Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Bayesian robot system identification with input and output noise[☆]

Jo-Anne Ting[a,*], Aaron D'Souza[b], Stefan Schaal[c,d]

[a] University of British Columbia, 201-2366 Main Mall, Vancouver, BC, Canada, V6T 1Z4
[b] Google, Inc., Mountain View, CA 94043, United States
[c] University of Southern California, Los Angeles, CA, 90089, United States
[d] ATR Computational Neuroscience Laboratories, Kyoto, Japan

ARTICLE INFO

Article history:
Received 16 June 2007
Revised and accepted 26 August 2010

Keywords:
High-dimensional regression
Input noise
Variational Bayesian methods
Rigid body dynamics
Parameter identification

ABSTRACT

For complex robots such as humanoids, model-based control is highly beneficial for accurate tracking while keeping negative feedback gains low for compliance. However, in such multi degree-of-freedom lightweight systems, conventional identification of rigid body dynamics models using CAD data and actuator models is inaccurate due to unknown nonlinear robot dynamic effects. An alternative method is data-driven parameter estimation, but significant noise in measured and inferred variables affects it adversely. Moreover, standard estimation procedures may give physically inconsistent results due to unmodeled nonlinearities or insufficiently rich data. This paper addresses these problems, proposing a Bayesian system identification technique for linear or piecewise linear systems. Inspired by Factor Analysis regression, we develop a computationally efficient variational Bayesian regression algorithm that is robust to ill-conditioned data, automatically detects relevant features, and identifies input and output noise. We evaluate our approach on rigid body parameter estimation for various robotic systems, achieving an error of up to three times lower than other state-of-the-art machine learning methods.

1. Introduction

Learning the equations of motion of a complex physical system for the purpose of control is a common problem in robotics. A typical system identification approach first collects a representative data set from the robot by measuring positions and motor commands during some explorative movements. Then, we can obtain velocity and acceleration information by numerically differentiating position data. The data can also be digitally filtered to reduce noise. As a third step, we apply a function approximator to learn the mapping from positions, velocities and accelerations to motor commands. Such a function often has hundreds of inputs for complex robots. Finally, this mapping can be inserted into the control loop of the robot, where appropriate motor commands are predicted from the desired position, velocity and acceleration information—all of which are noiseless data.

The sample scenario above is representative for a large number of system identification problems. From a machine learning point of view, the interesting components are that the learning data is high dimensional, has irrelevant and redundant dimensions and, despite digital filtering, usually contains a significant amount of noise in the inputs to the function approximator. Moreover, predictions are required from *noiseless* input data, since inputs generated during control originate from a planning system without noise. The quality of control strongly depends on the quality of the learned internal model in advanced controllers and is critical in many robotic applications such as haptic devices, surgical robotics, and safe compliant assistive robots in human environments.

Ideally, system identification can be performed based on the CAD data of a robot provided by the manufacturer, at least in the context of rigid body dynamic (RBD) systems—which will be the exemplary scope of this paper. However, many modern lightweight robots such as humanoid robots have significant additional nonlinear dynamics beyond the rigid body dynamics model, due to actuator dynamics, routing of cables, use of protective shells and other sources. In such cases, instead of trying to explicitly model all possible nonlinear effects in the robot, empirical system identification methods appear to be more useful. Under the assumption that a rigid body dynamics (RBD) model is sufficient to capture the entire robot's dynamics, this problem is theoretically straightforward as all unknown parameters of the robot such as mass, center of mass and inertial parameters

[☆] This research was supported in part by National Science Foundation grants ECS-0325383, IIS-0312802, IIS-0082995, ECS-0326095, ANI-0224419, a NASA grant AC#98-516, an AFOSR grant on Intelligent Control, the ERATO Kawato Dynamic Brain Project funded by the Japanese Science and Technology Agency, and the ATR Computational Neuroscience Laboratories.

[*] Corresponding author. Tel.: +1 604 827 5554.
E-mail addresses: jting@acm.org (J.-A. Ting), adsouza@google.com (A. D'Souza), sschaal@usc.edu (S. Schaal).

0893-6080/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2010.08.011

appear linearly in the rigid body dynamics equations (An, Atkeson, & Hollerbach, 1988). Hence, after appropriate re-arrangement of the RBD equations of motion, parameter identification can be performed with linear regression techniques.

In this paper, we address the problem above in the context of linear regression, since an extension to nonlinear regression is straightforward using locally weighted learning methods (Atkeson, Moore, & Schaal, 1997). If we wanted to use traditional linear regression techniques for this scenario, we would encounter several deficiencies. First, for high dimensional robotic systems, it is not easy to generate sufficiently rich data so that all parameters will be properly identifiable. As a result, the regression problem for RBD parameter estimation is almost always numerically ill-conditioned and bears the danger of generating parameter estimates that strongly deviate from the true values, despite a seemingly low error fit of the data. For such ill-conditioned data sets in high dimensional spaces, most traditional linear regression techniques break down numerically since they are unable to generate sparse and unbiased solutions identifying redundant and/or irrelevant dimensions.

Second, sensory data collected from a robot is noisy. Noise sources exist in both input and output data, and this effect is additionally amplified by numerical differentiation to obtain derivative data. Even digital filtering will always leave some noise in the signals in order to avoid oversmoothing of data. Traditional linear regression techniques like Ordinary Least Squares (OLS) regression are only capable of dealing with noise in the output data, and the presence of input noise introduces a persistent bias to the regression solution. Alternative methods such as Total Least Squares (TLS) (Golub & Van Loan, 1989; Van Huffel & Vanderwalle, 1991) – otherwise known as orthogonal-least squares regression (Hollerbach & Wampler, 1996) or, in statistics, as errors-in-variables (EIV) when applied to a linear model (Van Huffel & Lemmerling, 2002) – address input noise, but they assume that the variances of input noise and output noise are the same (Rao & Principe, 2002). In real-world systems, this assumption is not necessarily true and, again, the resulting estimates will be biased, leading to inferior generalization.

Finally, there is no mechanism in the regression problem for RBD model identification that ensures the identified parameters are physically plausible. Particularly in the light of insufficiently rich data and nonlinearities beyond the RBD model, one often encounters physically incorrectly identified parameters such as negative values on the diagonal of an inertia matrix.

Various methods exist to deal with some of the problems mentioned above, such as regression based on singular-value decomposition (SVD) or ridge regression to cope with ill-conditioned data (Belsley, Kuh, & Welsch, 1980), stepwise regression (Draper & Smith, 1981) and LASSO (Least Absolute Shrinkage and Selection Operator) regression (Tibshirani, 1996) to produce sparse solutions, or TLS/orthogonal-least squares/EIV to address input noise (Hollerbach & Wampler, 1996). Nevertheless, a comprehensive approach addressing the entire set of issues has not been suggested so far. Recent work such as (Rao, Erdogmus, Rao, & Principe, 2003) has addressed the problem of input noise, but in the context of system identification of a time-series, while ignoring the problems associated with ill-conditioned data in high dimensional spaces. In this paper, we suggest a Bayesian estimation approach to the RBD parameter estimation problem that has all the desired properties below:

- Explicitly identifies input and output noise in the data.
- Is robust in face of ill-conditioned data.
- Detects non-identifiable parameters.
- Produces physically correct parameter estimates.

A key inspiration of our novel technique is a recently developed Bayesian machine learning framework that enables us to recast OLS regression in an advanced algorithm for input noise clean-up and numerical robustness, especially for very high dimensional estimation problems. A post-processing step ensures that the rigid body parameters are physically consistent by nonlinearly projecting the results of the Bayesian estimate onto the constraints. We will sketch the derivation of this algorithm and compare its performance with other approaches in the context of the identification of RBD parameters on synthetic data and real robotic data. On synthetic data, our algorithm achieves up to 300% improvement over other methods, and on the actual robot data we observed about 5%–25% higher accuracy in a parameter estimation problem for rigid body dynamics.

This paper is structured as follows. First, we motivate the problem of input noise in linear regression applications and identify possible solutions. Then, based on these insights, we introduce a novel estimation technique that incorporates input noise detection and uses Bayesian regularization methods to ensure robustness to ill-conditioned data. Third, we add a post-processing step to our algorithm that enforces physical correctness of the estimated RBD parameters. Finally, we evaluate our approach for parameter estimation on synthetic data and on a RBD parameter estimation for two robotic platforms: a 7 degree-of-freedom (DOF) robotic vision head and a 10 DOF robotic anthropomorphic arm.

2. High dimensional regression with input noise

Let us examine some of the problems associated with traditional system identification methods before introducing our de-noising solution. We embed our discussions in the context of RBD parameter estimation—a problem that is linear in the open parameters despite the high level of nonlinearity of the RBD equations of motion. We discuss general nonlinear system identification at the end of this paper.

The general RBD equations of motions are (Sciavicco & Siciliano, 1996):

$$\mathbf{M}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \tau \tag{1}$$

where $\mathbf{q}$, $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$ denote the vectors of joint positions, velocities, and accelerations, respectively. The matrix $\mathbf{M}(\mathbf{q})$ is the RBD inertia matrix, the matrix $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$ has terms about coriolis and centripetal forces, and the vector $\mathbf{G}(\mathbf{q})$ represents torques due to gravity. Eq. (1) has one row for every degree-of-freedom (DOF) of the robot, e.g., 30–50 rows for a humanoid robot. Every DOF is physically characterized by at least 10 parameters: a mass parameter, a center of mass vector, and a positive definite inertia matrix; friction parameters can increase the number of parameters. Thus, for robot systems with many DOFs, identifying RBD parameters is a problem involving hundreds of dimensions. Interestingly, these parameters appear linearly in Eq. (1), such that, after some complex rearrangement of the terms in Eq. (1), the system identification problem for RBD becomes a linear regression problem.

We can now switch to viewing this system identification problem from the stance of machine learning. Let us assume we have a data set $\{\mathbf{x}_i, y_i\}_{i=1}^{N}$ consisting of N samples, where $\mathbf{x}_i \in \Re^{d\times 1}$ (d is the dimensionality of the input data) and y_i is a scalar. As mentioned previously, the RBD equations can be re-arranged to yield this structure. We create a matrix $\mathbf{X} \in \Re^{N\times d}$, where the input vectors $\mathbf{x}_i$ are arranged in the rows of $\mathbf{X}$, and a vector $\mathbf{y} \in \Re^{N\times 1}$, where the corresponding scalar outputs y_i are coefficients of $\mathbf{y}$. A general model for linear regression with noise-contaminated input and output data are then:

$$\begin{aligned} y_i &= \sum_{m=1}^{d} w_{zm} t_{im} + \epsilon_{y_i} \\ x_{im} &= w_{xm} t_{im} + \epsilon_{x_{im}} \end{aligned} \tag{2}$$

where $\mathbf{t}_i$ is noiseless input data composed of t_{im} elements, $\mathbf{w}_z$ and $\mathbf{w}_x$ are regression vectors composed of w_{zm} and w_{xm}elements, respectively, and ϵ_y and ϵ_x are additive mean-zero Gaussian noise. Only $\mathbf{X}$ and $\mathbf{y}$ are observable. Note that if the input data is noiseless (i.e., $x_{im} = w_{xm}t_{im}$), we obtain the familiar linear regression equation of $y_i = \boldsymbol{\beta}_{\text{OLS}}^T \mathbf{x}_i + \epsilon_{y_i}$, where $\boldsymbol{\beta}_{\text{OLS}} = w_{zm}/w_{xm}$. The slightly more general formulation in Eq. (2) with distinct w_{xm} and w_{zm} coefficients will be useful in preparing our new algorithm.

The OLS estimate of the regression vector $\boldsymbol{\beta}_{\text{OLS}}$ is $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$, where $\boldsymbol{\beta}_{\text{OLS}}$ is composed of the parameters $\mathbf{w}_z$ and $\mathbf{w}_x$, as discussed above. The first major issue with OLS regression in high dimensional spaces is that the full rank assumption of $(\mathbf{X}^T\mathbf{X})^{-1}$ is often violated due to under-constrained data sets. For more than 500 input dimensions, the matrix inversion required in OLS also becomes rather expensive. Ridge regression can fix the problem of ill-conditioned matrices by introducing an uncontrolled amount of bias. There exist also alternative methods to invert the matrix more efficiently (Hastie & Tibshirani, 1990; Strassen, 1969), as for instance through singular value decomposition factorization (Belsley et al., 1980). Nevertheless, all these methods are unable to model noise in input data and require the manual tuning of meta parameters, which can strongly influence the quality of estimation results.

If we examine Eq. (2), we see that if the input data is noiseless (i.e., $x_{im} = w_{xm}t_{im}$), the true regression vector $\boldsymbol{\beta}_{\text{OLS}}$ will be composed of the coefficients w_{zm}/w_{xm}. This is exactly what the OLS estimate of the regression vector will be for noiseless input data. However, when the input data are contaminated with noise, it can be shown that the OLS estimate will be $\boldsymbol{\beta}_{\text{OLS,noise}} = \gamma\boldsymbol{\beta}_{\text{true}}$, where $0 < \gamma < 1$ and the exact value of γ depends on the amount of input noise. Thus, OLS regression underestimates the regression vector and produces biased predictions, a problem that cannot be fixed by adding more training data.

Intentionally, the input/output noise model formulation in Eq. (2) was chosen such that it coincides with a version of a Factor Analysis (Massey, 1965) tailored for regression problems. The intuition of this model is given in Fig. 1(a): every observed input x_{im} and output y_i is assumed to be generated by a set of hidden variables t_{im} and contaminated with some noise, as given in Eq. (2). The graphical model in Fig. 1(a) compactly describes the full multi-dimensional system: the variables x_{im}, t_{im}, w_{xm} and w_{zm} are duplicated d times for the d input dimensions of the data—as represented by the four nodes in the plate indexed by m. The other plate, indexed by i, shows that there are N samples of observed $\{\mathbf{x}_i, y_i\}$ data. The goal of learning is to find the parameters w_{xm} and w_{zm}, which can only be achieved by estimating the hidden variables t_{im} and the variances of all random variables. For technical reasons, it needs to be assumed that all t_{im} follow a Normal distribution with mean zero and unit variance, i.e., $t_{im} \sim \text{Normal}(0, 1)$, such that all parameters of the model are well-constrained (the degrees of freedom in the system needs to be constrained so that there are unique solutions for the factor loading parameters w_z and w_x).

The specific version of factor analysis for regression depicted in Fig. 1(a) is called joint-space Factor Analysis regression or Joint Factor Analysis (JFA) regression, as both input and output variables are treated the same in the estimation process (i.e., only their joint distribution matters). While Joint Factor Analysis regression is well-suited for modeling regression problems with noisy input data, it does not handle ill-conditioned data very well and is computationally expensive for high dimensions due to a repeated high dimensional matrix inversion in the ensuing iterative estimation procedure.

In the following section, we will develop a Bayesian treatment of Joint Factor Analysis regression that is robust to ill-conditioned data, automatically detects non-identifiable parameters, detects noise in input and output data and, finally, due to some post-processing, produces physically consistent parameters for RBD in a computationally inexpensive manner.

3. Bayesian parameter estimation of noisy linear regression

Fig. 1 illustrates the successive modifications of the graphical model needed to derive a Bayesian version of Joint Factor Analysis regression.

3.1. EM-based joint factor analysis regression

To start, we introduce the hidden variables z_{im} such that $z_{im} = w_{zm}t_{im}$. This trick, introduced by D'Souza, Vijayakumar, and Schaal (2004), allows us to avoid any form of matrix inversion in the resulting learning algorithm. With this modification, the noisy linear regression model in Eq. (2) becomes:

$$\begin{aligned} y_i &= \sum_{m=1}^{d} z_{im} + \epsilon_{y_i} \\ z_{im} &= w_{zm}t_{im} + \epsilon_{z_{im}} \\ x_{im} &= w_{xm}t_{im} + \epsilon_{x_{im}} \end{aligned} \tag{3}$$

where ϵ_z is additive mean-zero Gaussian noise and only $\mathbf{X}$ and $\mathbf{y}$ are observable. Due to the hidden variables z_{im} and t_{im}, we formulate the estimation of all open parameters as a maximum likelihood problem using the Expectation-Maximization (EM) algorithm (Dempster, Laird, & Rubin, 1977). For this purpose, we make the following standard assumptions about the underlying probability distributions:

$$\begin{aligned} &y_i|\mathbf{z}_i \sim \text{Normal}(\mathbf{1}^T\mathbf{z}_i, \psi_y) \\ &z_{im}|t_{im}, w_{zm} \sim \text{Normal}(w_{zm}t_{im}, \psi_{zm}) \\ &x_{im}|t_{im}, w_{xm} \sim \text{Normal}(w_{xm}t_{im}, \psi_{xm}) \\ &t_{im} \sim \text{Normal}(0, 1) \end{aligned} \tag{4}$$

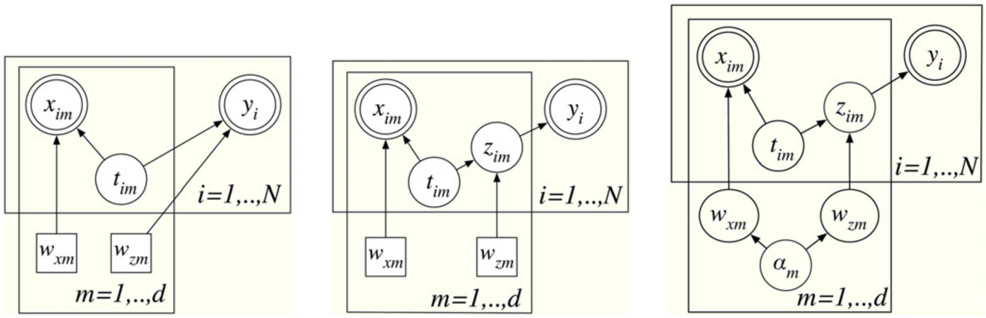
where $\mathbf{1} = [1, 1, \ldots, 1]^T$, $\mathbf{z}_i \in \Re^{d\times 1}$ is composed of z_{im} elements, $\mathbf{w}_z \in \Re^{d\times 1}$ is composed of w_{zm} elements, and $\mathbf{w}_x$, ψ_z and ψ_x are similarly composed of w_{xm}, ψ_{zm} and ψ_{xm} elements, respectively. As Fig. 1(b) shows, the regression coefficients w_{zm} are now behind the fan-in to the output y_i. This new formulation of Joint Factor Analysis regression decouples the input dimensions and generates a learning algorithm that operates with $O(d)$ computational complexity per EM iteration, where d is number of input dimensions, instead of $O(d^3)$ as in traditional Joint Factor Analysis regression.

3.2. Automatic feature detection

The efficient maximum likelihood formulation of Joint Factor Analysis regression is, however, still vulnerable to ill-conditioned data. Thus, we introduce a Bayesian layer on top of this model by treating the regression parameters $\mathbf{w}_z$ and $\mathbf{w}_x$ probabilistically to protect against overfitting, as shown in Fig. 1(c). To do this, we introduce so-called "precision" variables α_m over each regression parameter w_{zm}. The same α_m is also placed over each w_{xm}, leading to a coupled regularization of w_{zm} and w_{xm}. As a result, the regression parameters are now distributed as $w_{zm} \sim \text{Normal}(0, 1/\alpha_m)$ and $w_{xm} \sim \text{Normal}(0, 1/\alpha_m)$, where α_m takes on a Gamma distribution with parameters a_{α_m} and b_{α_m}, shown below:

$$\begin{aligned} p(\mathbf{w}_z|\boldsymbol{\alpha}) &= \prod_{m=1}^{d} \left(\frac{\alpha_m}{2\pi}\right)^{\frac{1}{2}} \exp\left\{-\frac{\alpha_m}{2}w_{zm}^2\right\} \\ p(\mathbf{w}_x|\boldsymbol{\alpha}) &= \prod_{m=1}^{d} \left(\frac{\alpha_m}{2\pi}\right)^{\frac{1}{2}} \exp\left\{-\frac{\alpha_m}{2}w_{xm}^2\right\} \\ p(\boldsymbol{\alpha}) &= \prod_{m=1}^{d} \frac{b_{\alpha_m}^{a_{\alpha_m}}}{\Gamma(a_{\alpha_m})} \alpha_m^{(a_{\alpha_m}-1)} \exp\{-b_{\alpha_m}\alpha_m\}. \end{aligned} \tag{5}$$

Please cite this article in press as: Ting, J.-A., et al. Bayesian robot system identification with input and output noise. Neural Networks (2010), doi:10.1016/j.neunet.2010.08.011



(a) Joint Factor Analysis (JFA) regression.

(b) A modified model of JFA regression for efficient estimation.

(c) A Bayesian version of JFA regression.

Fig. 1. Graphical models for noisy linear regression. Random variables are in circular nodes, observed random variables are in double circles and point estimated parameters are in square nodes. d is the total number of input dimensions while N is the total number of samples in the dataset.

The rationale of this Bayesian modeling technique is as follows. The key quantity that determines the relevance of a regression input is the parameter α_m. A priori, we assume that every w_{zm} has a mean-zero distribution with broad variance $1/\alpha_m$. We also assume that the precision α_m has an initial value 1 with large variance by setting both the initial values of a_{α_m} and b_{α_m} to 10^{-6}. If the posterior value of α_m turns out to be very large after all model parameters are estimated, then the corresponding posterior distribution of w_{zm} must be sharply peaked at zero. Thus, this gives strong evidence that $w_{zm} = 0$ and that the input t_m contributes no information to the regression model. If an input t_m contributes no information to the output, then it is also irrelevant how much it contributes to x_{im}. That is to say, the corresponding inputs x_m could be treated as pure noise. Coupling both w_{zm} and w_{xm} with the same precision variable α_m accomplishes exactly this effect. In this way, the Bayesian approach automatically detects irrelevant input dimensions and regularizes against ill-conditioned data sets.

Even with the Bayesian layer added, the entire regression problem can be treated as an EM-like learning problem (Ghahramani & Beal, 2000). Our goal is to maximize the log likelihood $\log p(\mathbf{y}|\mathbf{X})$, which is often called an "incomplete" log likelihood, as all hidden probabilistic variables are marginalized out. However, due to analytical problems, we do not have access to this incomplete log likelihood, but rather only to a lower bound of it. This lower bound is based on an expected value of the so-called "complete" data likelihood, $\langle \log p(\mathbf{y}, \mathbf{Z}, \mathbf{T}, \mathbf{w}_z, \mathbf{w}_x, \boldsymbol{\alpha}|\mathbf{X})\rangle$,[1] formulated over all variables of the learning problem, where:

$$\begin{aligned}\log p(\mathbf{y}, \mathbf{Z}, \mathbf{T}, \mathbf{w}_z, \mathbf{w}_x, \boldsymbol{\alpha}|\mathbf{X}) &= \sum_{i=1}^{N} \log p(y_i|\mathbf{z}_i)\\ &+ \sum_{i=1}^{N}\sum_{m=1}^{d} \log p(z_{im}|w_{zm}, t_{im}) + \sum_{i=1}^{N}\sum_{m=1}^{d} \log p(x_{im}|w_{xm}, t_{im})\\ &+ \sum_{i=1}^{N}\sum_{m=1}^{d} \log p(t_{im}) + \sum_{m=1}^{d} \log\{p(w_{zm}|\alpha_m)p(\alpha_m)\}\\ &+ \sum_{m=1}^{d} \log\{p(w_{xm}|\alpha_m)p(\alpha_m)\} + \mathrm{const}_{\mathbf{y},\mathbf{Z},\mathbf{T},\mathbf{w}_z,\mathbf{w}_x,\boldsymbol{\alpha}} \qquad (6)\end{aligned}$$

and where $\mathbf{Z} \in \Re^{N\times d}$ with the vector $\mathbf{z}_i$ in its rows and $\mathbf{T} \in \Re^{N\times d}$ with the vector $\mathbf{t}_i$ in its rows. The expectation of this complete data likelihood should be taken with respect to the true posterior distribution of all hidden variables $Q(\boldsymbol{\alpha}, \mathbf{w}_z, \mathbf{w}_x, \mathbf{Z}, \mathbf{T})$. Unfortunately, this is an analytically intractable expression. Instead, a lower bound can be formulated using a technique from variational calculus where we make a factorial approximation of the true posterior in terms of: $Q(\boldsymbol{\alpha}, \mathbf{w}_z, \mathbf{w}_x, \mathbf{Z}, \mathbf{T}) = Q(\boldsymbol{\alpha})Q(\mathbf{w}_z)Q(\mathbf{w}_x)Q(\mathbf{Z}, \mathbf{T})$. Such a variational factorial approximation (Ghahramani & Beal, 2000) allows us to derive analytically tractable update equations for fast, efficient inference, thus, avoiding computationally intensive Monte Carlo sampling of integrals. Variational approximations trade off accuracy over computation time. While losing a small amount of accuracy, all resulting posterior distributions over hidden variables now become analytically tractable and have the following distributions:

$$\begin{aligned}&y_i|\mathbf{z}_i \sim \mathrm{Normal}(\mathbf{1}^T\mathbf{z}_i, \psi_y)\\ &z_{im}|t_{im}, w_{zm} \sim \mathrm{Normal}(w_{zm}t_{im}, \psi_{zm})\\ &w_{zm}|\alpha_m \sim \mathrm{Normal}(0, 1/\alpha_m)\\ &w_{xm}|\alpha_m \sim \mathrm{Normal}(0, 1/\alpha_m)\\ &\alpha_m \sim \mathrm{Gamma}(\hat{a}_{\alpha_m}, \hat{b}_{\alpha_m}).\end{aligned} \qquad (7)$$

As a result, we now have a mechanism that infers the significance of each dimension's contribution to the observed output $\mathbf{y}$ and observed inputs $\mathbf{X}$.

We can derive the EM update equations using standard manipulations of Gaussian and Gamma distributions (the Gamma distribution is analytically convenient since it is a conjugate distribution for the Gaussian precision, e.g., (Ting et al., 2005), reaching the following:

E-*step*:

$$\sigma^2_{w_{zm}} = \frac{1}{\frac{1}{\psi_{zm}}\sum_{i=1}^{N}\langle t^2_{im}\rangle + \langle\alpha_m\rangle} \qquad (8)$$

$$\langle w_{zm}\rangle = \frac{\sigma^2_{w_{zm}}}{\psi_{zm}}\sum_{i=1}^{N}\langle z_{im}t_{im}\rangle \qquad (9)$$

$$\sigma^2_{w_{xm}} = \frac{1}{\frac{1}{\psi_{xm}}\sum_{i=1}^{N}\langle t^2_{im}\rangle + \langle\alpha_m\rangle} \qquad (10)$$

$$\langle w_{xm}\rangle = \frac{\sigma^2_{w_{xm}}}{\psi_{xm}}\sum_{i=1}^{N}x_{im}\langle t_{im}\rangle \qquad (11)$$

$$\hat{a}_{\alpha_m} = a_{\alpha_{m0}} + 1 \qquad (12)$$

$$\hat{b}_{\alpha_m} = b_{\alpha_{m0}} + \frac{\langle w^2_{zm}\rangle + \langle w^2_{xm}\rangle}{2} \qquad (13)$$

[1] Note that $\langle\,\rangle$ denotes the expectation operator.

Please cite this article in press as: Ting, J.-A., et al. Bayesian robot system identification with input and output noise. Neural Networks (2010), doi:10.1016/j.neunet.2010.08.011

M-*step*:

$$\psi_y = \frac{1}{N}\sum_{i=1}^{N}\left(y_i^2 - 2\mathbf{1}y_i\langle \mathbf{z}_i\rangle + \mathbf{1}^T\langle \mathbf{z}_i\mathbf{z}_i^T\rangle\mathbf{1}\right) \tag{14}$$

$$\psi_{zm} = \frac{1}{N}\sum_{i=1}^{N}\left(\langle z_{im}^2\rangle - 2\langle w_{zm}\rangle\langle z_{im}t_{im}\rangle + \langle w_{zm}^2\rangle\langle t_{im}^2\rangle\right) \tag{15}$$

$$\psi_{xm} = \frac{1}{N}\sum_{i=1}^{N}\left(x_{im}^2 - 2\langle w_{xm}\rangle\langle t_{im}\rangle x_{im} + \langle w_{xm}^2\rangle\langle t_{im}^2\rangle\right) \tag{16}$$

where the covariance matrix, $\mathbf{\Sigma}$, of the joint posterior distribution of $\mathbf{Z}$ and $\mathbf{T}$ is $\begin{bmatrix}\mathbf{\Sigma}_{zz} & \mathbf{\Sigma}_{zt}\\ \mathbf{\Sigma}_{tz} & \mathbf{\Sigma}_{tt}\end{bmatrix}$,with:

$$\mathbf{\Sigma}_{zz} = \mathbf{M} - \frac{\mathbf{M}\mathbf{1}\mathbf{1}^T\mathbf{M}}{\psi_y + \mathbf{1}^T\mathbf{M}\mathbf{1}} \tag{17}$$

$$\mathbf{\Sigma}_{tt} = \mathbf{K}^{-1} + \mathbf{K}^{-1}\langle \mathbf{W}_z\rangle^T\mathbf{\Psi}_z^{-1}\mathbf{\Sigma}_{zz}\mathbf{\Psi}_z^{-1}\langle \mathbf{W}_z\rangle\mathbf{K}^{-1} \tag{18}$$

$$\mathbf{\Sigma}_{zt} = -\mathbf{\Sigma}_{zz}\langle \mathbf{W}_z\rangle\mathbf{\Psi}_z^{-1}\mathbf{K}^{-1} \tag{19}$$

$$\mathbf{\Sigma}_{tz} = \mathbf{\Sigma}_{zt}^T \tag{20}$$

$$\mathbf{K} = \mathbf{I} + \langle \mathbf{W}_x^T\mathbf{W}_x\rangle\mathbf{\Psi}_x^{-1} + \langle \mathbf{W}_z^T\mathbf{W}_z\rangle\mathbf{\Psi}_z^{-1} \tag{21}$$

$$\mathbf{M} = \mathbf{\Psi}_z + \langle \mathbf{W}_z\rangle\left(\mathbf{I} + \langle \mathbf{W}_x^T\mathbf{W}_x\rangle\mathbf{\Psi}_x^{-1} + (\mathbf{\Sigma}_{\mathbf{W}_z})_{mm}\mathbf{\Psi}_z^{-1}\right)^{-1}\langle \mathbf{W}_z\rangle^T \tag{22}$$

and where $\langle \mathbf{W}_x\rangle$ is a diagonal d by d matrix with $\langle \mathbf{w}_x\rangle$ along its diagonal. Similarly, $\langle \mathbf{W}_z\rangle$, $\mathbf{\Psi}_x$, $\mathbf{\Psi}_z$ are d by d diagonal matrices with diagonal vectors of $\langle \mathbf{w}_z\rangle$, ψ_x and ψ_z, respectively. The E-step updates for $\mathbf{Z}$ and $\mathbf{T}$ are then:

$$\langle \mathbf{z}_i\rangle = \frac{y_i}{\psi_y}\mathbf{1}^T\mathbf{\Sigma}_{zz} + x_i\langle \mathbf{W}_x\rangle^T\mathbf{\Psi}_x^{-1}\mathbf{\Sigma}_{tz} \tag{23}$$

$$\langle \mathbf{t}_i\rangle = \frac{y_i}{\psi_y}\mathbf{1}^T\mathbf{\Sigma}_{zz}\langle \mathbf{W}_z\rangle\mathbf{\Psi}_z^{-1}\mathbf{K}^{-1} + x_i\langle \mathbf{W}_x\rangle^T\mathbf{\Psi}_x^{-1}\mathbf{\Sigma}_{tt} \tag{24}$$

$$\sigma_z^2 = \mathrm{diag}(\mathbf{\Sigma}_{zz}), \quad \sigma_t^2 = \mathrm{diag}(\mathbf{\Sigma}_{tt}), \quad \mathrm{cov}(\mathbf{z},\mathbf{t}) = \mathrm{diag}(\mathbf{\Sigma}_{zt}). \tag{25}$$

The final regression solution regularizes over the number of retained inputs in the regression vector, performing a functionality similar to Automatic Relevance Determination (ARD) (Neal, 1994). It is important to notice that *the resulting generalized EM updates still have a computational complexity of $O(d)$ for each EM iteration*—a level of efficiency that has not been accomplished with previous Joint Factor Analysis regression models, especially with one containing a full Bayesian treatment of JFA regression. Due to the three mechanisms introduced above – (i) a latent variable model to de-noise input data, (ii) an ARD framework to deal with high-dimensional input data, and (iii) a variational approximation to infer the regression solution quickly and efficiently – the result is an efficient Bayesian algorithm that is robust to high dimensional ill-conditioned noisy data.

3.3. *Inference of regression solution*

Estimating the rather complex probabilistic Bayesian model for Joint Factor Analysis regression gives us the distributions and mean values for all hidden variables. However, one additional step is required to infer the final regression parameters, which, in our application, are the RBD parameters. For this purpose, we consider the predictive distribution $p(y^q|\mathbf{x}^q)$ for a new noisy test input $\mathbf{x}^q$ and its unknown output y^q. We can calculate $\langle y^q|\mathbf{x}^q\rangle$, the mean of the distribution associated with $p(y^q|\mathbf{x}^q)$, by conditioning y^q on $\mathbf{x}^q$ and marginalizing out all hidden variables. Since an analytical solution of the resulting integral is only possible for the probabilistic Joint Factor Analysis regression model in Fig. 1(b) and not for the full Bayesian treatment, we restrict our computations to this simpler probabilistic model, and assume that $\mathbf{W}_x$ and $\mathbf{W}_z$ are replaced by their point estimates $\langle \mathbf{W}_x\rangle$ and $\langle \mathbf{W}_z\rangle$, such that our results will hold in approximation for the Bayesian model.

Thus, the predictive distribution is:

$$p(y^q|\mathbf{x}^q,\mathbf{X},\mathbf{Y}) = \iint p(y^q,\mathbf{Z},\mathbf{T}|\mathbf{x}^q,\mathbf{X},\mathbf{Y})\mathrm{d}\mathbf{Z}\mathrm{d}\mathbf{T} \tag{26}$$

where $\mathbf{X}$ and $\mathbf{Y}$ are noisy input and noisy output data used for training. From solving this integral, can infer the value of the regression estimate $\hat{\boldsymbol{\beta}}$, since $\langle y^q|\mathbf{x}^q\rangle = \hat{\boldsymbol{\beta}}^T\mathbf{x}^q$. The resulting regression estimate, given noisy inputs $\mathbf{x}^q$ and noisy outputs y^q, is $\hat{\boldsymbol{\beta}}_{\text{noise}}$:

$$\hat{\boldsymbol{\beta}}_{\text{noise}} = \frac{\psi_y\mathbf{1}^T\mathbf{B}^{-1}}{\psi_y - \mathbf{1}^T\mathbf{B}^{-1}\mathbf{1}}\mathbf{\Psi}_z^{-1}\langle \mathbf{W}_z\rangle\mathbf{A}_{\text{noise}}^{-1}\langle \mathbf{W}_x\rangle^T\mathbf{\Psi}_x^{-1} \tag{27}$$

where $\mathbf{\Psi}_x$ is a diagonal matrix with the vector ψ_x on its diagonal ($\langle \mathbf{W}_x\rangle$, $\langle \mathbf{W}_z\rangle$, $\mathbf{\Psi}_z$ are similarly defined diagonal matrices with vectors of $\langle \mathbf{w}_x\rangle$, $\langle \mathbf{w}_z\rangle$ and ψ_z on their diagonals, respectively) and:

$$\mathbf{A}_{\text{noise}} = \mathbf{I} + \langle \mathbf{W}_x^T\mathbf{W}_x\rangle\mathbf{\Psi}_x^{-1} + \langle \mathbf{W}_z^T\mathbf{W}_z\rangle\mathbf{\Psi}_z^{-1}$$
$$\mathbf{B} = \left(\frac{\mathbf{1}\mathbf{1}^T}{\psi_y} + \mathbf{\Psi}_z^{-1} - \mathbf{\Psi}_z^{-1}\langle \mathbf{W}_z\rangle^T\mathbf{A}^{-1}\langle \mathbf{W}_z\rangle\mathbf{\Psi}_z^{-1}\right).$$

If we compare $\hat{\boldsymbol{\beta}}_{\text{noise}}$ in Eq. (27) to $\hat{\boldsymbol{\beta}}_{\text{JFA}}$, the regression estimate derived for Joint Factor Analysis regression, given below:

$$\hat{\boldsymbol{\beta}}_{\text{JFA}} = \mathbf{W}_z\mathbf{A}_{\text{JFA}}^{-1}\mathbf{W}_x^T\Psi_x^{-1} \tag{28}$$
$$\mathbf{A}_{\text{JFA}} = \mathbf{I} + \mathbf{W}_x^T\mathbf{W}_x\Psi_x^{-1}\mathbf{W}_x$$

we can see that $\hat{\boldsymbol{\beta}}_{\text{noise}}$ contains an additional term $\langle \mathbf{W}_z^T\mathbf{W}_z\rangle\mathbf{\Psi}_z^{-1}$ in its $\mathbf{A}$ expression, due to the introduction of hidden variables $\mathbf{z}$. $\hat{\boldsymbol{\beta}}_{\text{noise}}$ is scaled by an additional variance-related term because of this issue as well.

It is important to note that the regression vector $\hat{\boldsymbol{\beta}}_{\text{noise}}$ given by Eq. (27) is for optimal prediction from *noisy* input data. However, for system identification in RBD, we are interested in obtaining the true regression vector, which is the regression vector that predicts the output from *noiseless* inputs. Thus, the result in Eq. (27) is not quite suitable and what we want to calculate is the mean of $p(y^q|\mathbf{t}^q)$, where $\mathbf{t}^q$ are noiseless inputs. To address this issue, we can take the limit of $\hat{\boldsymbol{\beta}}_{\text{noise}}$ by letting $\psi_x \to 0$ and interpret the resulting expression to be the true regression vector for noiseless inputs (as $\psi_x \to 0$, the amount of input noise approaches 0). The resulting regression vector estimate $\hat{\boldsymbol{\beta}}_{\text{true}}$ becomes:

$$\hat{\boldsymbol{\beta}}_{\text{true}} = \frac{\psi_y\mathbf{1}^T\mathbf{C}^{-1}}{\psi_y - \mathbf{1}^T\mathbf{C}^{-1}\mathbf{1}}\mathbf{\Psi}_z^{-1}\langle \mathbf{W}_z\rangle^T\langle \mathbf{W}_x\rangle^{-1} \tag{29}$$

where $\mathbf{C} = \left(\frac{\mathbf{1}\mathbf{1}^T}{\psi_y} + \mathbf{\Psi}_z^{-1}\right)$, and this is the desired regression vector estimate for noiseless data that we use in our evaluations.

4. Post-processing for physically consistent rigid body parameters

Before our evaluations, we need to return for a moment to the specifics of our intended application domain of RBD parameter estimation. Given a Bayesian estimate of the RBD parameters, we would like to ensure that the inferred regression vector satisfies the constraints given by positive definite inertia matrices and the parallel axis theorem (Landau & Lifschitz, 1984). In our RBD estimation problem, there are 11 RBD parameters for each DOF, which we arrange in an 11-dimensional vector

$\boldsymbol{\theta}$ consisting of the following parameters: mass, three center-of-mass coefficients multiplied by the mass and six inertial parameters. This choice of parameterization is the only one that is identifiable using linear regression (An et al., 1988). Additionally, we include viscous friction as the 11th parameter. In order to enforce the aforementioned physical constraints, we introduce a 11-dimensional virtual parameter vector $\hat{\boldsymbol{\theta}}$ that we assume is used in a nonlinear transformation to generate $\boldsymbol{\theta}$, e.g., $\boldsymbol{\theta} = f(\hat{\boldsymbol{\theta}})$. This nonlinear transformation between virtual parameters $\hat{\boldsymbol{\theta}}$ and actual parameters $\boldsymbol{\theta}$ is shown below for one DOF:

$$\begin{aligned}
\boldsymbol{\theta}_1 &= \hat{\boldsymbol{\theta}}_1^2 \\
\boldsymbol{\theta}_2 &= \hat{\boldsymbol{\theta}}_2\hat{\boldsymbol{\theta}}_1^2 \\
\boldsymbol{\theta}_3 &= \hat{\boldsymbol{\theta}}_3\hat{\boldsymbol{\theta}}_1^2 \\
\boldsymbol{\theta}_4 &= \hat{\boldsymbol{\theta}}_4\hat{\boldsymbol{\theta}}_1^2 \\
\boldsymbol{\theta}_5 &= \hat{\boldsymbol{\theta}}_5^2 + \left(\hat{\boldsymbol{\theta}}_4^2 + \hat{\boldsymbol{\theta}}_3^2\right)\hat{\boldsymbol{\theta}}_1^2 \\
\boldsymbol{\theta}_6 &= \hat{\boldsymbol{\theta}}_5\hat{\boldsymbol{\theta}}_6 - \hat{\boldsymbol{\theta}}_2\hat{\boldsymbol{\theta}}_3\hat{\boldsymbol{\theta}}_1^2 \\
\boldsymbol{\theta}_7 &= \hat{\boldsymbol{\theta}}_5\hat{\boldsymbol{\theta}}_7 - \hat{\boldsymbol{\theta}}_2\hat{\boldsymbol{\theta}}_4\hat{\boldsymbol{\theta}}_1^2 \\
\boldsymbol{\theta}_8 &= \hat{\boldsymbol{\theta}}_6^2 + \hat{\boldsymbol{\theta}}_8^2 + \left(\hat{\boldsymbol{\theta}}_2^2 + \hat{\boldsymbol{\theta}}_4^2\right)\hat{\boldsymbol{\theta}}_1^2 \\
\boldsymbol{\theta}_9 &= \hat{\boldsymbol{\theta}}_6\hat{\boldsymbol{\theta}}_7 + \hat{\boldsymbol{\theta}}_8\hat{\boldsymbol{\theta}}_9 - \hat{\boldsymbol{\theta}}_3\hat{\boldsymbol{\theta}}_4\hat{\boldsymbol{\theta}}_1^2 \\
\boldsymbol{\theta}_{10} &= \hat{\boldsymbol{\theta}}_7^2 + \hat{\boldsymbol{\theta}}_9^2 + \hat{\boldsymbol{\theta}}_{10}^2 + \left(\hat{\boldsymbol{\theta}}_2^2 + \hat{\boldsymbol{\theta}}_3^2\right)\hat{\boldsymbol{\theta}}_1^2 \\
\boldsymbol{\theta}_{11} &= \hat{\boldsymbol{\theta}}_{11}^2.
\end{aligned} \tag{30}$$

In essence, the virtual parameters $\hat{\boldsymbol{\theta}}$ correspond to the square root of the mass, the true center-of-mass coordinates (i.e., not multiplied by the mass), a Cholesky decomposition (Nash, 1990) of the DOF's inertia matrix at the center of gravity to ensure positive definiteness of the inertia matrix, and the square root of the viscous friction coefficient. The functions in Eq. (30) encode the parallel axis theorem and some additional constraints, ensuring that the mass and viscous friction coefficients remain strictly positive. Given the above formulation, any arbitrary set of virtual parameters gives rise to a physically consistent set of actual parameters for the RBD problem. For a robotic system with s DOFs, Eq. (30) is repeated for each DOF. Since there are 11 features for each DOF, the result is a $11s$-dimensional regression vector $\boldsymbol{\theta}$, where $\boldsymbol{\theta}_m = f_m(\hat{\boldsymbol{\theta}})$ (for $m = 1, \ldots, d$ where $d = 11s$).

There are at least two possible ways to enforce the physical constraints of RBD parameters in our Bayesian estimation algorithm. The first (ideal) approach involves reformulating our algorithm using the virtual parameters $\hat{\boldsymbol{\theta}}$ described previously instead of the actual parameters $\boldsymbol{\theta}$. Unfortunately, this method will lead to an analytically intractable set of update equations due to the nonlinear relationship between virtual and actual parameters. In the second approach, we can consider a post-processing step, where the unconstrained parameters are appropriately projected on to the constrained parameters. For this purpose, we assume that we would like to find the optimal virtual parameters in a least squares sense, i.e., by minimizing the cost function:

$$J = \left\langle \frac{1}{2}\left(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\right)^T\left(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\right)\right\rangle \tag{31}$$

where $\mathbf{X}$ and $\mathbf{y}$ are input and output data, and we have the constraints of $\theta_m = f_m(\hat{\boldsymbol{\theta}})$. For the moment, we will ignore issues of noise in input data and ill-conditioned data sets. Let us assume that some arbitrary estimation algorithm generated an estimate for the unconstrained parameters as $\boldsymbol{\theta}_{uc}$. Thus, the constrained parameters can be written as $\boldsymbol{\theta} = \boldsymbol{\theta}_{uc} + \Delta\boldsymbol{\theta}$, where $\Delta\boldsymbol{\theta}$ denotes the difference between constrained and unconstrained parameters. Substituting this into Eq. (31) results in:

$$\begin{aligned}
J &= \left\langle \frac{1}{2}\left(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\right)^T\left(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\right)\right\rangle \\
&= \frac{1}{2}\left\langle\left(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}_{uc}\right)^T\left(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}_{uc}\right)\right\rangle - \left\langle\left(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}_{uc}\right)^T\mathbf{X}\Delta\boldsymbol{\theta}\right\rangle \\
&\quad + \frac{1}{2}\left\langle\Delta\boldsymbol{\theta}^T\mathbf{X}^T\mathbf{X}\Delta\boldsymbol{\theta}\right\rangle.
\end{aligned} \tag{32}$$

Minimizing this cost function with respect to the virtual parameters only requires consideration of the second and third terms of Eq. (32), since the first term does not depend on the virtual parameters.

Now, let us consider algorithms to generate $\boldsymbol{\theta}_{uc}$. Among the most straightforward algorithms is OLS, which is equivalent to reformulating Eq. (31) in terms of $\boldsymbol{\theta}_{uc}$:

$$J_{uc} = \left\langle \frac{1}{2}\left(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}_{uc}\right)^T\left(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}_{uc}\right)\right\rangle, \tag{33}$$

taking the derivative $\frac{\partial J_{uc}}{\partial \boldsymbol{\theta}_{uc}}$ and setting it to zero:

$$\frac{\partial J}{\partial \boldsymbol{\theta}_{uc}} = -\left(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}_{uc}\right)^T\mathbf{X} = 0. \tag{34}$$

If we insert this result into Eq. (32), we see that the second term of this cost function equals zero, leaving only the third term to be considered in order to obtain the optimal virtual parameters. Thus, we can conclude that for optimal projection of the unconstrained parameters on to the constrained parameters, all we need to do is to minimize the difference between unconstrained and constrained parameters under the metric $\mathbf{X}^T\mathbf{X}$.

We can consider other algorithms (other than OLS) to generate $\boldsymbol{\theta}_{uc}$. For instance, SVD regression (Belsley et al., 1980) performs OLS in a subspace of the original input dimensionality of the regression problem. Thus, the cost functions in Eqs. (33) and (32) would be formulated only over the input dimensions that were identified to be relevant to the regression problem. Hence, the results regarding the minimization of the difference between unconstrained and constrained parameters hold as well.

More interestingly, if we use our Bayesian estimation method to generate $\boldsymbol{\theta}_{uc}$, the result will be similar to SVD regression in that some of the input dimensions will be eliminated. Additionally, the algorithm also estimates the noise in the inputs and returns a regression vector that can be applied to noiseless query points. If we re-express the noisy inputs $\mathbf{X}$ as $\mathbf{X}_t + \Gamma$, where $\mathbf{X}_t$ are noiseless inputs and Γ is the input noise, then we can re-write the third term of Eq. (32) in terms of de-noised quantities:

$$\frac{1}{2}\Delta\boldsymbol{\theta}^T\left(\mathbf{X}_t^T\mathbf{X}_t\right)\Delta\boldsymbol{\theta}. \tag{35}$$

The second term of Eq. (32) does not yield exactly zero as in an OLS regression, but, empirically, it is very close to zero, such that only the term in Eq. (35) matters in the actual optimization problem.

In summary, we can see that in order to minimize the least squared error in Eq. (31) with respect to the physically constrained parameters of RBD, we can follow an approximate two-step procedure. First, we apply our Bayesian algorithm (or any other algorithm, for that matter) to come up with an optimal unconstrained parameter estimate $\boldsymbol{\theta}_{uc}$. Then, we find the virtual parameter estimates $\hat{\boldsymbol{\theta}}$ (and the corresponding physically consistent parameter estimates $\boldsymbol{\theta}$) such that the error between $\boldsymbol{\theta}$ and $\boldsymbol{\theta}_{uc}$ is minimized in the sense of Eq. (35). If the noiseless inputs are not estimated explicitly, the term $\mathbf{X}_t$ is replaced by the noisy inputs $\mathbf{X}$. The optimization of Eq. (35) is easily achieved numerically as it is a simple convex function with a unique global minimum. If $\boldsymbol{\theta}_{uc}$ is estimated by OLS or SVD regression, the results for the constrained parameters are optimal. If $\boldsymbol{\theta}_{uc}$ is estimated by our Bayesian or any other nonlinear method, the results for the constrained parameters are approximately optimal. Empirically, we found that the above proposed procedure always achieves satisfying results.

5. Evaluation

We evaluated our algorithm on both synthetic data and robotic data for the task of system identification. The goal of these evaluations was to determine how well our Bayesian de-noising algorithm performs compared to other standard techniques for parameter estimation in the presence of noisy input and noisy output data.

First, we start by evaluating our algorithm on a synthetic dataset in order to illustrate its effectiveness at de-noising input and output data. Then, we apply the algorithms on a 7 DOF robotic oculomotor vision head, shown in Fig. 4, and on a 10 DOF robotic anthropomorphic arm, shown in Fig. 5, for the task of parameter estimation in rigid body dynamics.

5.1. Synthetic data set

We synthesized random input training data consisting of 10 relevant dimensions and 90 irrelevant and redundant dimensions. The first 10 input dimensions were drawn from a multi-dimensional Gaussian distribution with a random covariance matrix. The output data were generated using an ordered regression vector $\boldsymbol{\beta}_{\text{true}} = [1, 2, \ldots, 10]^T$. Output noise was added with a signal-to-noise ratio (SNR) of 5. Then, we added Gaussian noise with varying SNRs (a SNR of 2 for strongly noisy input data and a SNR of 5 for less noisy input data) to the relevant 10 input dimensions. A varying number of redundant data vectors was added to the input data, and these were generated from random convex combinations of the 10 noisy relevant data vectors. Finally, we added irrelevant data columns, drawn from a Normal(0, 1) distribution, until a total of 100 input dimensions was attained. The result was an input training dataset that contained irrelevant and redundant dimensions. Test data was created using the same method outlined above, except that input and output data were both noiseless.

We compared our Bayesian de-noising algorithm with the following methods: (i) OLS regression; (ii) stepwise regression (Draper & Smith, 1981), which tends to be inconsistent in the presence of collinear inputs (Derksen & Keselman, 1992); (iii) Partial Least Squares regression (PLS) (Wold, 1975), a slightly heuristic but empirically successful regression method for high dimensional data; (iv) LASSO regression (Tibshirani, 1996), which gives sparse solutions by shrinking certain coefficients to zero under the control of a manually set tuning parameter; (v) our probabilistic treatment of Joint Factor Analysis regression in Fig. 1(b); and (vi) our Bayesian de-noising algorithm shown in Fig. 1(c). In this synthetic evaluation, there was no need to constrain parameters according to some physical consistency rules.

The Bayesian de-noising algorithm had an improvement of 10%–300% compared to other algorithms, as the black bars in Figs. 2 and 3 illustrate. One interesting observation is that for the case where the 90 input dimensions are all irrelevant, the Bayesian de-noising algorithm did not give a significant reduction in error as in the other three scenarios. This result can be explained by the fact that the other algorithms suffer primarily from redundant inputs, but not so much from irrelevant inputs, which does not cause numerical problems. The true power of our Bayesian algorithm lies in its ability to identify the relevant dimensions in the presence of redundant and irrelevant data.

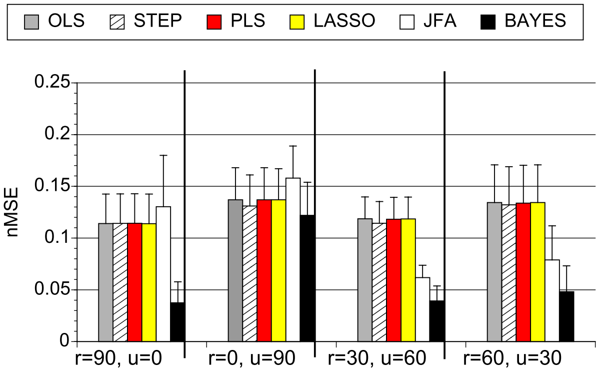


Fig. 2. Average normalized mean squared errors (nMSE) on noiseless (clean) test data for a 100 dimensional dataset with 10 relevant input dimensions and various combinations of redundant input dimensions r and irrelevant input dimensions u, averaged over 10 trials: input data has SNR $=$ 2 and output data has SNR $=$ 5. Algorithms evaluated include OLS, stepwise regression (STEP), PLS regression (PLS), LASSO regression (LASSO), Joint Factor Analysis regression (JFA) and our Bayesian de-noising algorithm (BAYES).

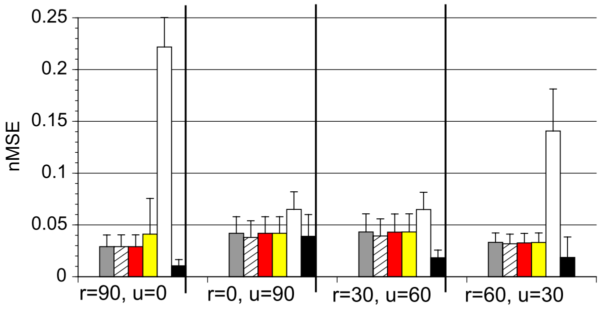


Fig. 3. Average normalized mean squared errors (nMSE) on noiseless (clean) test data for a 100 dimensional dataset with 10 relevant input dimensions and various combinations of redundant input dimensions r and irrelevant input dimensions u, averaged over 10 trials: input data has SNR $=$ 5 and output data has SNR $=$ 5. Algorithms evaluated include OLS, stepwise regression (STEP), PLS regression (PLS), LASSO regression (LASSO), Joint Factor Analysis regression (JFA) and our Bayesian de-noising algorithm (BAYES).

5.2. Robotic oculomotor vision head

Next, we move on to a 7 DOF robotic vision head manufactured by Sarcos Inc. (Salt Lake City, Utah) as shown in Fig. 4, possessing 3 DOFs in the neck and 2 DOFs for each eye. With 11 parameters per DOF, this gives a total of 77 parameters for RBD estimation. The kinematic structure of this robotic systems always creates non-identifiable parameters and thus, redundancies in the parameter estimation problem (An et al., 1988). The robot is controlled at 420 Hz with a VxWorks real-time operating system running out of a VME bus. For the training set, we collected about 500,000 data points from the robotic system while it performed sinusoidal movements with varying frequencies and phase offsets in all DOFs.

Of the six methods evaluated in the previous experiment on synthetic data, OLS regression, PLS regression and JFA regression failed to explicitly eliminate irrelevant input features and did not perform any form of reasonable parameter identification. For this reason, we omitted these three methods from the RBD identification experiments, since the parameter results produced by these methods would crash our robotic hardware. Instead, we took the remaining three methods (stepwise regression, LASSO regression and our Bayesian de-noising algorithm) and augmented each with an additional projection step such that the resulting parameter values would be physically consistent RBD parameters. We compared these three algorithms with a hand-crafted technique that consisted of OLS with ridge regression

Please cite this article in press as: Ting, J.-A., et al. Bayesian robot system identification with input and output noise. Neural Networks (2010), doi:10.1016/j.neunet.2010.08.011

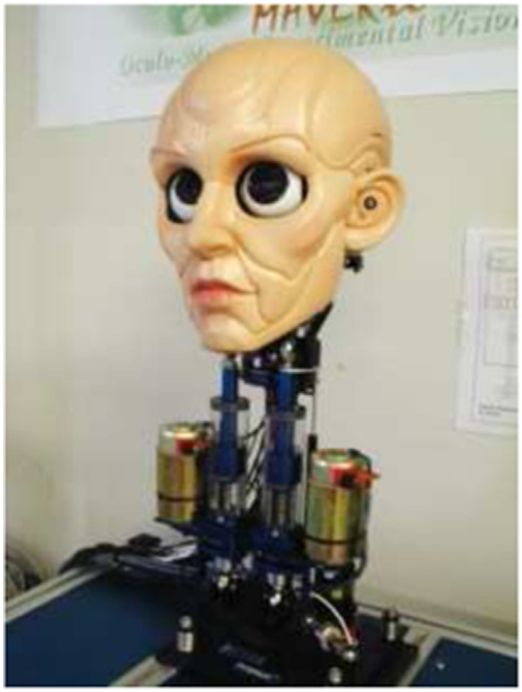


Fig. 4. Sarcos oculomotor vision head.

using a hand-tuned ridge regularization parameter. A nonlinear gradient descent method identified the virtual parameters of the system based on the unconstrained parameter estimate. All four algorithms produced physically consistent RBD parameters.

For evaluation, we implemented a computed torque control law on the robot (Sciavicco & Siciliano, 1996), using estimated parameters from each technique. Results are quantified as the root mean squared errors in position tracking, velocity tracking and the root mean squared feedback command. The left columns of Figs. 6–8 show these results averaged over all 7 DOFs of the robot's head. The Bayesian parameter estimation approach, shown in black bars, performed around 10%–25% better than ridge regression with nonlinear gradient descent. LASSO regression performed worse than ridge regression, and stepwise regression produced RBD parameters that were so physically off that they were impossible to run on the robotic head (hence, the lack of root mean squared error bars for stepwise regression in the figures). This can be explained by stepwise regression's failure to identify the relevant features in the dataset, resulting in RBD parameter values that were plainly wrong.

5.3. Robotic anthropomorphic arm

We also evaluated the parameter estimation algorithms on a 10 DOF robotic anthropomorphic arm made by Sarcos Inc. (Salt Lake City, Utah), shown in Fig. 5. With 3 DOFs in the shoulder, 1 DOF in the elbow, 3 DOFs in the wrist and 3 DOFs in the fingers, we obtained a total of 110 regression parameters. We collected about a million data points from the robotic arm over a period of 40 min, gathering data at a rate of 480 samples per second. During this time period, the arm performed sinusoidal movements with varying frequencies and phase offsets in all DOFs. We downsampled the data collected to a more manageable size of 500,000 and evaluated the algorithms in a similar approach as for the robotic vision head. The right columns of Figs. 6–8 display the results averaged over all 10 DOFs of the robot arm. The Bayesian parameter estimation approach, shown in the black bars, performed around 5%–20% better than the other techniques. LASSO regression failed, due to its over-aggressive clipping of relevant dimensions, and stepwise regression produced RBD parameters that were impossible to run on the robotic arm.

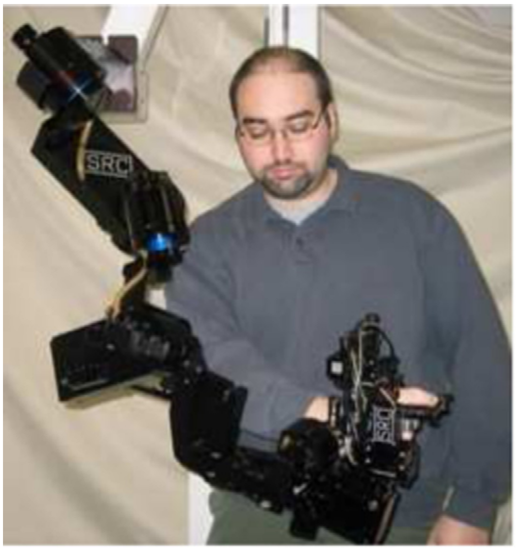


Fig. 5. Sarcos anthropomorphic arm.

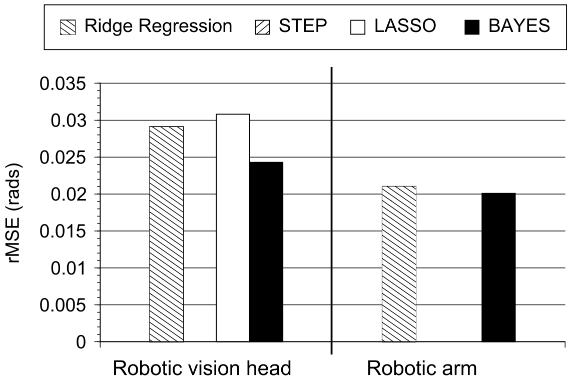


Fig. 6. Average root mean squared error (rMSE) for position in radians for the robotic vision head (averaged over all 7 DOFs) and the anthropomorphic arm (averaged over all 10 DOFs). Algorithms that do not have error bars indicate a failure to produce estimates that could be run on the robot. Algorithms evaluated include ridge regression with nonlinear gradient descent, stepwise regression with the projection step (STEP), LASSO regression with the projection step (STEP), and our Bayesian de-noising algorithm (BAYES). Standard deviations are negligible and thus omitted.

6. Discussion

This paper addresses the problem of learning for system identification, as, for example, in a scenario where we have observed a system through empirical data and would like to uncover its true parameters. Learning for system identification differs from learning for prediction. Learning for prediction is the more common problem setting in many machine learning techniques for regression. Good prediction is often possible without modeling all components of the generative system. For example, if we were solely interested in prediction, it is possible to predict well in the presence of noisy test input data by using a regression estimate that is not the true regression solution (e.g., β_{true}). However, if our intention is to estimate β_{true} and not just to predict what the output should be, then using linear regression will fail since it is not built to deal with noise in the input data.

The interesting new component of system identification – where the regression parameters are estimated – comes from

Please cite this article in press as: Ting, J.-A., et al. Bayesian robot system identification with input and output noise. Neural Networks (2010), doi:10.1016/j.neunet.2010.08.011

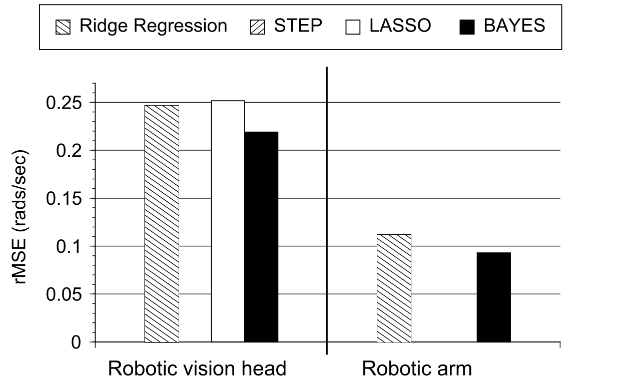


Fig. 7. Average root mean squared error (rMSE) for velocity in rad/s for the robotic vision head (averaged over all 7 DOFs) and the anthropomorphic arm (averaged over all 10 DOFs). Algorithms that do not have error bars indicate a failure to produce estimates that could be run on the robot. Algorithms evaluated include ridge regression with nonlinear gradient descent, stepwise regression with the projection step (STEP), LASSO regression with the projection step (STEP), and our Bayesian de-noising algorithm (BAYES). Standard deviations are negligible and thus omitted.

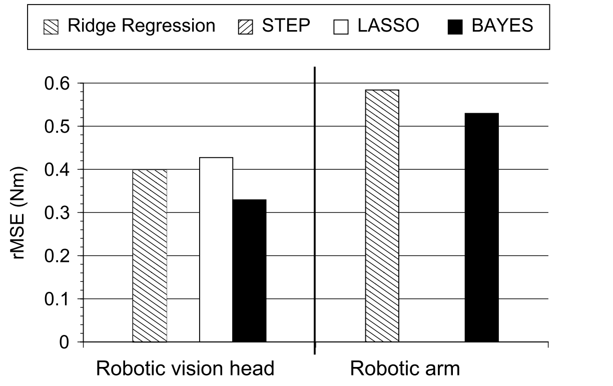


Fig. 8. Average root mean squared error (rMSE) for feedback command in Newton-meters for the robotic vision head (averaged over all 7 DOFs) and the anthropomorphic arm (averaged over all 10 DOFs). Algorithms that do not have error bars indicate a failure to produce estimates that could be run on the robot. Algorithms evaluated include ridge regression with nonlinear gradient descent, stepwise regression with the projection step (STEP), LASSO regression with the projection step (STEP), and our Bayesian de-noising algorithm (BAYES). Standard deviations are negligible and thus omitted.

the desire to use the identified model in other ways than in the training scenario. In robotics, a typical example is the use of the system model for prediction with noiseless input data. In this scenario, the training data might have been contaminated by a large amount of input noise. Another typical application is to create an analytical inverse of an identified model as often needed in model-based control. For such applications, the system model needs to be identified as accurately as possible. This is only possible if all parameters of the data generating model (in particular all noise processes) are identified accurately.

As an aside, note that if the robotic plant is changed (say, a forearm falls off or an eye stops working on the humanoid), then the current method, as well as most learning methods, would not be robust. The system will have changed drastically in structure in a way that cannot be accounted for by adding noise (or outliers, which, by definition, are infrequent, spurious data samples) to the sensory data. Our model can be adjusted to account for such failure scenarios by incorporating continuous online learning mechanisms or a switching system that switches to "failure" mode.

We address linear system identification for situations where noise exists in both input and output data—a typical case in most robotic applications where data is derived from noisy sensors. Additionally, we allow for the case of hundreds or thousands of input dimensions, where many dimensions are potentially redundant or irrelevant. To date, no efficient and robust algorithm has been suggested for such a problem setup. Inspired by factor analysis regression, a classical machine learning technique, we develop a novel full Bayesian treatment of the linear system identification problem. Due to effective Bayesian regularization, this algorithm is robust to high dimensional, ill-conditioned data with noise-contaminated input and output data and remains computationally efficient, i.e., $O(d)$ per iteration of the underlying EM-like algorithm, where d is the number of input dimensions. This algorithm has no parameters that need manual tuning.

We used this algorithm to estimate parameters in rigid body dynamics—an estimation problem that is linear in the unknown parameters. Since these parameters have a physical meaning, it was necessary to enforce physical consistent parameters with a post-processing step. The physical constraints arose from positive definiteness of inertia matrices, positiveness of mass parameters, and the parallel axis theorem. We demonstrated the efficiency of our algorithm by applying it to a synthetic dataset, a 7 DOF robotic vision head and a 10 DOF robotic anthropomorphic arm. Our algorithm successfully identified the system parameters with 10%–300% higher accuracy than alternative methods on synthetic data for parameter estimation in linear regression. It performed 5%–25% better on real robot data, proving to be a competitive alternative for parameter estimation on complex high degree-of-freedom robotic systems.

If desired, our Bayesian algorithm can easily be extended to nonlinear system identification in the framework of Locally Weighted Learning (LWL) (Atkeson et al., 1997). The only modification needed is to change the linear regression problem to a Bayesian weighted linear regression problem (Gelman, Carlin, Stern, & Rubin, 2000). Thus, a piecewise linear model identification can be achieved, similar to (Schaal & Atkeson, 1998; Vijayakumar, D'Souza, & Schaal, 2005). Parameters identified in such a nonparameteric way usually lack any physical interpretability, such that our suggested post-processing to enforce physical correctness of the parameters is not applicable. We will address the full nonlinear system identification problem in future work.

References

An, C. H., Atkeson, C. G., & Hollerbach, J. M. (1988). *Model-based control of a robot manipulator*. Cambridge, MA: MIT Press.

Atkeson, C., Moore, A., & Schaal, S. (1997). Locally weighted learning. *Artificial Intelligence Review*, *11*, 11–73.

Belsley, D. A., Kuh, E., & Welsch, R. E. (1980). *Regression diagnostics: identifying influential data and sources of collinearity*. New York: Wiley.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, *39*(1), 1–38.

Derksen, S., & Keselman, H. (1992). Backward, forward and stepwise automated subset selection algorithms: frequency of obtaining authentic and noise variables. *British Journal of Mathematical and Statistical Psychology*, *45*, 265–282.

Draper, N. R., & Smith, H. (1981). *Applied regression analysis*. New York: Wiley.

D'Souza, A., Vijayakumar, S., & Schaal, S. (2004). The Bayesian backfitting relevance vector machine. In *Proceedings of the 21st international conference on machine learning*. New York, NY, USA: ACM Press.

Gelman, A., Carlin, J., Stern, H., & Rubin, D. (2000). *Bayesian data analysis*. Chapman and Hall.

Ghahramani, Z., & Beal, M. (2000). Graphical models and variational methods. In D. Saad, & M. Opper (Eds.), *Advanced mean field methods—theory and practice*. Cambridge, MA: MIT Press.

Golub, G. H., & Van Loan, C. (1989). *Matrix computations*. Baltimore: John Hopkins University Press.

Hastie, T. J., & Tibshirani, R. J. (1990). *Monographs on statistics and applied probability*: *Vol. 43. Generalized additive models*. London, UK: Chapman and Hall.

Hollerbach, J. M., & Wampler, C. W. (1996). The calibration index and the role of input noise in robot calibration. In G. Giralt, & G. Hirzinger (Eds.), *Robotics research: the seventh international symposium* (pp. 558–568). London: Springer.

Landau, L. D., & Lifschitz, E. M. (1984). *Electrodynamics of continuous media*. Pergamon Press.

Massey, W. (1965). Principal component regression in exploratory statistical research. *Journal of the American Statistical Association*, *60*, 234–246.

Please cite this article in press as: Ting, J.-A., et al. Bayesian robot system identification with input and output noise. Neural Networks (2010), doi:10.1016/j.neunet.2010.08.011

Nash, J. C. (1990). The Cholesky decomposition. In *Compact numerical methods for computers: linear algebra and function minimization* (pp. 84–93).

Neal, R. (1994). Bayesian learning for neural networks. *Ph.D. thesis*. Dept. of Computer Science. University of Toronto.

Rao, Y. N., Erdogmus, D., Rao, G. Y., & Principe, J. (2003). Fast error whitening algorithms for system identification and control. In *Proceedings of international workshop on neural networks for signal processing* (pp. 309–318). Toulouse: IEEE.

Rao, Y. N., & Principe, J. (2002). Efficient total least squares method for system modeling using minor component analysis. In *Proceedings of international workshop on neural networks for signal processing* (pp. 259–268). Martigny: IEEE.

Schaal, S., & Atkeson, C. G. (1998). Constructive incremental learning from only local information. *Neural Computation*, *10*(8), 2047–2084.

Sciavicco, L., & Siciliano, B. (1996). *Modeling and control of robot manipulators*. MacGraw-Hill.

Strassen, V. (1969). Gaussian elimination is not optimal. *Numerische Mathematik*, *13*, 354–356.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, *58*(1), 267–288.

Ting, J., D'Souza, A., Yamamoto, K., Yoshioka, T., Hoffman, D., Kakei, S., et al. (2005). Predicting EMG data from M1 neurons with variational Bayesian least squares. In *Proceedings of advances in neural information processing systems 18*. MIT Press.

Van Huffel, S., & Lemmerling, P. (2002). *Total least squares and errors-in-variables modeling: analysis, algorithms and applications*. Dordrecht, Netherlands: Kluwer Academic Publishers.

Van Huffel, S., & Vanderwalle, J. (1991). *The total least squares problem: computational aspects and analysis*. Society for Industrial and Applied Mathematics.

Vijayakumar, S., D'Souza, A., & Schaal, S. (2005). Incremental online learning in high dimensions. *Neural Computation*, *17*, 1–33.

Wold, H. (1975). Soft modeling by latent variables: the nonlinear iterative partial least squares approach. In J. Gani (Ed.), *Perspectives in probability and statistics, papers in honor of M.S. Bartlett*. London: Academic Press.

Please cite this article in press as: Ting, J.-A., et al. Bayesian robot system identification with input and output noise. Neural Networks (2010), doi:10.1016/j.neunet.2010.08.011