SIMULTANEOUS TRACKING AND SPARSE CALIBRATION IN GROUND SENSOR NETWORKS USING EVIDENCE APPROXIMATION

Marek Syldatk, Fredrik Gustafsson

Department of Electrical Engineering, Linköping University
Linköping, Sweden
{marek.syldatk, fredrik.gustafsson}@liu.se

ABSTRACT

Calibration of ground sensor networks is a complex task in practice. To tackle the problem, we propose an approach based on simultaneous tracking of targets of opportunity and sparse estimation of the bias parameters. The evidence approximation method is used to get a sparse estimate of the bias parameters, and the method is here extended with a novel marginalization step where a state smoother is invoked. A simulation study shows that the non-zero bias parameters are detected and well estimated using only one target of opportunity passing by the network.

Index Terms— Evidence Approximation, Parameter Estimation, Bayesian Inference, Sensor Networks, Sparsity

1. INTRODUCTION

Calibration of the Ground Sensor Network (GSN), also known as sensor registration or bias estimation, is a crucial element for performance of the entire system. Improper alignment of the sensors might decrease the performance of the network and in fact result in degrading the quality of tracking, appearance of ghost tracks and problems in measurement to track association.

There are many previous works dealing with bias estimation [1][2][3][4]. Here we focus on methods applicable to GSN. The different biases, here denoted bias parameters, may include location errors, orientation errors, range measurements etc., depending on sensor type. There are two main groups of techniques for calibrating the sensors: using reference targets, or using targets of opportunity. Both can be performed on-line or off-line.

In this application, the reference targets could typically be one or more vehicles carrying satellite navigation equipment, e.g. GPS. One must make sure that enough measurements are generated for each sensor with the target at different locations. Once the data has been collected, one can apply e.g. a maximum likelihood (ML) method [3], to estimate the bias parameters that make observations of the target fit the reference data best.

In case no reference targets can be used, one needs to track targets of opportunity for the purpose of bias estimation. One approach is to simultaneously estimate the track states and biases, which can be accomplished by forming augmented state vectors that combine target state estimates together with bias estimates [1]. With a large number of targets and sensors this will hardly be tractable due to computational requirements etc. There are however suboptimal but very efficient techniques to decouple target state and bias estimation process, fully feasible for on-line operation [5][2][4].

As the number of measurements suitable for bias estimation can be quite low, it is desirable to find a method that uses them as effectively as possible. One such off-line method is the Expectation Maximization (EM) algorithm [6][7][8], where the entire set of measurements is processed iteratively to provide both state estimates and bias parameters.

This paper examines a slightly different approach to the off-line bias estimation with targets of opportunity. Because the number of measurements is usually low, it is useful to apply some regularization to the maximum likelihood estimation, in order to avoid the problems with observability. In practice, only a few sensors need calibration usually, and most of the bias parameters are zero, so one should use this extra information.

In this paper biases are considered to be stochastic variables and a method called *Type-II Bayes* [9], *evidence approximation* [10] or *sparse Bayesian learning* [11] is applied, where each bias parameter has its own regularization parameter, corresponding to the priors in a Bayesian framework. Those parameters are estimated together with bias parameters. The approach utilizes *Occam's razor* [10][12], which lets us find a good balance between model complexity and fit to data. It also implies sparsity through regularization and, in addition, provides us a very useful information about how well each of the parameters is determined by the data.

Application of different regularizers for each parameter is a basic idea underlying the Relevance Vector Machines algorithm, provided by Tipping in [13], using the same sparse Bayesian learning framework by Mackay [10].

Our approach to simultaneous tracking and calibration requires a novel extension to the evidence approximation method, where the target trajectories are marginalized using state smoothers for computing the evidence function.

The method will be described in detail in Section 3 after the formal definition in Section 2. Section 4 provides simulation results. In Section 5 final conclusions are stated and future work is suggested.

2. PROBLEM FORMULATION

A ground target is detected by a number of sensors, resulting in a set of observations $Y_N = \{y_n\}_{n=1}^{N}$ of the target states $X_N = \{x_n\}_{n=1}^{N}$, where y_n is a single measurement, or set of measurements stacked as a vector, and x_n is target state at time $n = 1 \ldots N$. The measurements are affected by a number of bias parameters and by a measurement noise. All the bias parameters, for all sensors, are collected in a single vector $\boldsymbol{\theta}$.

The measurement model for each sensor is assumed to be a known, nonlinear function $h(x_n, \boldsymbol{\theta})$ of the state x_n, with measure-

The research leading to these results has received funding from the EU's Seventh Framework Program under grant agreement no 238710. The research has been carried out in the MC IMPULSE project: https://mcimpulse.isy.liu.se/.

ment bias vector $\boldsymbol{\theta} = [\theta_1 \dots \theta_m \dots \theta_M]^T$, and is defined as

$$y_n = h(x_n, \boldsymbol{\theta}) + v_n \tag{1}$$

where v_n is an additive noise with Normal distribution and known covariance matrix R.

As was partially mentioned in the introduction, there are three main problems with calibration of the sensor network. Low number of measurements causes problems with observability, and thus the maximum likelihood methods do not usually provide reliable results. The problem can be solved by applying some constraints on the estimates, which in the Bayesian framework are solved by assigning the priors. Choosing a correct prior is also a problem by itself, since it should utilize the sparsity feature of the bias vector. Finally, a correct calibration cannot be performed, when the state of the ground target is unknown, which is usually the case.

2.1. General estimation framework

In the estimation approach, the bias vector $\boldsymbol{\theta}$ is assumed to be a stochastic variable. The method consists of finding a posterior distribution of the bias parameters, by using the Bayesian maximum a posteriori method

$$p(\boldsymbol{\theta}|Y_N, \boldsymbol{\alpha}) = \frac{p(Y_N|\boldsymbol{\theta}, \boldsymbol{\alpha})p(\boldsymbol{\theta}|\boldsymbol{\alpha})}{p(Y_N|\boldsymbol{\alpha})} \tag{2}$$

The set of *hyper-parameters* $\boldsymbol{\alpha}$, determines the prior distribution for $\boldsymbol{\theta}$, given by $p(\boldsymbol{\theta}|\boldsymbol{\alpha})$. Since those parameters are in general unknown, a method using maximization of an *evidence* function (also called a *marginal likelihood*) is applied. Evidence, which is a normalizing term in (2), is obtained by integrating out the parameter vector, as

$$p(Y_N|\boldsymbol{\alpha}) = \int p(Y_N|\boldsymbol{\theta}, \boldsymbol{\alpha})p(\boldsymbol{\theta}|\boldsymbol{\alpha})d\boldsymbol{\theta} \tag{3}$$

A general framework for the procedure of maximizing the evidence, called *evidence procedure*, is provided for example in [10][13][14].

In this paper a general case is handled, when the likelihood $p(Y_N|\boldsymbol{\theta}, \boldsymbol{\alpha})$ depends also on the state of the target X_N, that is generally unknown. The set of those target states is then integrated out from $p(Y_N|\boldsymbol{\theta}, \boldsymbol{\alpha}, X_N)$ by making use of an estimated state distribution $\hat{p}(X_N)$, obtained from a filtering/smoothing algorithm [15][16][17]. Thus the likelihood, independent of X_N, is computed using

$$p(Y_N|\boldsymbol{\theta}, \boldsymbol{\alpha}) = \int p(Y_N|X_N, \boldsymbol{\theta}, \boldsymbol{\alpha})\hat{p}(X_N)dX_N \tag{4}$$

3. SPARSE BAYESIAN CALIBRATION ALGORITHM

Having previously defined the measurements in (1) and using the assumption that measurements are independent, the likelihood for full data set can be easily written as

$$p(Y_N|X_N, \boldsymbol{\theta}, \boldsymbol{\alpha}) = \mathcal{N}(\mathbf{Y}; \mathbf{h}(\mathbf{X}, \boldsymbol{\theta}), \mathbf{R}) \tag{5}$$

where $\mathbf{Y} = [y_1^T \dots y_N^T]^T$, $\mathbf{X} = [x_1^T \dots x_N^T]^T$, $\mathbf{h}(\mathbf{X}, \boldsymbol{\theta}) = [h(x_1, \boldsymbol{\theta})^T \dots h(x_n, \boldsymbol{\theta})^T]^T$ and $\mathbf{R}$ is a covariance matrix with R matrices on diagonal.

In case of the sensor networks, the true value of the state x_n is usually unknown, so it is reasonable to use estimates obtained from for instance the Kalman or particle filter/smoother, using some initial value of $\boldsymbol{\theta}^0$ and $\boldsymbol{\alpha}^0$. For the Gaussian case, a set of state estimates $\hat{X}_N = \{\hat{x}_n\}_{n=1}^N$, together with its corresponding covariance matrices $\hat{P}_N = \{\hat{P}_n\}_{n=1}^N$, results in a distribution

$$P(X_N|Y_N, \boldsymbol{\theta}^0, \boldsymbol{\alpha}^0) = \mathcal{N}(\mathbf{X}|\hat{\mathbf{X}}, \hat{\mathbf{P}}) \tag{6}$$

where $\hat{\mathbf{X}} = [\hat{x}_1^T \dots \hat{x}_N^T]^T$ and $\hat{\mathbf{P}}$ is block diagonal with $\hat{P}_1^T \dots \hat{P}_N^T$ on the diagonal. By treating (6) as a distribution over the state, $\hat{p}(X_N) = P(X_N|Y_N, \boldsymbol{\theta}^0, \boldsymbol{\alpha}^0)$, one can remove the dependence on X_N from (5), by applying the marginalization defined in (4)

By linearizing the measurement function around some $\hat{\boldsymbol{\theta}}$ and $\hat{\mathbf{X}}$, Equation (1) can be rewritten, using a 1st order Taylor expansion, as

$$\bar{y}_n = H_{\boldsymbol{\theta}}(\hat{x}_n, \hat{\boldsymbol{\theta}})\boldsymbol{\theta} + H_x(\hat{x}_n, \hat{\boldsymbol{\theta}})(x - \hat{x}_n) + v_n \tag{7}$$

where $\bar{y}_n = y_n - h(\hat{x}_n, \hat{\boldsymbol{\theta}}) + H_{\hat{\boldsymbol{\theta}}}(\hat{x}_n, \hat{\boldsymbol{\theta}})\hat{\boldsymbol{\theta}}$ and

$$H_{\boldsymbol{\theta}}(\hat{x}_n, \hat{\boldsymbol{\theta}}) = \left.\frac{\partial h(x_n, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\theta\, =\, \hat{\boldsymbol{\theta}},\, x\, =\, \hat{x}_n} \tag{8}$$

$$H_x(\hat{x}_n, \hat{\boldsymbol{\theta}}) = \left.\frac{\partial h(x_n, \boldsymbol{\theta})}{\partial x_n}\right|_{\theta\, =\, \hat{\boldsymbol{\theta}},\, x\, =\, \hat{x}_n} \tag{9}$$

Thus the result of integration, where some terms got cancelled in the meantime, is given by

$$p(Y_N|\boldsymbol{\theta}, \boldsymbol{\alpha}) \approx \mathcal{N}(\bar{\mathbf{Y}}|\mathbf{H}_{\boldsymbol{\theta}}\boldsymbol{\theta}, \bar{\mathbf{R}}) \tag{10}$$

where the covariance matrix $\bar{\mathbf{R}}$ is defined as $\bar{\mathbf{R}} = (\mathbf{R} + \mathbf{H_x}\hat{\mathbf{P}}\mathbf{H_x^T})$, and $\bar{\mathbf{Y}} = [\bar{y}_1 \dots \bar{y}_N]^T$, $\mathbf{H}_{\boldsymbol{\theta}} = [H_{\boldsymbol{\theta}}(\hat{x}_1, \hat{\boldsymbol{\theta}})^T \dots H_{\boldsymbol{\theta}}(\hat{x}_N, \hat{\boldsymbol{\theta}})^T]^T$ and $\mathbf{H_X}$ is defined analogously.

In a Bayesian framework, to infer the values of $\boldsymbol{\theta}$, we need to apply some constraint on the data, which is accomplished by assigning a prior. Here we use a zero mean Gaussian prior

$$p(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \prod_{m=1}^{M} \mathcal{N}(\theta_m|0, \alpha_m) = \mathcal{N}(\boldsymbol{\theta}|\mathbf{0}, \mathbf{A}^{-1}) \tag{11}$$

where $\boldsymbol{\alpha} = [\alpha_1 \dots \alpha_M]^T$ is the hyper-parameter defining the precision, and $\mathbf{A} = diag(\alpha_1, \dots, \alpha_M)$. The problem is how to assign the hyper-parameters? In this framework, the hyper-parameters are assumed unknown and inferred from the data together with the set of parameters $\boldsymbol{\theta}$. Therefore, by having defined the likelihood and prior above, one only needs to define the prior for hyper-parameters. Here we will consider flat prior that is non informative [18], $p(\boldsymbol{\alpha}) = const$, giving equal probability for all possible values of $\boldsymbol{\alpha}$. Having defined the prior, we can now proceed to the Bayesian inference of unknown variables.

3.1. Parameter estimation

In this step the values of $\boldsymbol{\theta}$ will be inferred from the data. In a fully Bayesian framework, the posterior over the parameter set $\boldsymbol{\theta}$ should be obtained by integrating out the hyper-parameters, so then

$$p(\boldsymbol{\theta}|Y_N) = \int p(\boldsymbol{\theta}|Y_N, \boldsymbol{\alpha})p(\boldsymbol{\alpha}|Y_N)d\boldsymbol{\alpha} \tag{12}$$

By assuming the posterior for $\boldsymbol{\alpha}$ is sharply peaked around its estimate $\hat{\boldsymbol{\alpha}}$, so $p(\boldsymbol{\alpha}) \approx \delta(\boldsymbol{\alpha} - \hat{\boldsymbol{\alpha}})$, we can [10] use the approximation

$$p(\boldsymbol{\theta}|Y_N) \simeq p(\boldsymbol{\theta}|Y_N, \boldsymbol{\alpha})|_{\boldsymbol{\alpha} = \hat{\boldsymbol{\alpha}}} \tag{13}$$

Hereby the posterior above can be rewritten, using Bayes rule, as

$$p(\boldsymbol{\theta}|Y_N) \simeq \left.\frac{p(Y_N|\boldsymbol{\theta}, \boldsymbol{\alpha})p(\boldsymbol{\theta}|\boldsymbol{\alpha})}{p(Y_N|\boldsymbol{\alpha})}\right|_{\boldsymbol{\alpha}\, =\, \hat{\boldsymbol{\alpha}}} \tag{14}$$

The posterior for $\boldsymbol{\theta}$ is obtained using (10) and (11), together with a simple rule for the posterior distribution in a Gaussian case, as

$$p(\boldsymbol{\theta}|Y_N) = \mathcal{N}(\boldsymbol{\theta}|\mathbf{m}, \boldsymbol{\Sigma}) \tag{15}$$

which is also Gaussian, with mean and covariance defined by

$$\mathbf{m} = \mathbf{\Sigma}(\mathbf{H}_{\boldsymbol{\theta}}^{\mathbf{T}}\bar{\mathbf{R}}^{-\mathbf{1}}\bar{\mathbf{Y}}) \tag{16}$$

$$\mathbf{\Sigma} = (\mathbf{H}_{\boldsymbol{\theta}}^{\mathbf{T}}\bar{\mathbf{R}}^{-\mathbf{1}}\mathbf{H}_{\boldsymbol{\theta}} + \mathbf{A})^{-\mathbf{1}} \tag{17}$$

It is sometimes useful to obtain a point estimate of the bias vector, which maximizes the posterior

$$\hat{\boldsymbol{\theta}}_{map} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|Y_N) \tag{18}$$

which is equal to the mean defined in (16).

3.2. Hyper-parameter estimation

To infer the hyper-parameters, Bayes rule is applied again, so

$$p(\boldsymbol{\alpha}|Y_N) = \frac{p(Y_N|\boldsymbol{\alpha})p(\boldsymbol{\alpha})}{p(Y_N)} \tag{19}$$

As was declared before, the prior $p(\boldsymbol{\alpha})$ is assumed to be flat, so to obtain optimal values of hyper-parameters, we only need to consider the first term in a nominator, $p(Y_N|\boldsymbol{\alpha})$. As one can easily notice, it is the normalizing constant of Equation (14), defined previously in (3) as an *evidence*. To obtain the evidence, the parameter $\boldsymbol{\theta}$ needs to be integrated out as in (3), so using (10) and (11), one can write

$$p(Y_N|\boldsymbol{\alpha}) = \int \mathcal{N}(\bar{\mathbf{Y}}|\mathbf{H}_{\boldsymbol{\theta}}\boldsymbol{\theta}, \bar{\mathbf{R}})\mathcal{N}(\boldsymbol{\theta}|\mathbf{0}, \mathbf{A}^{-\mathbf{1}})d\boldsymbol{\theta} \tag{20}$$

and the optimal value is then the one that maximizes the likelihood

$$\hat{\boldsymbol{\alpha}} = \arg\max_{\boldsymbol{\alpha}} p(Y_N|\boldsymbol{\alpha}) \tag{21}$$

Because (20) is a simple convolution of Gaussians, the evidence can be easily computed as

$$p(Y_N|\boldsymbol{\alpha}) = \frac{|\bar{\mathbf{R}}|^{-\frac{1}{2}}}{(2\pi)^{\frac{Nd_y}{2}}}|\mathbf{A}|^{\frac{1}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}e^{-Q(\bar{\mathbf{Y}})} \tag{22}$$

where d_y is a dimension of $\bar{\mathbf{Y}}$, and the quadratic term $Q(\bar{\mathbf{Y}})$ is given by

$$Q(\bar{\mathbf{Y}}) = \frac{1}{2}(\bar{\mathbf{Y}}^{\mathbf{T}}\bar{\mathbf{R}}^{-\mathbf{1}}\bar{\mathbf{Y}} - \mathbf{m}^{\mathbf{T}}\mathbf{\Sigma}^{-\mathbf{1}}\mathbf{m}) \tag{23}$$

Now, to obtain the estimate of $\boldsymbol{\alpha}$, the approach presented in [10] will be used. By taking the derivative of a logarithm of the evidence (22) with respect to α_m, and equalling it to zero, we get

$$-\frac{1}{2}\mathbf{m}_m^2 - \frac{1}{2}\mathbf{\Sigma}_{mm} + \frac{1}{2\alpha_m} = 0 \tag{24}$$

where $\mathbf{m}_m$ is the m-th element of the $\mathbf{m}$ vector, defined in (16), and $\mathbf{\Sigma}_{mm}$ is the m-th diagonal element of the covariance matrix defined in (17). The equation yields a following solution

$$\alpha_m = \frac{\gamma_m}{\mathbf{m}_m^2} \tag{25}$$

where $\gamma_m = 1 - \alpha_m\mathbf{\Sigma}_{mm}$. According to [10][12], the new parameter can take values in a range of $\gamma_m \in [0 \ldots 1]$, and it determines how well the corresponding parameter θ_m is determined by the data. For small standard deviation of the prior, when α_m is large, the estimates are strongly constrained by the prior. In that case $\mathbf{\Sigma}_{mm}$ in (17) is dominated by hyper-parameter, so $\mathbf{\Sigma}_{mm} \approx \alpha_m^{-1}$, and thus $\gamma_{mm} \approx 0$. On the other hand, when α_m takes a small value, which means the corresponding estimate $\mathbf{m}_m$ is well fit to the data, then $\gamma_m \approx 1$.

As we can see the algorithm requires an iterative approach. We need to start with some initial estimates of $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$ and iteratively refine the estimates. The procedure is shown in Algorithm 1

Algorithm 1 Sparse calibration algorithm

```
1. Initiate at $i = 0$ with $\boldsymbol{\theta}^0$ and $\boldsymbol{\alpha}^0$.
2. Use a filtering/smoothing algorithm to obtain state estimates
   $\hat{X}_N$ and corresponding covariances $\hat{P}_N$, using $\boldsymbol{\theta}^0$ as a bias
   estimate.
3. For $i = 1$:
    (a) Linearize the measurement function as in (7) using $\hat{X}_N$
        and $\hat{\boldsymbol{\theta}}^{i-1}$.
    (b) Set $\boldsymbol{\alpha} = \hat{\boldsymbol{\alpha}}^{i-1}$ and using Equation (16), (17) and (18)
        obtain new estimate of bias vector $\hat{\boldsymbol{\theta}}^i = \hat{\boldsymbol{\theta}}_{map} = \mathbf{m}$.
    (c) Set $\gamma_m = 1 - \hat{\alpha}_m^{i-1}\mathbf{\Sigma}_{mm}$ and obtain new estimate of
        $\boldsymbol{\alpha}$ using (25), given by $\hat{\alpha}_m^i = \gamma_m/\mathbf{m}_m^2$
    (d) If converged: (4); Otherwise: $i = i + 1$ and repeat (3).
4. End iterations.
```

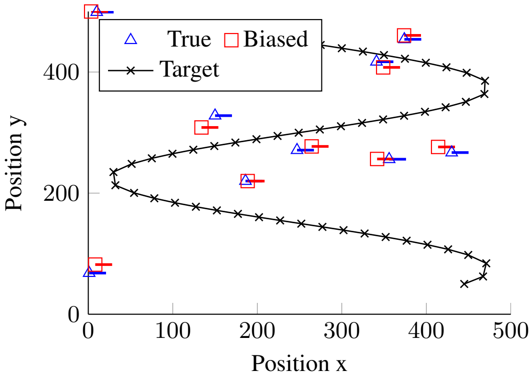

Fig. 1. Scenario overview with target ground truth trajectory together with true and biased sensor positions.

4. EXPERIMENTAL RESULTS

In the experimental scenario, $N_S = 10$ sensors are distributed randomly on an area of 500×500 meters. Sensors are gathering range measurements (in meters) and bearing measurements (in radians). Both positions and measurements are biased, with additive biases. Positioning biases related to the x- and y-axis are defined as θ_x and θ_y respectively; measurement biases related to range- and bearing-bias are defined as θ_r and θ_a respectively. Simulations are performed in a Monte Carlo (MC) manner, with biases randomly generated in each MC run, according to zero-mean normal distribution with standard deviations σ_x, σ_y, σ_r and σ_a respectively. Bias vector is also assumed to be sparse, with sparsity s_f, representing a fraction of non-zero elements, ranging from 0 to 1. Measurements, as in (1), are collected using a following vector measurement function, defined for each sensor as

$$h(x_n) = \begin{bmatrix} h_1(x_n) \\ h_2(x_n) \end{bmatrix} = \begin{bmatrix} \sqrt{d_{x,n}^2 + d_{y,n}^2} + \theta_r \\ \arctan\frac{d_{y,n}}{d_{x,n}} + \theta_a \end{bmatrix} \tag{26}$$

$$d_{x,n} = p_{x,n}^x - (p_x^s - \theta_x), \quad d_{y,n} = p_{y,n}^x - (p_y^s - \theta_y) \tag{27}$$

and where state vector $x_n = [p_{x,n}^x, p_{y,n}^x, v_{x,n}^x, v_{y,n}^x]^T$, with elements corresponding to x- and y-position and x- and y-velocities of the target at time step n respectively and $p_x^s$and p_y^s are positions of the sensor in x- and y-axis.

An overview of the simulated Scenario, with $N = 50$ of measurements, is presented on Figure 1. Parameters of the Scenario are summarized in Table 4. For parameter estimation, smoothed state estimates $\hat{X}_N$ are obtained from Kalman smoother using a standard Constant Velocity model.

Table 1. Parameters of the Scenario.

Parameter	Symbol	Value
Number of measurements per sensor	N	50
Number of sensors	N_S	10
Range measurement noise std	σ_y^r	10 m
Angle measurement noise std	σ_y^r	10°
Number of MC runs	MCn	250
Sparsity range	s_p	0 : 0.05 : 1
Number of iterations	I	25
EKF initial position std	σ_p	10 m
EKF initial velocity std	σ_v	5 m/s
Bias: x/y position std	σ_x / σ_y	10 m / 10 m
Bias: range/angle std	σ_r / σ_a	10 m / 10°

Figure 2 presents the summary RMS Error (mean + 1 standard deviation) of bias estimation results, where maximum likelihood (ML), maximum a posteriori (MAP) and evidence approximation (EA) methods are compared for different bias vector sparsities. In the case of MAP, the true prior, from which biases were generated, was used. Figure 3 presents detailed results for one Monte Carlo simulation.

4.1. Summary

As we can see in Figure 2, the proposed method provides much better results compared to traditional maximum likelihood, and slightly, but consistently, performs worse than the MAP method. In the simulation, MAP is assumed to use the true prior distribution from which the biases were generated, including the knowledge about the sparsity shape of the bias vector. It is important to notice, that the more the bias vector contains zero element ($s_p \to 0$), the better the EA and MAP methods perform compared to ML. It is due to the fact, that both methods utilize the feature of sparsity of the bias vector, which is implied by using the correct prior (MAP) or through Occam's razor (EA). On the other hand, ML does not utilize this feature, and since there are no priors applied, the method tends to provide unreliable results in case, when the estimated parameters are not well determined by the measurement data.

Figure 3 presents one Monte Carlo run estimation results for sensors 5, 6 and 7, with corresponding bias parameters θ_{17} to θ_{28} and related γ_m values obtained from the EA algorithm. In that case only parameters number 18, 21, 26 and 27 had non-zero values. As we can see, for most θ_m all 3 algorithms managed to provide quite similar results. We shall now look at the difference that appeared between them.

One can observe that EA algorithm correctly detected most of the zero and non-zero elements. The parameters, that were well determined by the measurements, are indicated by the value of a corresponding gamma parameter close to unity. One can observe this situation for θ_{21}, where all methods give very good results, and the parameter is well determined by the data, so $\gamma_{21} \simeq 1$. On the other hand, for elements θ_{25} and θ_{27}, the ML has significantly diverged, most probably due to the problem with observability. In that case EA algorithm performed more robust, and has shrunk the values towards zero, what is indicated by value of $\gamma_{25,27} \simeq 0$. The parameters that were estimated with values close to zero, unless they were well determined by the data, were shrunk towards the prior mean value (zero), and therefore the sparsity feature was implied. Situation like this can be observed for example for parameters $\theta_{22,23,24}$.

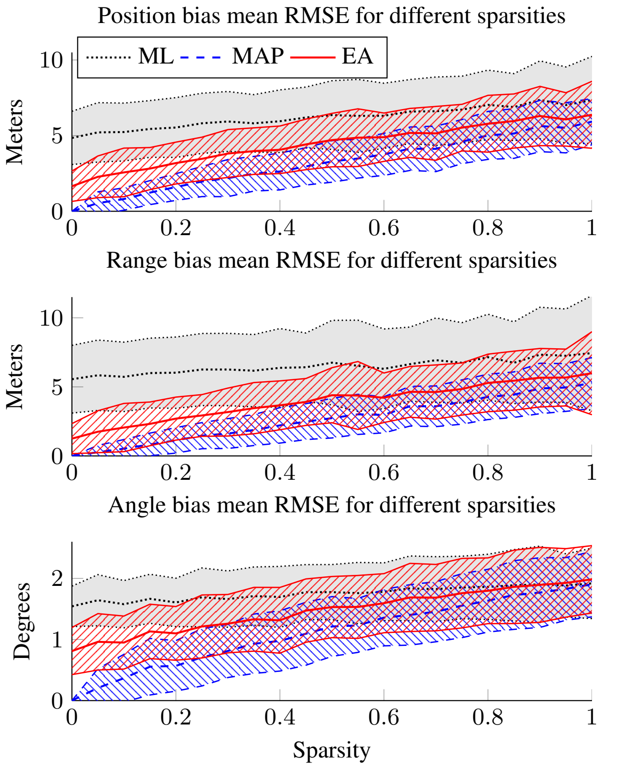

Fig. 2. Mean and standard deviation of the bias estimation RMS Error for different sparsities.

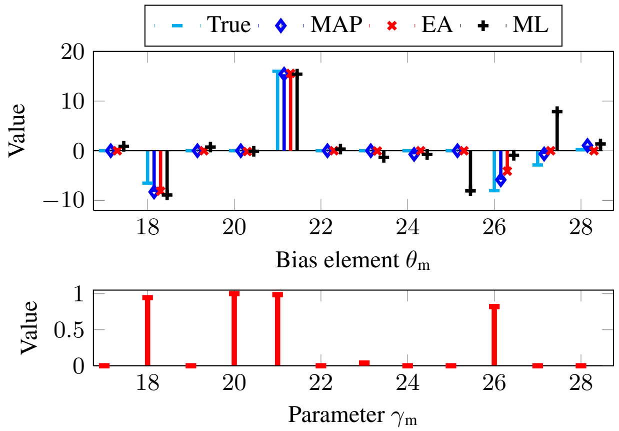

Fig. 3. Comparison of estimation methods and γ parameter .

5. CONCLUSIONS

The proposed evidence approximation (EA) method estimates the bias parameters in a sensor calibration problem, jointly with the state trajectories of targets of opportunity. As a bonus, EA estimates the prior of the bias parameters, which can in itself serve as an indicator of how well the bias parameters are estimated. This is a useful and practical advantage compared to the more direct maximum likelihood (ML) approach. Further, the EA method provides a sparse bias vector in contrast to the ML method, which is useful in practical ground sensor network deployments, where many sensors can be assumed bias-free.

6. REFERENCES

[1] S. Dhar, “Application of a recursive method for registration error correction in tracking with multiple sensors,” in *American Control Conference, 1993*, june 1993, pp. 869 –874.

[2] E. Sviestins, “Online bias estimation for multisensor tracking,” in *Information, Decision and Control, 1999. IDC 99. Proceedings. 1999*, 1999, pp. 221 –226.

[3] N. Okello and B. Ristic, “Maximum likelihood registration for multiple dissimilar sensors,” *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 39, no. 3, pp. 1074 – 1083, july 2003.

[4] J. Vermaak, S. Maskell, and M. Briers, “Online sensor registration,” in *Aerospace Conference, 2005 IEEE*, march 2005, pp. 2117 –2125.

[5] M. Ignagni, “An alternate derivation and extension of friendland’s two-stage kalman estimator,” *Automatic Control, IEEE Transactions on*, vol. 26, no. 3, pp. 746 – 750, jun 1981.

[6] Zhenhua Li, H. Leung, and Liang Tao, “Simultaneous registration and fusion of radar and esm by em-eks,” in *Intelligent Control and Automation (WCICA), 2010 8th World Congress on*, july 2010, pp. 1130 –1134.

[7] T.B. Schön, A. Wills, and B. Ninness, “System identification of nonlinear state-space models,” *Automatica*, vol. 47, no. 1, pp. 39–49, 2011.

[8] M. Syldatk, E. Sviestins, and F. Gustafsson, “Expectation maximization algorithm for calibration of ground sensor networks using a road constrained particle filter,” in *Information Fusion (FUSION), 2012 15th International Conference on*. IEEE, 2012, pp. 771–778.

[9] J.O. Berger, *Statistical decision theory and Bayesian analysis*, Springer, 1985.

[10] D.J.C. MacKay, “Bayesian interpolation,” *Neural computation*, vol. 4, no. 3, pp. 415–447, 1992.

[11] D.P. Wipf and B.D. Rao, “Sparse bayesian learning for basis selection,” *Signal Processing, IEEE Transactions on*, vol. 52, no. 8, pp. 2153–2164, 2004.

[12] C.M. Bishop, *Pattern recognition and machine learning (information science and statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, 2006.

[13] M.E. Tipping, “Sparse bayesian learning and the relevance vector machine,” *The Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

[14] SF Gull and J. Skilling, “Quantified maximum entropy memsys5 users manual,” *Maximum Entropy Data Consultants Ltd., Suffolk*, 1999.

[15] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman filter: Particle filters for tracking applications*, Artech House Publishers, 2004.

[16] R.P.S. Mahler, *Statistical multisource-multitarget information fusion*, vol. 685, Artech House ˆ eBoston Boston, 2007.

[17] Fredrik Gustafsson, *Statistical Sensor Fusion*, Studentlitteratur, 1 edition, 2010.

[18] H. Jeffreys and H. Jeffreys, “An invariant form for the prior probability in estimation problems,” *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 186, no. 1007, pp. 453–461, 1946.