A Bayesian Approach to Nonlinear Parameter Identification for Rigid Body Dynamics

Jo-Anne Ting*, Michael Mistry*, Jan Peters*, Stefan Schaal*† and Jun Nakanishi†‡

*Computer Science, University of Southern California, Los Angeles, CA, 90089-2520
†ATR Computational Neuroscience Labs, Kyoto 619-0288, Japan
‡ICORP, Japan Science and Technology Agency Kyoto 619-0288, Japan
Email: {joanneti, mmistry, jrpeters, sschaal}@usc.edu, jun@atr.jp

***Abstract*— For robots of increasing complexity such as humanoid robots, conventional identification of rigid body dynamics models based on CAD data and actuator models becomes difficult and inaccurate due to the large number of additional nonlinear effects in these systems, e.g., stemming from stiff wires, hydraulic hoses, protective shells, skin, etc. Data driven parameter estimation offers an alternative model identification method, but it is often burdened by various other problems, such as significant noise in all measured or inferred variables of the robot. The danger of physically inconsistent results also exists due to unmodeled nonlinearities or insufficiently rich data. In this paper, we address all these problems by developing a Bayesian parameter identification method that can automatically detect noise in both input and output data for the regression algorithm that performs system identification. A post-processing step ensures physically consistent rigid body parameters by nonlinearly projecting the result of the Bayesian estimation onto constraints given by positive definite inertia matrices and the parallel axis theorem. We demonstrate on synthetic and actual robot data that our technique performs parameter identification with 5 to 20% higher accuracy than traditional methods. Due to the resulting physically consistent parameters, our algorithm enables us to apply advanced control methods that algebraically require physical consistency on robotic platforms.**

I. Introduction

Advanced robot control algorithms usually rely on model-based control techniques in order to accomplish a desired level of accuracy and compliance. Typical examples include computed torque control, inverse dynamics control and operational space control [1], [2]. Depending on their sophistication, these model-based controllers also have different levels of demands on the quality of the identified robot model. For instance, computed torque control is generally rather insensitive to modeling errors, while operational space control, with its explicit use of both the rigid body dynamics inertia matrix and the centripetal/Coriolis force vector, degrades significantly in face of modeling errors. Thus, accurate model identification is a highly important topic for advanced robot control, and many modern robotics applications rely on it (e.g., as in haptic robotic devices, robotic surgery and the safe application of 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 systems—which will be the scope of this paper. However, many modern light-weight 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 dynamics, this problem is theoretically straightforward as all unknown parameters of the robot such as mass, center of mass and inertial parameters appear linearly in the rigid body dynamics equations [3]. Hence, after an appropriate re-arrangement of the RBD equations of motion, parameter identification can be performed with linear regression techniques.

Several problems, however, make this seemingly straightforward empirical RBD system identification approach more challenging. Firstly, for high dimensional robotic systems, it is not easy to generate sufficiently rich data so that all parameters will be properly identifiable. Secondly, 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 done to obtain derivative data from the sensors. Traditional linear regression techniques are only capable of dealing with output noise, and the presence of input noise introduces a persistent bias to the regression solution. Digital filtering of the data can eliminate some noise, but it often eliminates important structure as well. Techniques exist such as Total Least Squares (TLS) [4], [5], otherwise known as orthogonal least-squares regression [6], to perform parameter estimation, but it assumes that the variances of input noise and output noise are the same [7]. In real-world systems where this assumption does not hold, the resulting parameter estimates will be biased [8]. Thirdly, 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 inertial matrix. Using physically incorrect data in model-based control leads to dangerously unstable controllers. The final problem with empirical RBD system identification is that some RBD parameters of a robot are not identifiable at all [3].

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.

Various methods exist to deal with some of the problems mentioned above such as regression based on singular-value decomposition (SVD), ridge regression to cope with the ill-conditioned data [9], or TLS and orthogonal-least squares to address input noise [6]. Nevertheless, a comprehensive approach to address the entire set of issues has not been suggested so far. Recent work such as [10] 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:

- 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 component of our technique is a recently developed Bayesian machine learning framework that enables us to recast ordinary least squares (OLS) regression in a more 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 results with other approaches in the context of identification of RBD parameters on synthetic data and on a robotic vision head. On average, our algorithm achieves a 5 to 20% improvement when compared to other standard techniques for the identification of RBD parameters.

The remaining paper is structured as follows. First, we motivate the problem of input noise in linear regression problems and outline possible solutions. Then, based on these insights, we develop a novel estimation technique that incorporates input noise detection and employs Bayesian regularization methods to ensure robustness of the algorithm for 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 RBD parameter estimation on synthetic data, on a 7 degree-of-freedom (DOF) robotic vision head and on a 10 DOF robotic anthropomorphic arm.

II. High Dimensional Regression with Input Noise

Before describing our solution to RBD parameter identification, it is useful to examine some of the problems associated with traditional estimation methods. For robot systems with many DOFs, the linear regression problem for identifying RBD parameters has hundreds of dimensions, i.e., at least 10 parameters for every DOF. Hence, estimation algorithms need to be suitable for this dimensionality. The re-arrangement of the RBD equations for parameter estimation creates a matrix where the input vectors $\mathbf{x}$ are arranged in the rows of the matrix $\mathbf{X}$ and the corresponding scalar outputs y are the coefficients of the vector $\mathbf{y}$. A general model for such linear regression with noise-contaminated input and output data is:

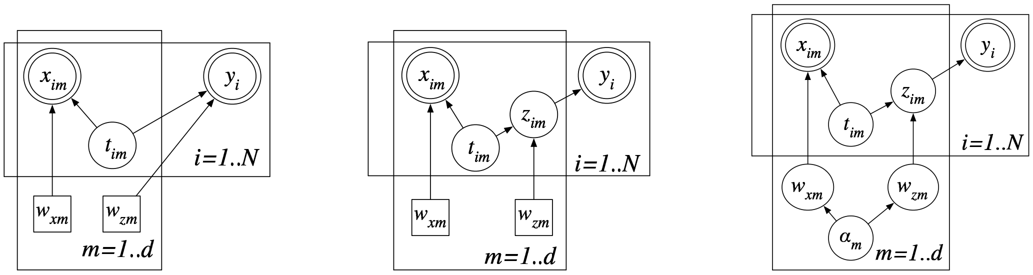
$$\begin{aligned} y &= \sum_{m=1}^{d} w_{zm} t_m + \epsilon_y \\ x_m &= w_{xm} t_m + \epsilon_{xm} \end{aligned} \tag{1}$$

where d is the number of input dimensions, $\mathbf{t}$ is noiseless input data composed of t_m components, $\mathbf{w_z}$ and $\mathbf{w_x}$ are regression vectors composed of w_{zm} and w_{xm} components respectively, and ϵ_y and ϵ_x are additive mean-zero noise. Only $\mathbf{X}$ and y are observable. Note that if the input data is noiseless (that is, $x_m = w_{xm} t_m$) and if we set $w_{xm} = 1$, then we obtain the familiar linear regression equation of $y = \mathbf{w_z}^T x + \epsilon_y$ for noiseless input data and noisy output data. The slightly more general formulation above will be useful in preparing our novel estimation algorithm.

The OLS estimate of the regression vector β_{OLS} is $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$, where β_{OLS} is composed of the parameters $\mathbf{w_z}$ and $\mathbf{w_x}$, as discussed in the following paragraph. 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 underconstrained datasets. 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 [11], [12], as for instance through singular value decomposition factorization (e.g., using the Matlab pinv() function). 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 the estimation results. Moreover, in the case of RBD parameter estimation, there is no mechanism that ensures the parameter estimates are physically consistent.

If we examine Eq. (1), we see that if the input data is noiseless (that is, $x_m = w_{xm} t_m$), the true regression vector β_{true} 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 is contaminated with noise, it can be shown that the OLS estimate will be $\beta_{OLS,noise} = \gamma\beta_{true}$, where $0 < \gamma < 1$, where its exact value depends on the amount of input noise. Thus, OLS regression underestimates the true regression vector β_{true}. For the application of RBD parameter estimation, we obtain a persistent bias in the model identification such that the model is inaccurate, and this is a problem that cannot be fixed by simply adding more data.

Intentionally, the input/output noise model formulation in Eq. (1) was chosen such that it coincides with a version of a well-known algorithm in the statistical learning community called Factor Analysis [13]. The intuition of this model is



(a) Joint Factor Analysis (JFA)

(b) A modified model of JFA for efficient estimation

(c) A Bayesian version of JFA

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.

given in Figure 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, exactly as given in Eq. (1). The graphical model in Figure 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—represented by the inner plate containing four nodes. The outer plate shows that there are N samples of observed $\{\mathbf{x}, y\}$ 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 all noise processes as well. 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 specific version of factor analysis for regression depicted in Figure 1(a) is called joint-space Factor Analysis or Joint Factor Analysis (JFA), as both input and output variables are actually treated the same in the estimation process, i.e., only their joint distribution matters. While Joint Factor Analysis is well suited for modeling regression models with input noise, it does not handle ill-conditioned data very well and is computationally very expensive for high dimensions.

In the following section, we will develop a Bayesian treatment of Joint Factor Analysis that is robust to ill-conditioned data, automatically detects non-identifiable parameters and, due to some post-processing, produces physically consistent parameters for RBD—all in a computationally efficient way.

III. Bayesian Parameter Estimation of Noisy Linear Regression

Figure 1 demonstrates the successive modifications of the graphical model needed to derive a Bayesian version of Joint Factor Analysis regression. As a first step, we introduce the hidden variables z_{im} such that $z_{im} = w_{zm} t_{im}$. This trick was introduced by [14] and allows us to avoid any form of matrix inversion in the resulting learning algorithm. With this modification, the noisy linear regression model in Eq. (1) is modified to become:

$$y_i = \sum_{m=1}^{d} z_{im} + \epsilon_y \tag{2}$$

Due to the hidden variables z_{im} and t_{im}, we formulate the estimation of all open parameters as a maximum likelihood estimation problem using the Expectation-Maximization (EM) algorithm [15]. For this purpose, the following standard assumptions about the probability distributions of the random variables are made:

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

where $\mathbf{1} = [1, 1, ...1]^T$, $\mathbf{z_i}$ is a d by 1 vector, $\mathbf{w_z}$ is a d by 1 vector composed of w_{zm} elements and $\mathbf{w_x}$, $\psi_\mathbf{z}$, and $\psi_\mathbf{x}$ are similarly composed of w_{xm}, ψ_{zm} and ψ_{zm} elements, respectively. As Figure 1(b) shows, the regression coefficients w_{zm} are now behind the fan-in to the output y_i. This decouples the input dimensions and generates a learning algorithm that operates with $O(d)$ computational complexity, instead of $O(d^3)$ as in traditional Joint Factor Analysis.

The efficient maximum likelihood formulation of Joint Factor Analysis 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 Figure 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 used for each w_{xm}. As a result, the regression parameters are now distributed as follows: $w_{zm} \sim \text{Normal}(0, 1/\alpha_m)$ and $w_{xm} \sim \text{Normal}(0, 1/\alpha_m)$, where α_m takes on a Gamma distribution. 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 variance $1/\alpha_m$. If the value of an 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 giving 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}. Hence, 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 achieves this effect. In this way, the Bayesian approach automatically detects irrelevant input dimensions and regularizes against ill-conditioned datasets.

Even with the Bayesian layer added, the entire regression problem can be treated as an EM-like learning problem [16]. Given the data $D = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$, 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 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, $\langle \log p(\mathbf{y}, \mathbf{Z}, \mathbf{T}, \mathbf{w}_z, \mathbf{w}_x, \alpha, |\mathbf{X}) \rangle$, formulated over all variables of the learning problem, where:

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

The expectation of this complete data likelihood should be taken with respect to the true posterior distribution of all hidden variables $Q(\alpha, \mathbf{w_z}, \mathbf{w_x}, \mathbf{Z}, \mathbf{T})$. 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: $Q(\alpha, \mathbf{w_z}, \mathbf{w_x}, \mathbf{Z}, \mathbf{T}) = Q(\alpha)Q(\mathbf{w_z})Q(\mathbf{w_x})Q(\mathbf{Z}, \mathbf{T})$. While losing a small amount of accuracy, all resulting posterior distributions over hidden variables become analytically tractable and have the following distributions:

$$\begin{aligned} y_i|z_i &\sim \text{Normal}(\mathbf{1}^T\mathbf{z_i}, \psi_y) \\ z_{im}|w_{zm} &\sim \text{Normal}(w_{zm}t_{im}, \psi_{zm}) \\ w_{zm}|\alpha_m &\sim \text{Normal}(0, 1/\alpha_m) \\ w_{xm}|\alpha_m &\sim \text{Normal}(0, 1/\alpha_m) \\ \alpha_m &\sim \text{Gamma}(a_{\alpha_m}, b_{\alpha_m}) \end{aligned}$$

As a final result, we now have a mechanism that infers the significance of each dimension's contribution to the observed output y and observed inputs $\mathbf{x}$. The resulting EM update equations are list in the Appendix. The final regression solution regularizes over the number of retained inputs in the regression vector, performing a functionality similar to Automatic Relevance Determination (ARD) [17]. Notably, the EM updates have a computation complexity of $O(d)$ per EM iteration, where d is the number of input dimensions, instead of the $O(d^3)$ of Joint Factor Analysis that arises due to a matrix inversion. The final result is an efficient Bayesian algorithm that is robust to high dimensional ill-conditioned noisy data.

A. Inference of Regression Solution

Estimating the rather complex probabilistic Bayesian model for Joint Factor Analysis reveals distributions and mean values for all hidden variables. One additional step, however, 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 to obtain:

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

where $\mathbf{X}$ and $\mathbf{Y}$ are the data used for training. We can infer the value of the regression estimate $\hat{b}$, since $\langle y^q|\mathbf{x}^q \rangle = \hat{b}^T\mathbf{x}^q$. Since an analytical solution for the integral above is only possible for the probabilistic Joint Factor Analysis model in Figure 1(b) and not the full Bayesian treatment, we restrict our computations to the simpler probabilistic model, assuming that the results will hold in approximation for the Bayesian model. The resulting regression estimate, given noisy inputs $\mathbf{x}^q$ and noisy outputs y^q, is $\hat{b}_{noise}$:

$$\hat{b}_{noise} = \frac{\psi_y \mathbf{1}^T\mathbf{B}^{-1}}{\psi_y - \mathbf{1}^T\mathbf{B}^{-1}\mathbf{1}} \mathbf{\Psi}_z^{-1} \langle \mathbf{W}_z \rangle \mathbf{A}^{-1} \langle \mathbf{W}_x \rangle^T \mathbf{\Psi}_x^{-1} \quad (3)$$

where $\mathbf{\Psi}_x$ is a diagonal matrix with the vector $\psi_\mathbf{x}$ 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 $\psi_\mathbf{z}$ on their diagonals, respectively), $A = \left(\mathbf{I} + \left\langle \mathbf{W}_x^T\mathbf{W}_x \right\rangle \mathbf{\Psi}_x^{-1} + \left\langle \mathbf{W}_z^T\mathbf{W}_z \right\rangle \mathbf{\Psi}_z^{-1}\right)$ and $B = \left(\frac{\mathbf{1}\mathbf{1}^T}{\psi_y} + \mathbf{\Psi}_z^{-1} - \mathbf{\Psi}_z^{-1} \langle \mathbf{W}_z \rangle^T \mathbf{A}^{-1} \langle \mathbf{W}_z \rangle \mathbf{\Psi}_z^{-1}\right)$. Note that Eq. (3) is similar in form to the regression estimate derived for Joint Factor Analysis regression (which can be found in the Appendix). The major difference is that Eq. (3) contains an additional term $\left\langle \mathbf{W}_z^T\mathbf{W}_z \right\rangle \mathbf{\Psi}_z^{-1}$, due to the introduction of hidden variables z. The regression estimate is scaled by an additional term as well.

It is important to note that the regression vector given by Eq. (3) 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 output from *noiseless* inputs. Thus, the result in Eq. (3) 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 asymptotic estimate of Eq. (3) by letting $\psi_\mathbf{x} \rightarrow 0$ and interpret the resulting expression to be the true regression vector for noiseless inputs (as $\psi_\mathbf{x} \rightarrow 0$, the amount of input noise approaches 0). The resulting regression vector estimate $\hat{b}_{true}$ becomes:

$$\hat{b}_{true} = \frac{\psi_y \mathbf{1}^T\mathbf{C}^{-1}}{\psi_y - \mathbf{1}^T\mathbf{C}^{-1}\mathbf{1}} \mathbf{\Psi}_z^{-1} \langle \mathbf{W}_z \rangle^T \langle \mathbf{W}_x \rangle^{-1} \quad (4)$$

where $\mathbf{C} = \left(\frac{\mathbf{1}\mathbf{1}^T}{\psi_y} + \mathbf{\Psi}_z^{-1}\right)$, which is the desired regression vector estimate.

B. Post-processing for Physically Consistent Rigid Body Parameters

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. These constraints are shown in Eq. (5) for one DOF. There are 11 parameters for each DOF, which we arrange in an 11-dimensional vector θ consisting of the following 10 parameters: mass, three center of mass coefficients multiplied by the mass and six inertial parameters (cf. [3]). Additionally, we include viscous friction as the 11th parameter. Now, this parameter vector θ is assumed to be generated through a nonlinear transformation from a 11-dimensional virtual parameter vector $\hat{\theta}$. In essence, these virtual parameters $\hat{\theta}$ correspond to the square root of the mass, the true center-of-mass coordinates (i.e., not multiplied by the mass), the six inertial parameters describing the inertia matrix at the DOF's center of gravity, and the square root of the viscous friction coefficient. The following functions show the relationship between virtual parameters $\hat{\theta}$ and actual parameters θ:

$$\begin{aligned}
\theta_1 &= \hat{\theta}_1^2, \theta_2 = \hat{\theta}_2\hat{\theta}_1^2, \theta_3 = \hat{\theta}_3\hat{\theta}_1^2, \theta_4 = \hat{\theta}_4\hat{\theta}_1^2, \theta_{11} = \hat{\theta}_{11}^2 \\
\theta_5 &= \hat{\theta}_5^2 + \left(\hat{\theta}_4^2 + \hat{\theta}_3^2\right)\hat{\theta}_1^2 \\
\theta_6 &= \hat{\theta}_5\hat{\theta}_6 - \hat{\theta}_2\hat{\theta}_3\hat{\theta}_1^2, \theta_7 = \hat{\theta}_5\hat{\theta}_7 - \hat{\theta}_2\hat{\theta}_4\hat{\theta}_1^2 \\
\theta_8 &= \hat{\theta}_6^2 + \hat{\theta}_8^2 + \left(\hat{\theta}_2^2 + \hat{\theta}_4^2\right)\hat{\theta}_1^2 \\
\theta_9 &= \hat{\theta}_6\hat{\theta}_7 + \hat{\theta}_8\hat{\theta}_9 - \hat{\theta}_3\hat{\theta}_4\hat{\theta}_1^2 \\
\theta_{10} &= \hat{\theta}_7^2 + \hat{\theta}_9^2 + \hat{\theta}_{10}^2 + \left(\hat{\theta}_2^2 + \hat{\theta}_3^2\right)\hat{\theta}_1^2
\end{aligned} \tag{5}$$

These functions encode the parallel axis theorem and some additional constraints, essentially ensuring that 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. (5) is repeated for each DOF. The result is a regression vector θ, where $\theta_m = f_m(\hat{\theta})$ (for $m = 1..d$ where $d = 11s$ and s is the number of DOFs in the system).

Our Bayesian parameter estimation method (as well as any other traditional RBD parameter estimation method) generates the parameter vector θ, not the virtual parameters $\hat{\theta}$. We want to find the optimal parameters $\theta_{opt,RBD}$ that satisfy the physical RBD constraints while minimizing the least squared error in the prediction. That is to say, we want to minimize:

$$\left\langle \frac{1}{2}\left(\mathbf{y} - \mathbf{X}\theta_{opt,RBD}\right)^T\left(\mathbf{y} - \mathbf{X}\theta_{opt,RBD}\right)\right\rangle \tag{6}$$

We can re-express $\theta_{opt,RBD}$ as $(\theta_{Bayes} - \Delta\theta)$. Eq. (6) can then be simplified in order to obtain:

$$\begin{aligned}
&\left\langle \frac{1}{2}\left(\mathbf{y} - \mathbf{X}\theta_{opt,RBD}\right)^T\left(\mathbf{y} - \mathbf{X}\theta_{opt,RBD}\right)\right\rangle \\
&= \frac{1}{2}\left\langle\left(\mathbf{y} - \mathbf{X}\theta_{Bayes}\right)^T\left(\mathbf{y} - \mathbf{X}\theta_{Bayes}\right)\right\rangle \\
&+ \left\langle\left(\mathbf{y} - \mathbf{X}\theta_{Bayes}\right)^T\mathbf{X}\Delta\theta\right\rangle + \frac{1}{2}\left\langle\Delta\theta^T\mathbf{X}^T\mathbf{X}\Delta\theta\right\rangle
\end{aligned} \tag{7}$$

Notice that the minimization of the first term $\left\langle\left(\mathbf{y} - \mathbf{X}\theta_{Bayes}\right)^T\left(\mathbf{y} - \mathbf{X}\theta_{Bayes}\right)\right\rangle$ is what our Bayesian EM-based algorithm does, since it finds the unconstrained parameter estimates that minimize the least squares error using the noisy input and output data. The second term in Eq. (7) is equal to 0 (refer to the Appendix for details). Now, if we re-express the noisy inputs $\mathbf{X}$ as $\mathbf{X}_t + \mathbf{\Gamma}$, where $\mathbf{X}_t$ are noiseless inputs and $\mathbf{\Gamma}$ is the input noise, then we can re-write the third term in Eq. (7) as:

$$\begin{aligned}
&\frac{1}{2}\left\langle\Delta\theta^T\mathbf{X}^T\mathbf{X}\Delta\theta\right\rangle \\
&= \frac{1}{2}\Delta\theta^T\mathbf{X}^T\mathbf{X}\Delta\theta + \Delta\theta^T\mathbf{X}_t^T\left\langle\mathbf{\Gamma}\right\rangle\Delta\theta + \frac{1}{2}\Delta\theta^T\left\langle\mathbf{\Gamma}^T\mathbf{\Gamma}\right\rangle\Delta\theta \\
&= \frac{1}{2}\Delta\theta^T\left(\mathbf{X}^T\mathbf{X} + \left\langle\mathbf{\Gamma}^T\mathbf{\Gamma}\right\rangle\right)\Delta\theta
\end{aligned} \tag{8}$$

since the input noise is modeled with a Gaussian distribution with a mean of zero and some variance $\mathbf{\Psi}_x$. Hence, as the above equation shows, to minimize the third term in Eq. (7), we need to find the parameter estimate $\theta_{opt,RBD}$ that is closest to θ_{Bayes}, under the metric $\mathbf{X}^T\mathbf{X}$ (plus noise variance), as possible. In summary, we can see that in order to minimize the least squared error in Eq. (6), we can do so in two independent minimization steps. First, we apply our Bayesian algorithm (or any other algorithm, for that matter) to come up with an optimal unconstrained parameter estimate. Then, we find the physically consistent parameter estimates $\theta_{opt,RBD}$ such that the error between $\theta_{opt,RBD}$ and the optimal unconstrained parameter estimates is minimized in the sense of Eq. (8).

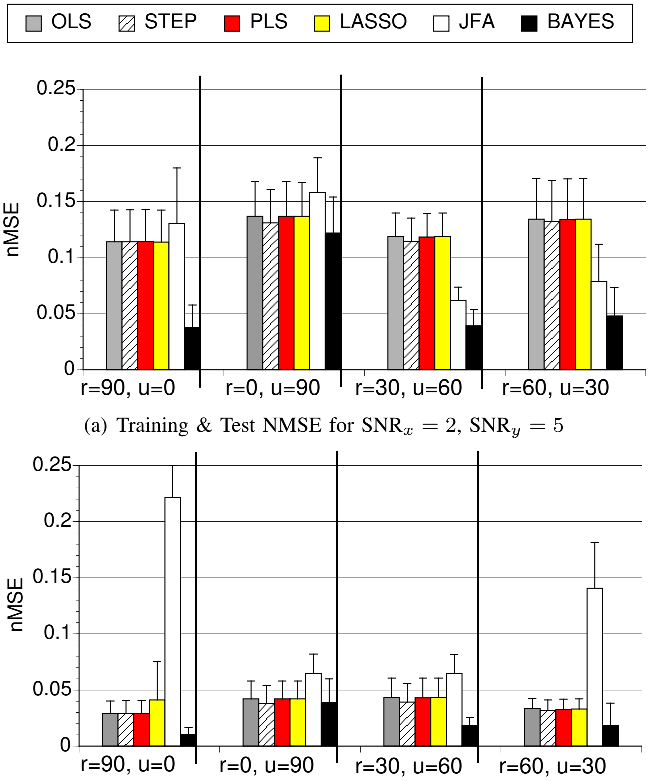
IV. Evaluation

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 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 and on a 10 DOF robotic anthropomorphic arm for the task of RBD parameter estimation.

A. 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



(a) Training & Test NMSE for $\text{SNR}_x = 2$, $\text{SNR}_y = 5$

(b) Training & Test NMSE for $\text{SNR}_x = 5$, $\text{SNR}_y = 5$

Fig. 2. Average normalized mean squared errors on noiseless (clean) test data and noisy 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, for different levels of noisy data

matrix. The output data was generated using an ordered regression vector $b_{true} = [1, 2, ..., 10]^T$. A signal-to-noise ratio (SNR) of 5 was added to the outputs. 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 dimension were 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. A second test dataset consisting of noisy input and output data, possessing the same noise characteristics as the training dataset, was also generated.

We compared our Bayesian de-noising algorithm with the following methods: OLS regression; stepwise regression [18], which tends to be inconsistent in the presence of collinear inputs [19]; Partial Least Squares regression (PLS) [20], a slightly heuristic but empirically successful regression method for high dimensional data; LASSO regression [21], which gives sparse solutions by shrinking certain coefficients to 0 under the control of a manually set tuning parameter; our probabilistic treatment of Joint Factor Analysis in Figure 1(b); and our Bayesian de-noising algorithm shown in Figure 1(c).

The Bayesian de-noising algorithm had an improvement of 10 to 20% compared to other algorithms for strongly noisy input data and an improvement of 7 to 50% for less noisy input data, as the black bars in Figures 2(a) and 2(b) illustrate. One interesting observation is that for the case where the 90 input dimensions are all irrelevant, the Bayesian de-noising algorithm does not give such a significant reduction in error as in the other 3 scenarios. This can be explained by the fact that the other algorithms cannot handle data containing redundancy, and that the true power of our algorithm lies in its ability to identify relevant dimensions in the presence of redundant and irrelevant data.

B. *Robotic Oculomotor Vision Head*

Next, we move on to a 7 DOF robotic vision head manufactured by Sarcos as shown in Figure 3, possessing 3 DOFs in the neck and 2 DOFs for each eye. With 11 features per DOF, this gives a total of 77 features. This kinematic structure of robotic systems always creates non-identifiable parameters and thus, redundancies [3]. The robot is controlled at 420 Hz with a VxWorks real-time operating system running out of a VME bus. 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.

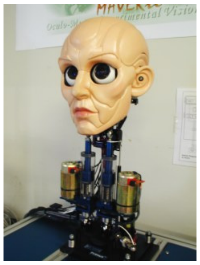


Fig. 3. Sarcos Oculomotor Vision Head

We compared our Bayesian algorithm with 3 other techniques for parameter estimation on the robot data. The first technique consisted of ridge regression using a hand-tuned regularization parameter with nonlinear gradient descent performed on the virtual parameters of the system. The second algorithm was a version of LASSO regression that had the additional step of projecting the resulting parameter values onto the constraint space in order to produce physically consistent RBD parameters. Finally, the last algorithm was a version of stepwise regression with the additional projection step. All four algorithms produced physically consistent RBD parameters. Note that the other algorithms used in the synthetic dataset like PLS and JFA were not applied, since they fail to explicitly eliminated irrelevant input features and do not perform any form of reasonable parameter identification.

For evaluation, we implemented a computed torque control law on the robot, 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. Table I shows these results averaged over all 7 DOFs. The Bayesian parameter estimation approach performed around 10 to 20% better than the ridge regression with gradient descent approach, thus validating the effectiveness of our methods. LASSO regression performed worse than ridge regression with gradient descent. As well,

	Position (radians)	Velocity (radians/sec)	Feedback Command (Newton-meter)
Ridge Regression	0.0291	0.2465	0.3969
Bayesian De-noising	0.0243	0.2189	0.3292
LASSO regression	0.0308	0.2517	0.4272
Stepwise regression	FAILURE	FAILURE	FAILURE

TABLE I

ROOT MEAN SQUARED ERRORS FOR POSITION (IN RADIANS), VELOCITY (IN RADIANS/SEC) AND FEEDBACK COMMAND (IN NEWTON-METERS) FOR THE SARCOS ROBOTIC VISION HEAD. ALGORITHMS EVALUATED INCLUDE RIDGE REGRESSION WITH NONLINEAR GRADIENT DESCENT, OUR BAYESIAN DE-NOISING ALGORITHM, LASSO REGRESSION WITH THE PROJECTION STEP, AND STEPWISE REGRESSION WITH THE PROJECTION STEP. STANDARD DEVIATIONS ARE NEGLIGIBLE AND THUS OMITTED.

	Position (radians)	Velocity (radians/sec)	Feedback Command (Newton-meter)
Ridge Regression	0.0210	0.1119	0.5839
Bayesian De-noising	0.0201	0.0930	0.5297
LASSO regression	FAILURE	FAILURE	FAILURE
Stepwise regression	FAILURE	FAILURE	FAILURE

TABLE II

ROOT MEAN SQUARED ERRORS FOR POSITION (IN RADIANS), VELOCITY (IN RADIANS/SEC) AND FEEDBACK COMMAND (IN NEWTON-METERS) FOR THE SARCOS ROBOTIC ANTHROPOMORPHIC ARM. ALGORITHMS EVALUATED INCLUDE RIDGE REGRESSION WITH NONLINEAR GRADIENT DESCENT, OUR BAYESIAN DE-NOISING ALGORITHM, LASSO REGRESSION WITH THE PROJECTION STEP, AND STEPWISE REGRESSION WITH THE PROJECTION STEP. STANDARD DEVIATIONS ARE NEGLIGIBLE AND THUS OMITTED.

stepwise regression produced RBD parameters that were so physically off that they were impossible to run on the robotic head. This can be explained by stepwise regression's failure to identify the relevant features in the dataset, resulting in RBD parameter values that were plain wrong.

C. Robotic Anthropomorphic Arm

We also evaluated the parameter estimation algorithms on a 10 DOF robotic anthropomorphic arm made by Sarcos, shown in Fig. 4. With 3 DOFs in the shoulder, 1 DOF in the elbow, 3 DOFs in the wrist and 3 DOFs in the fingers, this gives a total of 110 features. We collected about a million data points from the robotic arm over a period of 40 minutes, 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 500000 and evaluated the algorithms in a similar approach as for the robotic vision head. Table II displays the results averaged over all 10 DOFs. The Bayesian parameter estimation approach performed around 5 to 17% 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. 4. Sarcos Anthropomorphic Arm

V. CONCLUSION

We derived a Bayesian version of rigid body dynamics parameter estimation based on Joint Factor Analysis, a classical machine learning technique. The Bayesian parameter estimation algorithm is robust to high dimensional ill-conditioned data contaminated with noisy input and noisy output data. In order to produce physically coherent rigid body parameters of the robotic system, we additionally introduce a post-processing step that takes the Bayesian estimate and projects the solution onto a set of constraints, derived from the parallel axis theorem and from the need to ensure positive values for certain parameters. We demonstrate the efficiency of the algorithm by applying it on a synthetic dataset, a 7 DOF robotic vision head and a 10 DOF robotic anthropomorphic arm. Our algorithm successfully identified the system parameters from 5 to 20% higher accuracy than alternative methods, thus proving to be a competitive alternative for parameter estimation on complex high degree-of-freedom robotic systems.

ACKNOWLEDGMENTS

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.

REFERENCES

[1] L. Sciavicco and B. Siciliano. *Modeling and control of robot manipulators*. MacGraw-Hill, 1996.

[2] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal. Comparative experiments on task space control with redundancy resolution. In *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1575–1582. IEEE, 2005.

[3] C. H. An, C. G. Atkeson, and J. M. Hollerbach. *Model-based control of a robot manipulator*. MIT Press, 1988.

[4] G. H. Golub and C. Van Loan. *Matrix Computations*. John Hopkins University Press, 198.

[5] S. Van Huffel and J. Vanderwalle. The total least squares problem: Computational aspects and analysis. *Society for Industrial and Applied Mathematics*, 1991.

[6] J. M. Hollerbach and C. W. Wampler. The calibration index and the role of input noise in robot calibration. In G. Giralt and G. Hirzinger, editors, *Robotics Research: The Seventh International Symposium*, pages 558–568. Springer, 1996.

[7] Y. N. Rao and J.C. Principe. Efficient total least squares method for system modeling using minor component analysis. In *Proceedings of International Workshop on Neural Networks for Signal Processing*, pages 259–268. IEEE, 2002.

[8] S. C. Douglas. Analysis of an anti-hebbian adaptive FIR filtering algorithm. *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, 43(11), 1996.

[9] D. A. Belsley, E. Kuh, and R. E. Welsch. *Regression diagnostics: Identifying influential data and sources of collinearity*. Wiley, 1980.

[10] Y. N. Rao, D. Erdogmus, G. Y. Rao, and J.C. Principe. Fast error whitening algorithms for system identification and control. In *Proceedings of International Workshop on Neural Networks for Signal Processing*, pages 309–318. IEEE, 2003.

[11] V. Strassen. Gaussian elimination is not optimal. *Num Mathematik*, 13:354–356, 1969.

[12] T. J. Hastie and R. J. Tibshirani. *Generalized additive models*. Number 43 in Monographs on Statistics and Applied Probability. Chapman and Hall, 1990.

[13] W.F. Massey. Principal component regression in exploratory statistical research. *Journal of the American Statistical Association*, 60:234–246, 1965.

[14] A. D'Souza, S. Vijayakumar, and S. Schaal. The bayesian backfitting relevance vector machine. In *Proceedings of the 21st International Conference on Machine Learning*. ACM Press, 2004.

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

[16] Z. Ghahramani and M.J. Beal. Graphical models and variational methods. In D. Saad and M. Opper, editors, *Advanced Mean Field Methods - Theory and Practice*. MIT Press, 2000.

[17] R.M. Neal. *Bayesian learning for neural networks*. PhD thesis, Dept. of Computer Science, University of Toronto, 1994.

[18] N. R. Draper and H. Smith. *Applied Regression Analysis*. Wiley, 1981.

[19] S. Derksen and H.J. Keselman. 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, 1992.

[20] H. Wold. Soft modeling by latent variables: The nonlinear iterative partial least squares approach. In J. Gani, editor, *Perspectives in probability and statistics, papers in honor of M. S. Bartlett*. Academic Press, 1975.

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

Appendix

A. EM-update Equations

We can then derive the following EM updates using standard manipulations of normal distributions:

M-step :

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

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

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

E-step :

$$\sigma_{w_{zm}}^2 = \frac{1}{\frac{1}{\psi_{zm}}\sum_{i=1}^{N}\left\langle t_{im}^2\right\rangle + \left\langle\alpha_m\right\rangle}, \left\langle w_{zm}\right\rangle = \frac{\sigma_{w_{zm}}^2}{\psi_{zm}}\sum_{i=1}^{N}\left\langle z_{im}t_{im}\right\rangle$$

$$\sigma_{w_{xm}}^2 = \frac{1}{\frac{1}{\psi_{xm}}\sum_{i=1}^{N}\left\langle t_{im}^2\right\rangle + \left\langle\alpha_m\right\rangle}, \left\langle w_{xm}\right\rangle = \frac{\sigma_{w_{xm}}^2}{\psi_{xm}}\sum_{i=1}^{N}x_{im}\left\langle t_{im}\right\rangle$$

$$\hat{a}_{\alpha_m} = a_{\alpha_{m0}} + 1, \hat{b}_{\alpha_m} = b_{\alpha_{m0}} + \frac{\left\langle w_{zm}^2\right\rangle + \left\langle w_{xm}^2\right\rangle}{2}$$

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

$$\Sigma_{zz} = \mathbf{M} - \frac{\mathbf{M}\mathbf{1}\mathbf{1}^T\mathbf{M}}{\psi_y + \mathbf{1}^T\mathbf{M}\mathbf{1}}, \Sigma_{zt} = -\Sigma_{zz}\left\langle\mathbf{W}_z\right\rangle\mathbf{\Psi}_z^{-1}\mathbf{K}^{-1}, \Sigma_{tz} = \Sigma_{zt}^T$$

$$\Sigma_{tt} = \mathbf{K}^{-1} + \mathbf{K}^{-1}\left\langle\mathbf{W}_z\right\rangle^T\mathbf{\Psi}_z^{-1}\Sigma_{zz}\mathbf{\Psi}_z^{-1}\left\langle\mathbf{W}_z\right\rangle\mathbf{K}^{-1}$$

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

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

and where $\left\langle\mathbf{W}_x\right\rangle$ is a diagonal d by d matrix with $\left\langle\mathbf{w_x}\right\rangle$ along its diagonal. Similarly, $\left\langle\mathbf{W}_z\right\rangle$, $\mathbf{\Psi}_x$, $\mathbf{\Psi}_z$ are d by d diagonal matrices with diagonal vectors of $\left\langle\mathbf{w_z}\right\rangle, \psi_{\mathbf{x}}$ and $\psi_{\mathbf{z}}$. The E-step updates for $\mathbf{Z}$ and $\mathbf{T}$ are then:

$$\left\langle z_i\right\rangle = \frac{y_i}{\psi_y}\mathbf{1}^T\Sigma_{zz} + x_i\left\langle\mathbf{W}_x\right\rangle^T\mathbf{\Psi}_x^{-1}\Sigma_{tz}$$

$$\left\langle t_i\right\rangle = \frac{y_i}{\psi_y}\mathbf{1}^T\Sigma_{zz}\left\langle\mathbf{W}_z\right\rangle\mathbf{\Psi}_z^{-1}\mathbf{K}^{-1} + x_i\left\langle\mathbf{W}_x\right\rangle^T\mathbf{\Psi}_x^{-1}\Sigma_{tt}$$

$$\sigma_z^2 = \mathrm{diag}(\Sigma_{zz}), \sigma_t^2 = \mathrm{diag}(\Sigma_{tt}), \mathrm{cov}(z,t) = \mathrm{diag}(\Sigma_{zt})$$

B. Inference of Regression Estimate for Joint Factor Analysis regression

The regression esimate for Joint-Space Factor Analysis regression is as follows:

$$b_{JFA} = \mathbf{W}_z\left(\mathbf{I} + \mathbf{W}_x^T\mathbf{W}_x\Psi_x^{-1}\mathbf{W}_x\right)^{-1}\mathbf{W}_x^T\Psi_x^{-1} \qquad (9)$$

C. Minimizing Least Squared Error

We want to minimize the cost function J where $J = \frac{1}{2}\left(\mathbf{y} - \mathbf{X}\theta\right)^T\left(y - \mathbf{X}\theta\right)$. Now, let us differentiate J with respect to θ to get:

$$\frac{\partial J}{\partial\theta} = -\left(\mathbf{y} - \mathbf{X}\theta\right)^T\mathbf{X} \qquad (10)$$

By setting $\frac{\partial J}{\partial\theta}$ to 0 and solving for θ, we will get the solution to the minimization cost problem. We can see that when we have the optimal solution for θ that minimizes J, then $(\mathbf{y} - \mathbf{X}\theta)^T\mathbf{X} = 0$. When we are dealing with noisy instead of noiseless inputs, then we can resort to our EM-based de-noising algorithm that attempts to find a θ that is optimal.