$$\hat{\mathbf{Y}}_{RR} = \Phi\hat{\mathbf{w}}_{RR} \quad (13)$$

where $\hat{\mathbf{Y}}_{RR}$ is the ridge estimate. Note that, by using the Cholesky decomposition, in optimizing the regression weights we only have to invert the triangular matrix $\mathbf{R}$ which is $\mathcal{O}(n^2)$, as opposed to inverting the full $(\Phi^T\Phi + \lambda\mathbf{I})$ matrix which would be $\mathcal{O}(n^3)$. For this paper we consider three different mappings:

- *Complete data linear mapping:* We preserve a linear mapping for all sensor data, i.e. for $\mathbf{s}_0 \in \mathbf{X}$, $\mathbf{s}_0 \mapsto a + b\mathbf{s}_0$.
- *Complete data quadratic mapping:* We map all sensor data to a quadratic feature space: for $\mathbf{s}_0 \in \mathbf{X}$, $\mathbf{s}_0 \mapsto a + b\mathbf{s}_0 + c\mathbf{s}_0 \odot \mathbf{s}_0^1$.
- *Complete data physics-derived mapping:* For $\mathbf{s}_0 \in \mathbf{X}_s$, $\mathbf{s}_0 \mapsto a + c\mathbf{s}_0 \odot \mathbf{s}_0$. Meanwhile we preserve a linear mapping for temperature and irradiance data, i.e., for $\mathbf{s}_0 \in \mathbf{X}_e$, $\mathbf{s}_0 \mapsto a + b\mathbf{s}_0$.

Ridge regression was performed on the training and test data by incorporating data from the temperature and ambient sensor. The initial data set $\mathbf{X}$ were transformed into a polynomial feature space $\Phi \in \mathbb{R}^{N \times (ld)}$ where $l = l_s + l_e = 6$ and d is the order of the polynomial transform. To justify the use of a second-order polynomial feature space, a parametric sweep was performed where, for a given feature space dimension, the root-mean-squared error was computed for each dimension d :

$$RMSE_m = \sqrt{\frac{\sum_{i=1}^N (f_{m,i} - \hat{y}_{RR,m,i})^2}{N}} \quad (14)$$

where m is the force component being evaluated $m \in \{x, y, z\}$. The results, shown in Fig. 9(a), demonstrate that a quadratic feature space is optimum, after which overfitting occurs. Here, the dotted black line is the RMSE magnitude over all components, given by:

$$|RMSE| = \left(\sqrt{RMSE_x^2 + RMSE_y^2 + RMSE_z^2} \right) \quad (15)$$

The component-wise testing RMSE for each transform is shown in Fig. 9(b), demonstrating that the quadratic and physics-derived feature spaces perform similarly. Force tracking results for a quadratic feature space transform on full sensor data are given in Fig. 9(c), showing a much better prediction of F_z . Note that F_x and F_y tracking results are excluded as the results show a negligible improvement in performance over the baseline. Thus it is observed that including environmental sensor data, in addition to preservation of the physics of the system in regression, can greatly improve the performance.

¹ Here, $\odot$ refers to an element-wise multiplication operation.

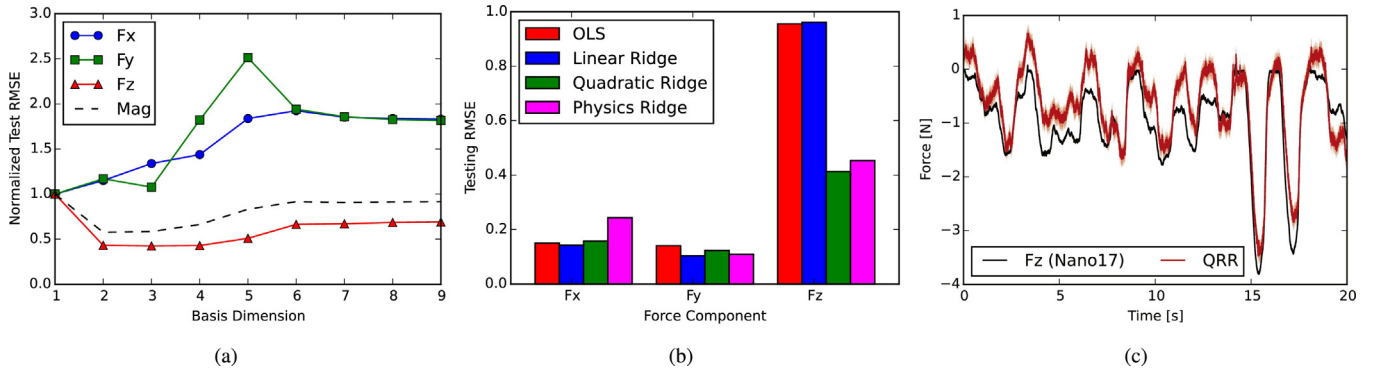


Fig. 9. (a) Parametric sweep for ridge regression with a polynomial feature space transformation where the testing error is plotted as a function of basis dimension (overfitting occurs after order 2), (b) bar plot comparing component-wise test data RMSE for each feature space employed as discussed in Section 3.2, and (c) $\hat{\mathbf{f}}_{z,RR}$ and $\mathbf{f}_z$ where Φ has a quadratic basis. We observe significant force tracking performance over the Moore–Penrose solution with $\mathbf{X}_s$ data only.

4. A modified model for kernel methods

To improve performance further, we modify the original model to include the OLS solution augmented with nonlinear function estimates of cross-axis terms, temperature sensitivity terms and ambient sensitivity terms, as follows:

$$\mathbf{Y} = \hat{\mathbf{Y}}_{MP} \begin{bmatrix} 1 & \varphi_{y,x} & \varphi_{z,x} \\ \varphi_{x,y} & 1 & \varphi_{z,y} \\ \varphi_{x,z} & \varphi_{y,z} & 1 \end{bmatrix} + \mathbf{X}_e \begin{bmatrix} \xi_{x,t} & \xi_{y,t} & \xi_{z,t} \\ \xi_{x,e} & \xi_{y,e} & \xi_{z,e} \end{bmatrix} \quad (16)$$

$$\mathbf{Y} = [\hat{\mathbf{Y}}_{MP} \quad \mathbf{X}_e] \begin{bmatrix} \varphi \\ \xi \end{bmatrix}$$

where $\hat{\mathbf{Y}}_{MP}$ is the original OLS hypothesis with sensor data in $\mathbf{X}_s$ (i.e. Eq. (7)), $\mathbf{X}_e = [\mathbf{s}_t, \mathbf{s}_e]$ are the temperature and irradiance readings, and $\varphi_{a,b}, \xi_{a,b}$ are nonlinear operators on components a, b . With this model, we nonlinearly approximate cross-axis terms $\varphi_{(\cdot,\cdot)}$ and environmental terms $\xi_{(\cdot,\cdot)}$ which are assumed to contribute to the error between the actual solution and the OLS solution. We define a new (modified) state matrix $\tilde{\mathbf{X}} = [\hat{\mathbf{Y}}_{MP} \quad \mathbf{X}_e] \in \mathbb{R}^{N \times (m+le)}$ for subsequent analysis. As this model introduces several new degrees-of-freedom to the problem, we explore kernelized methods to compute optimum estimates of φ and ξ . We make use of the ‘kernel trick’ which enables the use of linear algorithms to implicitly operate in a transformed feature space, resulting in an efficient means of constructing models which are nonlinear in input space. Below we explore full (kernelized ridge regression) and sparse (support vector regression) kernelized regression techniques, and attempt to speed up prediction time using stochastic methods.

4.1. Kernelized ridge regression

Saunders et al. [20] present a means of performing ridge regression using kernel operators (Kernelized ridge regression, or KRR). Given a test point $\mathbf{x}^*$, the output hypothesis $\hat{\mathbf{y}}$ is given by:

$$\hat{\mathbf{y}}_{KRR} = \mathbf{y}^T (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{k} \quad (17)$$

where $\mathbf{y}$ is the training data output, λ is a regularization parameter, $\mathbf{K}$ is a kernel matrix² with entries given by the following:

$$K_{i,j} = k(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j) \quad (18)$$

where $\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j$ are the i th and j th rows in $\tilde{\mathbf{X}}$ and $\mathbf{k}$ is the kernel vector with entries defined by the following:

$$k_i = k(\mathbf{x}^*, \tilde{\mathbf{x}}_i) \quad (19)$$

The computational complexity is dictated by the matrix inversion of $(\mathbf{K} + \lambda \mathbf{I})$ which is $\mathcal{O}(n^3)$, where n is the size of the training set. We cannot use the Cholesky decomposition as in regular ridge regression, as we cannot linearly separate $\hat{\mathbf{w}}$ from feature space $\tilde{\mathbf{X}}$ in the hypothesis evaluation. We consider three kernel transformations $k(\mathbf{x}, \mathbf{x}')$ as below:

- (1) Linear kernel: $\mathbf{x} \cdot \mathbf{x}'$.
- (2) Polynomial kernel: $(1 + \mathbf{x} \cdot \mathbf{x}')^d$ where d is the order.
- (3) Radial Basis Function (RBF) kernel: $\exp(-\nu \|\mathbf{x} - \mathbf{x}'\|^2)$ where ν is a free parameter.

KRR was performed on the modified model given in Eq. (16) to estimate nonlinear cross-axis and environmental dependencies φ and ξ , respectively. Due to memory constraints, the original data set was downsampled by 100 ($n = N/100$) to make the batch problem tractable. Although KRR was performed on linear, polynomial and RBF kernels, only the RBF results are shown in Fig. 10 for each component. We observe near perfect force tracking in F_x and F_y and significant improvements in F_z over the baseline.

4.2. ϵ -SVR and stochastic approximation

While kernelized ridge regression is adept at fitting nonlinear models for moderately-sized data sets, a lack of sparsity in the hypothesis lends to poor prediction scaling to larger sets in terms of computation time. We introduce sparsity by exploiting support vector regression (SVR) techniques which solve the same kernel problem by minimizing an l_1 objective function [21]. The idea is to ‘toss out’ hypotheses that lie within some user-specified ‘tube’ of the truth value, thereby embedding sparsity in the hypothesis that is a function of tube parameter ϵ .

Instead of applying the kernel trick directly to ridge regression to minimize an l_2 least-squares problem, we utilize sparse ϵ -sensitive support vectors for generating hypotheses. This involves finding an optimum set of multipliers $\alpha^{(*)}$ to minimize the following quadratic programming problem:

$$\begin{aligned} & -\frac{1}{2} \sum_{i,j} (\alpha(i) - \alpha(i)^*)(\alpha(j) - \alpha(j)^*) k(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j) \\ & - \epsilon \sum_{i=1}^N (\alpha(i) - \alpha(i)^*) + \sum_{i=1}^N y_i (\alpha(i) - \alpha(i)^*) \end{aligned} \quad (20)$$

² Not to be confused with stiffness matrix $\mathbf{K}_s$ defined earlier in the manuscript.

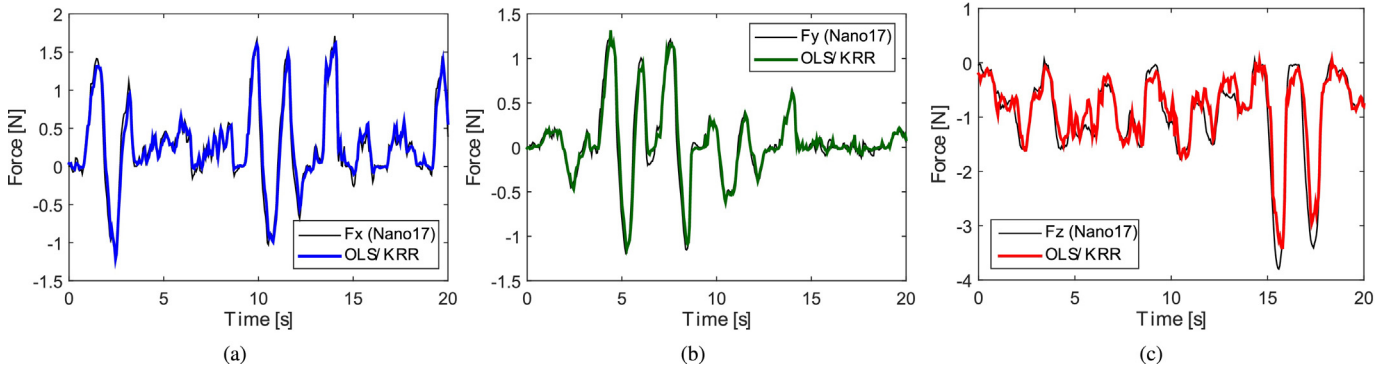


Fig. 10. Comparison of ground-truth force data with estimates generated via component-wise kernelized ridge regression with RBF kernel basis, where the data set has been downsampled for tractability (decimation = 100): (a) $\hat{\mathbf{f}}_{x,KRR}$ and $\mathbf{f}_x$, (b) $\hat{\mathbf{f}}_{y,KRR}$ and $\mathbf{f}_y$, (c) $\hat{\mathbf{f}}_{z,KRR}$ and $\mathbf{f}_z$. We observe significant performance improvements in force tracking of F_z , as well as noticeable performance improvements in F_x and F_y .

subject to

$$\sum_i^N (\alpha(i) - \alpha(i)^*) = 0 \quad (21)$$

$$\alpha(i), \alpha(i)^* \in [0, C] \quad (22)$$

where ϵ denotes the tube sensitivity (i.e. the amount of deviation from y_i allowed before penalization), and C is a regularization parameter.

We can represent the optimum parameter estimate as a linear combination of the support vectors transformed into a higher-dimensional space Φ using the representer theorem, as given below:

$$\mathbf{w} = \sum_{i \in \mathcal{D}_{SV}} (\alpha(i) - \alpha(i)^*) \Phi(\bar{\mathbf{x}}_i) \quad (23)$$

where we sum over the set of valid support vectors $\mathcal{D}_{SV}$. This leads to the following hypothesis for a test point $\mathbf{x}'$:

$$f(\mathbf{x}') = \sum_{i \in \mathcal{D}_{SV}} (\alpha(i) - \alpha(i)^*) k(\bar{\mathbf{x}}_i, \mathbf{x}') \quad (24)$$

This dual problem is inherently convex and can be posed as a quadratic programming problem which can be solved in batch form. We evaluated the same kernel transformations as used for KRR. Fig. 11(a) shows the force tracking performance in z after ϵ -SVR using RBF kernel. Note that F_x and F_y have been omitted for brevity as they are nearly identical to what is shown in Fig. 10(a) and (b). We see that the performance is similar to KRR but slightly worse, likely due to sparsity in the hypotheses.

4.2.1. Stochastic SVR for linear kernels

A drawback of batch-based SVR is that the approximation time depends on the size of the training set. For example, performing batch-based SGD with RBF kernel on the full dataset takes over two hours to train. To speed up the training process, we can exploit stochastic methods to iteratively minimize the SVR primal objective function via gradient descent. PEGASOS (Primal Estimated sub-GrAdient SOLver) was originally proposed in [22] as an efficient means of using minibatch stochastic gradient descent to solve binary classification problems ($y \in \{-1, 1\}$) using support vector machines. The general idea is to randomly select minibatches A_t of size q from the original data set $\mathcal{D}$ and perform SGD on the primal objective function $\mathbf{l}(\mathbf{w}; A_t^+)$ that contains the entries which correspond to valid support vectors in A_t (the subset $A_t^+ \subset A_t \subset \mathcal{D}$). We modified the original cost function to enable regression with support over $y \in \mathbb{R}$ with ϵ -sensitive tube loss, which linearly penalizes points that lie outside a 'tube' of width 2ϵ :

$$\mathbf{l}(\mathbf{w}; A_t) = \frac{\lambda}{2} \|\mathbf{w}\|^2 + \frac{1}{q} \sum_{i \in A_t} \max \{ |\langle \mathbf{w}, \bar{\mathbf{x}}_i \rangle - y_i| - \epsilon, 0 \} \quad (25)$$

For a linear kernel, the gradient with respect to parameter vector $\mathbf{w}$ can be easily calculated:

$$\frac{\partial \mathbf{l}(\mathbf{w}; A_t)}{\partial \mathbf{w}} = \lambda \mathbf{w} + \frac{1}{q} \sum_{i \in A_t} \text{sgn}(\langle \mathbf{w}, \bar{\mathbf{x}}_i \rangle - y_i) \bar{\mathbf{x}}_i \quad (26)$$

From this, we formulate a gradient descent algorithm with the following update step using an adaptive learning rate $\eta_t = 1/(\lambda t)$

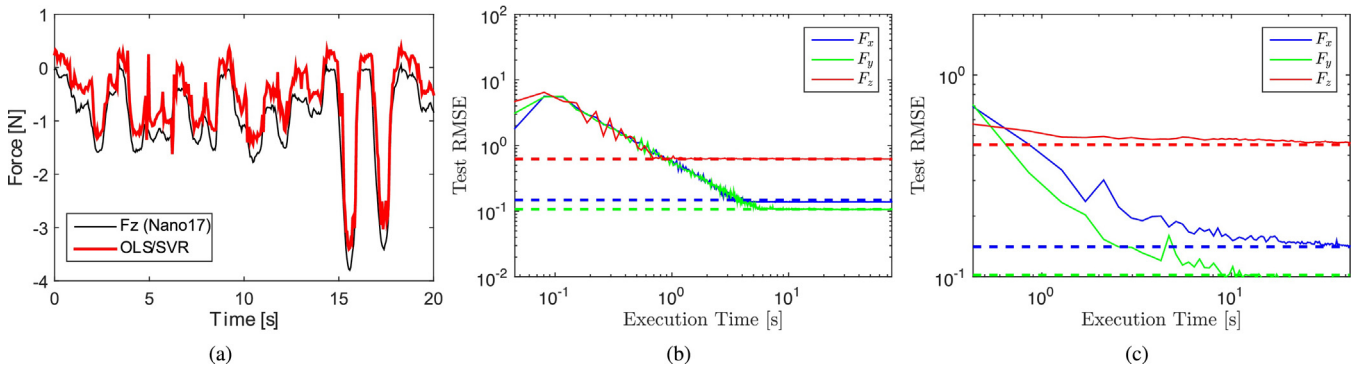


Fig. 11. ϵ -SVR results: (a) $\hat{\mathbf{f}}_{z,SVR}$ and $\mathbf{f}_z$ using batch-based ϵ -SVR with RBF kernel on a downsampled data set (decimation = 100), (b) linear PEGASOS on raw data set (decimation = 1) using a linear kernel, and (c) kernelized PEGASOS on downsampled data set (decimation = 100) using RBF kernel. Dashed lines indicate RMSE obtained via batch ϵ -SVR.

as proposed in [22]:

$$\mathbf{w}_{t+1/2} = \mathbf{w}_t - \eta_t \frac{\partial l(\mathbf{w}_t; A_t)}{\partial \mathbf{w}} = \mathbf{w}_t \left(1 - \frac{1}{t}\right) + \frac{\eta_t}{q} \sum_{i \in A_t} \text{sgn}(\langle \mathbf{w}_t, \tilde{\mathbf{x}}_i \rangle - y_i) \tilde{\mathbf{x}}_i \quad (27)$$

The final step is an optional projection step:

$$\mathbf{w}_{t+1} = \min \left\{ 1, \frac{1/\sqrt{\lambda}}{\|\mathbf{w}_{t+1/2}\|} \right\} \mathbf{w}_{t+1/2} \quad (28)$$

PEGASOS for a linear kernel converges with complexity $\mathcal{O}(d/(\lambda\epsilon))$ (d is the number of non-zero features in each training example, therefore the runtime is independent of the training set size) as opposed to $\mathcal{O}(d^2n)$ with batch-based SVR on a linear kernel [23].

4.2.2. Stochastic SVR for nonlinear kernels

To explore subgradient methods for SVR on nonlinear kernels, we use representer theorem to re-write $\mathbf{w}$ as a weighted linear combination of support vectors:

$$\mathbf{w} = \sum_{i \in \mathcal{D}_{SV}} \alpha(i) \Phi(\tilde{\mathbf{x}}_i) \quad (29)$$

Rather than replacing $\mathbf{w}$ in the primal objective function with the representer form and taking the gradient with respect to $\alpha(i)$, Shalev-Shwartz et al. [22] recommends taking the gradient with respect to $\mathbf{w}$ as the problem can be proven to be strictly convex. Following procedures outlined in [22] but using the ϵ -sensitive tube loss objective function:

$$l(\mathbf{w}, \tilde{\mathbf{x}}_i) = y_{i_t} - \langle \mathbf{w}_t, \phi(\tilde{\mathbf{x}}_i) \rangle \quad (30)$$

the update on $\mathbf{w}$ can be written as:

$$\mathbf{w}_{t+1} = \eta_t \sum_{i=1}^t \mathbb{1}[|l(\mathbf{w}, \tilde{\mathbf{x}}_i)| > \epsilon] \text{sgn}(l(\mathbf{w}, \tilde{\mathbf{x}}_i)) \phi(\tilde{\mathbf{x}}_i) \quad (31)$$

where $\mathbb{1}$ is the indicator function. We can treat $\alpha_{t+1}(j)$ as a counting variable that, for support vector j , counts up by η_t if the loss function is positive and counts down by η_t if the loss function is negative, so long as the tube sensitivity condition ($|l(\mathbf{w}, \tilde{\mathbf{x}}_i)| > \epsilon$) is met. We can then update $\mathbf{w}_{t+1}$ using representer theorem as in Eq. (29). In performing gradient descent, we now update each individual support vector (if it exists) until convergence.

Table 1

Algorithm performance summary, where bold indicates the best performance.

Algorithm	F_x RMSE (train)	F_y RMSE (train)	F_z RMSE (train)	F_x RMSE (test)	F_y RMSE (test)	F_z RMSE (test)	Training runtime [s]	$\hat{y}$ Pred. runtime [s]
Baseline (OLS, no s_e, s_t)	0.136	0.103	0.518	0.150	0.136	0.936	0.032	2.10E-5
Ridge (Linear)	0.128	0.091	0.288	0.142	0.103	0.961	0.201	2.50E-5
Ridge (Poly)	0.114	0.088	0.251	0.164	0.121	0.411	0.216	3.00E-5
Kernel Ridge (Linear)	0.126	0.099	0.483	0.137	0.105	0.629	129.8*	14.76*
Kernel Ridge (Poly)	0.130	0.096	0.297	0.131	0.092	0.306	3586*	15.06*
Kernel Ridge (RBF)	0.103	0.076	0.227	0.135	0.090	0.262	2340*	14.35*
Batch ϵ -SVR (Linear)	0.153	0.106	0.566	0.148	0.107	0.621	10,700*	6.554*
Batch ϵ -SVR (Poly)	0.123	0.088	0.404	0.147	0.104	0.416	15,900*	7.168*
Batch ϵ -SVR (RBF)	0.119	0.075	0.305	0.141	0.102	0.409	7900*	6.892*
SGD ϵ -SVR (Linear)	0.134	0.099	0.534	0.137	0.109	0.656	6.81	8.72E-4
SGD ϵ -SVR (Poly)	0.131	0.099	0.423	0.141	0.101	0.432	18,100*	0.491*
SGD ϵ -SVR (RBF)	0.121	0.096	0.332	0.145	0.102	0.414	3100*	0.288*

Note: (*) indicates approximate full data run time based on downsampled data run time.

4.3. Stochastic SVR results

Fig. 11(b) and (c) show the test set RMSE convergence to the batch value as a function of time for both linear and nonlinear kernels. Linear PEGASOS was performed on the full data set, and we observe convergence to the batch RMSE (from the downsampled dataset) after about six seconds. As such, this method is orders of magnitude more efficient than batch support vector regression using a linear kernel. However, for nonlinear kernels, each iteration requires a kernel evaluation over up to n training examples, and as a result, this algorithm is not particularly efficient for nonlinear kernels as the runtime complexity is $\mathcal{O}(n_{sv}/(\lambda\epsilon))$. Therefore the raw dataset was downsampled for tractability. The resulting convergence curve is shown in Fig. 11(c) and we observe convergence to the batch value after about 30 s on the subsampled data set. To summarize, while SGD methods are particularly efficient for approximation with linear kernels, nonlinear kernels are preferred due to improved tracking accuracy at the cost of runtime performance. In future work, we discuss nonlinear kernel approximation methods that could potentially speed up convergence of nonlinear stochastic support vector methods.

5. Discussion

Training and test set RMSE for each of the methods are summarized in Table 1. In addition, we also tabulate the runtime performance both in training over the entire dataset or a subsampled version, and prediction of $\hat{y}$ for a single test point x^* . Note that, for algorithms requiring a subsampled data set for tractability, the run time given is the predicted full-data run-time by considering the theoretical algorithmic runtime complexity. For stochastic methods, the prediction time is the amount of time required for the objective function to converge. We observe that kernelized ridge regression with an RBF kernel was the superior performer in terms of force tracking, despite relatively cumbersome computation complexities that were remedied via downsampling. Ridge regression on a quadratic feature space offers the best trade-off in terms of both performance and run-time.

Both kernelized ridge regression and support vector regression perform similarly in terms of force tracking capabilities given the same regularization parameter ($\lambda = 0.005$), although SVR is predictably less accurate due to sparsity in the hypothesis. We also observe reciprocal performance between the two, where kernel techniques were faster to train but slower to predict. This is to be expected, as the inherent sparsity in support vector approaches leads to more efficient hypothesis prediction. Prediction time can be sped up further by loosening the tube constraints (larger ϵ) at the cost of tracking accuracy.

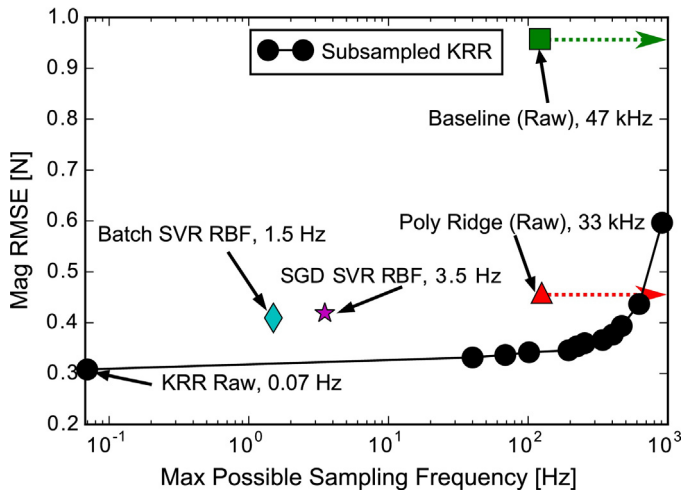


Fig. 12. Tracking performance of kernelized ridge regression (RBF kernel) vs. maximum sampling frequency ($1/t_{\text{predict}}$). Performance of other methods tested in this paper are provided for reference.

While kernel ridge regression and SVR offer the best tracking performance, there are obvious computational drawbacks that would potentially contraindicate their application in on-line implementations (density of hypothesis/prediction in the former and computationally-intensive quadratic programming in the latter). We showed that sub-gradient ϵ -SVR techniques using linear kernels can substantially improve runtime performance by several orders of magnitude both in training and prediction, however linear kernels do not result in the best tracking performance. We also derived the algorithm necessary for sub-gradient ϵ -SVR for nonlinear kernels; however, this algorithm requires a kernel evaluation for each iteration that hinders runtime performance. It is also clear that SVR offers marginal performance improvements over polynomial ridge regression despite longer runtimes. In the next section, we discuss how the prediction runtime of KRR can be improved via a downsampled training set without significantly affecting the tracking accuracy.

5.1. Implications on real-time force-feedback

We have seen that a modified model coupled with kernel-based regression techniques can improve tracking on all axes. However, we also observe in Table 1 rather long prediction runtimes when kernels are constructed from large ($n=200,000$) training sets. In robotics applications, we typically strive for sample rates on the order of hundreds of Hz for force-feedback and haptics applications [24]. Therefore it is necessary to drastically reduce the training set such that real-time prediction becomes tractable. To observe the effects of training size on tracking accuracy, we show how kernel ridge regression with an RBF kernel behaves with a subsampled training set. Subsampling the training set reduces the dimensionality of $\mathbf{K}$ and $\mathbf{k}$, thereby reducing the runtime complexity t_{predict} of predicting $\hat{y}$ at a test point x^* . In Fig. 12 we plot the overall RMSE magnitude (Eq. (15)) against the maximum sample rate for RBF kernel ridge regression with a subsampled training set. We see how tracking accuracy reduces with a lower-dimension training set, while the maximum possible sampling rate ($1/t_{\text{predict}}$) increases. We still observe that kernelized ridge regression is the superior performer over the ‘haptic regime’ (300–1000 Hz) in terms of tracking accuracy and sample rate. However, for high-speed applications (10 kHz or more), data-complete polynomial ridge regression provides the best compromise.

5.2. Future work

There are a few approximation approaches that could enable stochastic ϵ -SVR to be computationally practical in both training and prediction. In terms of training, Rahimi and Recht [25] propose a means of approximating the RBF kernel operator using a linear combination of random Fourier features (‘Random Kitchen Sinks’), reducing the computation time per iteration from $\mathcal{O}(nd)$ to $\mathcal{O}(B+d)$ (where B is the number of random features). For prediction, Claesen et al. [26] approximate the RBF kernel using a Maclaurin series approximation, reducing the prediction time complexity from $\mathcal{O}(n_{sv}d)$ to $\mathcal{O}(d^2)$. Future work could combine both of these techniques in kernelized PEGASOS to enable stochastic ϵ -SVR for RBF kernels with $\mathcal{O}((B+d)/(\lambda\epsilon))$ training complexity and $\mathcal{O}(d^2)$ prediction complexity. Additionally, newer techniques, such as Local Gaussian Process Regression, could be explored as well [27].

6. Conclusions

In this work, we experimentally demonstrate superior performance of LIM-based force sensors when environmental effects are sensed and compensated for. Using a novel, monolithic, origami-inspired manufacturing technique, we fabricate high-quality multi-axis force sensors with on-board temperature and irradiance sensing. We investigate machine learning algorithms to reject environmental disturbances in LIM force sensors based on on-board sensor data, including batch-based (linear/kernel ridge regression, support vector regression) and stochastic (SGD SVR) techniques. We show that such techniques result in drastic performance improvements over baseline results where environmental data were not accounted for. We assess prediction runtime performance of kernel-based regression techniques and show that superior performance is still achievable for high (>300 Hz) sample rates, thereby demonstrating applicability in haptic and force-feedback applications in surgical robotics. We conclude by discussing future plans to combine stochastic sub-gradient methods with nonlinear kernel approximations to further speed up run-time.

Acknowledgements

This material is based on work supported by Defense Advanced Research Projects Agency (DARPA), A2P (Grant No. FA8650-15-C-7548). This work was also partially funded by the Wyss Institute for Biologically Inspired Engineering and the John A. Paulson School of Engineering and Applied Sciences at Harvard University.

References

- [1] M. Lazeroms, G. Villavicencio, W. Jongkind, G. Honderd, Optical fibre force sensor for minimal-invasive-surgery grasping instruments, in: Proceedings of the 18th Annual International Conference of the IEEE on Engineering in Medicine and Biology Society, 1996. Bridging Disciplines for Biomedicine, vol. 1, 1996, pp. 234–235, <http://dx.doi.org/10.1109/IEMBS.1996.656931>.
- [2] P. Puangmal, H. Liu, L. Seneviratne, P. Dasgupta, K. Althoefer, Miniature 3-axis distal force sensor for minimally invasive surgical palpation, IEEE/ASME Trans. Mechatron. 17 (4) (2012) 646–656, <http://dx.doi.org/10.1109/TMECH.2011.2116033>.
- [3] Y. Noh, S. Sareh, H. Wurdemann, H. Liu, J. Back, J. Housden, K. Rhode, K. Althoefer, Three-axis fiber-optic body force sensor for flexible manipulators, Sensors J. (99) (2015), <http://dx.doi.org/10.1109/JSEN.2015.2488099>.
- [4] S.C. Ryu, P.E. Dupont, Fbg-based shape sensing tubes for continuum robots, in: 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 3531–3537, <http://dx.doi.org/10.1109/ICRA.2014.6907368>.
- [5] R.J. Roesthuis, M. Kemp, J.J. van den Dobbelsteen, S. Misra, Three-dimensional needle shape reconstruction using an array of fiber Bragg grating sensors, IEEE/ASME Trans. Mechatron. 19 (4) (2014) 1115–1126, <http://dx.doi.org/10.1109/TMECH.2013.2269836>.

- [6] J.A.C. Heijmans, M.P.H. Vleugels, E. Tabak, T.v.d. Dool, M.P. Oderwald, The impact of electrosurgical heat on optical force feedback sensors, in: Ch. 4th European Conference of the International Federation for Medical and Biological Engineering: ECIFMBE 2008, 23–27 November 2008, Antwerp, Belgium, Springer, Berlin, Heidelberg, 2009, pp. 914–917, http://dx.doi.org/10.1007/978-3-540-89208-3_218.
- [7] H. Cao, Y. Yu, Y. Ge, A research of multi-axis force sensor static decoupling method based on neural network, in: IEEE International Conference on Automation and Logistics, 2009. ICAL'09, 2009, pp. 875–879, <http://dx.doi.org/10.1109/ICAL.2009.5262800>.
- [8] J. Lei, L. Qiu, M. Liu, Q. Song, Y. Ge, Application of neural network to nonlinear static decoupling of robot wrist force sensor, in: The Sixth World Congress on Intelligent Control and Automation, 2006. WCICA 2006, Vol. 2, 2006, pp. 5282–5285, <http://dx.doi.org/10.1109/WCICA.2006.1714077>.
- [9] Z. Yao, F. Wang, W. Wang, Y. Qin, Neural-network-based six-axis force/torque robot sensor calibration, in: International Conference on Electrical and Control Engineering (ICECE), 2010, 2010, pp. 1336–1338, <http://dx.doi.org/10.1109/ICECE.2010.332>.
- [10] A. Chatterjee, S. Munshi, M. Dutta, A. Rakshit, An artificial neural linearizer for capacitive humidity sensor, in: Instrumentation and Measurement Technology Conference, 2000. IMTC 2000. Proceedings of the 17th IEEE, Vol. 1, 2000, pp. 313–317, <http://dx.doi.org/10.1109/IMTC.2000.846876>.
- [11] S. Vijayakumar, S. Schaal, Locally weighted projection regression: an $\alpha(n)$ algorithm for incremental real time learning in high dimensional spaces, in: Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Vol. 1, Stanford, CA, 2000, pp. 288–293, URL <http://www.cimc.usc.edu/publications/V/vijayakumar-ICML2000.pdf>.
- [12] J. Ma, A. Song, J. Xiao, A robust static decoupling algorithm for 3-axis force sensors based on coupling error model and ϵ -SVR, Sensors 12 (11) (2012) 14537–14555, <http://dx.doi.org/10.3390/s121114537>.
- [13] J.-r. Guo, Y.-g. He, C.-q. Liu, Nonlinear correction of photoelectric displacement sensor based on least square support vector machine, J. Central South Univ. Technol. 18 (5) (2011) 1614–1618, <http://dx.doi.org/10.1007/s11771-011-0880-6>.
- [14] X. Wang, Non-linearity estimation and temperature compensation of capacitor pressure sensors using least square support vector regression, in: IEEE International Symposium on Knowledge Acquisition and Modeling Workshop, 2008. KAM Workshop 2008, 2008, pp. 1161–1164, <http://dx.doi.org/10.1109/KAMW.2008.4810702>.
- [15] P.S. Sreetharan, J.P. Whitney, M.D. Strauss, R.J. Wood, Monolithic fabrication of millimeter-scale machines, J. Micromech. Microeng. 22 (5) (2012) 055027, URL <http://stacks.iop.org/0960-1317/22/i=5/a=055027>.
- [16] J. Gafford, S. Kesner, A. Degirmenci, R. Wood, R. Howe, C. Walsh, A monolithic approach to fabricating low-cost, millimeter-scale multi-axis force sensors for minimally-invasive surgery, in: 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 1419–1425, <http://dx.doi.org/10.1109/ICRA.2014.6907038>.
- [17] J. Gafford, R. Wood, C. Walsh, Self-assembling, low-cost, and modular mm-scale force sensor, Sensors J., IEEE 16 (1) (2016) 69–76, <http://dx.doi.org/10.1109/JSEN.2015.2476368>.
- [18] J. Gafford, S. Kesner, A. Degirmenci, R. Wood, R. Howe, C. Walsh, A monolithic approach to fabricating low-cost, millimeter-scale multi-axis force sensors for minimally-invasive surgery, in: 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 1419–1425, <http://dx.doi.org/10.1109/ICRA.2014.6907038>.
- [19] R. Penrose, A generalized inverse for matrices, Math. Proc. Cambridge Philos. Soc. 51 (3) (1954) 406–413, <http://dx.doi.org/10.1017/S0305004100030401>.
- [20] C. Saunders, A. Gammerman, V. Vovk, Ridge regression learning algorithm in dual variables, in: Proceedings of the Fifteenth International Conference on Machine Learning, ICML'98, Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, 1998, pp. 515–521, URL <http://dl.acm.org/citation.cfm?id=645527.657464>.
- [21] A. Smola, B. Schölkopf, A tutorial on support vector regression, Stat. Comput. 14 (3) (2004) 199–222, <http://dx.doi.org/10.1023/B:STCO.0000035301.49549.88>.
- [22] S. Shalev-Shwartz, Y. Singer, N. Srebro, A. Cotter, Pegasos: primal estimated sub-gradient solver for svm, Math. Progr. 127 (1) (2011) 3–30, <http://dx.doi.org/10.1007/s10107-010-0420-4>.
- [23] O. Chapelle, Training a support vector machine in the primal, Neural Comput. 19 (5) (2007) 1155–1178, <http://dx.doi.org/10.1162/neco.2007.19.5.1155>.
- [24] J. Hu, C.-Y. Chang, N. Tardella, J. Pratt, J. English, Effectiveness of haptic feedback in open surgery simulation and training systems, Stud. Health Technol. Inf. 119 (2006) 213–218, URL <http://europepmc.org/abstract/MED/16404047>.
- [25] A. Rahimi, B. Recht, Random features for large-scale kernel machines, in: Neural Information Processing Systems, 2007.
- [26] M. Claesen, F.D. Smet, J.A.K. Suykens, B.D. Moor, Fast prediction with SVM models containing RBF kernels, CoRR abs/1403.0736. URL: <http://arxiv.org/abs/1403.0736>.
- [27] D. Nguyen-Tuong, M. Seeger, J. Peters, Model learning with local gaussian process regression, Adv. Robot. 23 (15) (2009) 2015–2034, <http://dx.doi.org/10.1163/016918609X12529286896877>.

Biographies



Joshua Gafford received the B.S. degree in Mechanical Engineering from Massachusetts Institute of Technology, and M.S. degree in Mechanical Engineering from Stanford University in 2010 and 2012, respectively. He is currently pursuing his Ph.D. at the Harvard John A. Paulson School of Engineering and Applied Sciences. His research interests include design and fabrication of novel surgical end-effectors, meso-scale surgical robotic systems, sensors, and actuators using monolithic fabrication techniques.



Finale Doshi-Velez received the B.S., M.S. and Ph.D. degree from Massachusetts Institute of Technology in 2005, 2009 and 2012, respectively. Her PhD dissertation (MIT Computer Science, 2012) focused on Bayesian nonparametric approaches for reinforcement learning in partially-observable domains. She is currently an Assistant Professor of Computer Science at the Harvard John A. Paulson School of Engineering and Applied Sciences. Her research interests include Bayesian nonparametric models, sequential decision-making, and health informatics.



Robert Wood (M'01) received the M.S. and Ph.D. degrees from the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, in 2001 and 2004, respectively. He is currently the Charles River Professor of Engineering and Applied Sciences at the Harvard John A. Paulson School of Engineering and Applied Sciences, and is a Founding Core Faculty Member of the Wyss Institute for Biologically Inspired Engineering, Harvard University, Boston, MA, USA. His current research interests include microrobotics, soft material robots, and bio-inspired robotics.



Conor Walsh is an Assistant Professor of Mechanical and Biomedical Engineering at the John A. Paulson Harvard School of Engineering and Applied Sciences and a Core Faculty Member at the Wyss Institute for Biologically Inspired Engineering at Harvard University. He received his B.A.I and B.A. degrees in Mechanical and Manufacturing Engineering from Trinity College in Dublin, Ireland, in 2003, and M.S. and Ph.D. degrees in Mechanical Engineering from the Massachusetts Institute of Technology in 2006 and 2010. His current research interests focus on applying disruptive technologies to the development of robotic devices for augmenting and restoring human performance.