



Learning to combine multi-sensor information for context dependent state estimation

Alexandre Ravet, Simon Lacroix, Gautier Hattenberger

► To cite this version:

Alexandre Ravet, Simon Lacroix, Gautier Hattenberger. Learning to combine multi-sensor information for context dependent state estimation. CAP 2013, Conférence Francophone sur l'Apprentissage Automatique, Jul 2013, Lille, France. hal-00936112

HAL Id: hal-00936112

<https://enac.hal.science/hal-00936112>

Submitted on 3 Feb 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Learning to combine multi-sensor information for context dependent state estimation

Alexandre Ravet¹, Simon Lacroix², and Gautier Hattenberger³

^{1 2}LAAS - CNRS

³Ecole Nationale de l'Aviation Civile (ENAC)

May 31, 2013

Abstract

Multi-sensor information fusion for state estimation is a well studied problem in robotics, with many applications and well known benefits. While classical methods for information fusion, based on information theoretic frameworks, provide good performance in optimal operation context, they show a deficiency in their ability to evaluate and take into account sensor measurements validity. As a consequence, these approaches may use erroneous information from a sensor and ruin the benefits of sensor redundancy.

This work begins to address this problem by learning context-dependant knowledge about sensor reliability. This knowledge is later used as a decision rule in the fusion task in order to dynamically select the most appropriate subset of sensors. For this purpose we consider the use of the Mixture of Experts framework (ME), traditionally applied to regression and classification problems. In our context each expert is a Kalman filter fed by a subset of sensors, and a gating network serves as a mediator between individual filters, basing its decision on sensor inputs and additional information needed for reasoning about the operation context. The performance of this model is evaluated in the context of UAV take-off/landing task for altitude estimation.

Keywords: Sensor selection, Mixture of Experts Framework, Learning, UAV.

1 Introduction

State estimation is one of the main challenges in robotics. For many systems, it relies on multiple sensors, each one exhibiting an inherent observation uncertainty, operating range, and context dependent per-

formance. Uncertainty due to the observation noise received considerable attention over past decades and is commonly handled using Bayesian filtering [TBF05]. It is also well known that the use of redundant sensors significantly improves estimation accuracy and reliability. However, such methods do not provide any satisfying way to assess the validity of sensor measurements.

In the context of multi-sensor state estimation, most attempts to deal with this issue lead to self-contained systems, relying on information theoretic framework [SBP⁺07] or rejection schemes designed after experience on the system behaviour [TSL⁺12]. Only little work has been done in view of introducing knowledge about the contextual information in a generic framework for sensor reliability assessment. This work is motivated by the fact that an intelligent system should not only be able to select the sensor -or subset of sensor- based on an online performance measure, but should also encode knowledge about the reliability of a perception modality according to the current specific context.

This implies the ability for the system to discover the implicit operation contexts the robot is likely to encounter, based on the a priori unknown performances of each sensor in these contexts. Attributing belief about the reliability of a sensor in these contexts then require a complex reasoning on information acquired about the environment. Except for simple cases (reduced set of sensors, known environment) we can not easily implement these decision rules by hand. Addressing this problem introduces the need for such a system to be able to learn by itself how to achieve the sensor selection task. For this purpose, we propose to make use of a supervised learning algorithm, in order to learn a mapping from sensors measurements input space (and any other relevant information) to sensors reliability probabilities.

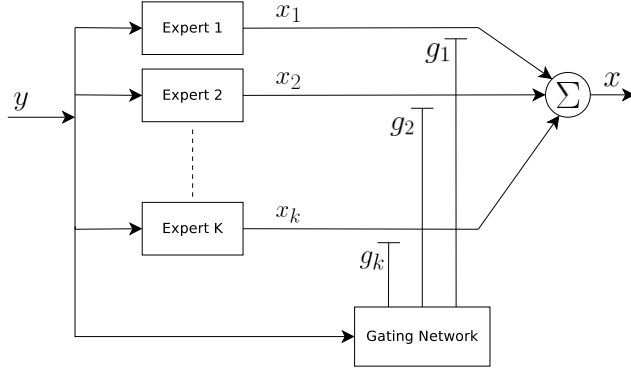


Figure 1: The basic mixture of experts framework

A well-designed robotic platform should exhibit various perception modalities relying on different but complementary physical principles. Consequently the set of perception modalities embedded on a robot does not provide direct commensurate measurements, and it is often easier and more modular to fuse information at a state vector level [HL97]. Furthermore, binding different subsets of sensors to different estimation filters allows us to map the sensor selection problem to the bank of Kalman filter approach. This method assumes that optimal filtering can be expressed by dynamically selecting the most suitable filter among a bank of filters. This approach emerged with the Magill’s filter bank [Mag65], and has been subsequently improved leading to general pseudo Bayes (GPB) methods and interacting multiple models (IMM) [BBS88]. The IMM received a lot of attention as it provides computational efficiency compared to GPB methods. Although some authors decided to augment the IMM with context-dependent information [SW09], this algorithm fundamentally relies on the exploitation of internal estimates and a known transition probability matrix between different filter models. Thus introduction of a knowledge about context dependent model reliability is not straightforward, especially if the user wants the system to learn this information.

Aiming at learning how to combine some complementary experts, the ME framework lends itself very well to the problem as it basically computes an optimal output through a weighted sum of individual experts. To achieve this mediation task, the ME relies on a gating network in charge of providing gating probabilities, equivalent to reliability coefficients over the set of experts (See Fig. 1). An important feature of the standard ME framework is its ability to learn the gating framework parameters as well as the expert models parameters through the use of simple learning methods.

This approach is known to be an efficient alternative to the filter bank approach [CBG98] [CBG97].

Based on the two following contributions :

- Application of the localized gating network to the mixture of kalman filters
- Application of the bank of kalman filter approach for implicit sensor selection

this article aim at showing how the mixture of kalman filter for implicit sensor selection can be applied to the altitude estimation task for a UAV.

The remainder of the paper is organized as follows. Section 2 introduces ME framework and concept of adaptive kalman filtering for sensor selection. Section 3 describes motivations for using the ME model in our test case scenario. It then focuses on gating model and training phase. Section 4 conveys the experimental results obtained for both simulation and real data scenario. Concluding remarks are finally made on section 5.

2 Theoretical Background

2.1 The mixture of experts framework

The mixture of experts approach basically consists in decomposing a complex problem into subtasks, each of which being handled by an appropriate expert. Traditionally used for regression or classification problems, the model learns to split the input space into overlapping regions within which assigned experts are active.

The standard ME framework [JJNH91] consists in a set of K experts modules and a gating network (See Fig. 1). Each expert $k = 1 \dots K$ associated with parameters λ_k looks at input vector y and compute a local output x_k through independent function $f_k(\lambda_k, y)$. In a probabilistic interpretation the output of an expert k can be viewed as the mean of a probability distribution $P(x|y, \lambda_k)$ with x the desired target value associated to sample y . Making the assumption that the different experts may be more competent in different regions of the input space (*i.e.* they have higher probability to produce the desired target x), the gating network mediates the outputs of the bank of experts. For this purpose the gating network produces for each expert k , a probability of its output x_k to be equal to the desired output x . This results in a set of gating probabilities g_k weighting the output of all experts while satisfying constraints $g_k \geq 0, k = 1 \dots K$, and $\sum_{k=1}^K g_k = 1$.

Given an input vector y and a target vector x , the probability of observing x is consequently written in

terms of gating probabilities and experts outputs (using product rule) as

$$\begin{aligned}
P(x|y, \Theta, \Lambda) &= \sum_{k=1}^K P(x, k|y, \Theta, \Lambda) \\
&= \sum_{k=1}^K P(k|y, \Theta) P(x|k, y, \Lambda) \\
&= \sum_{k=1}^K g_k(y, \theta_k) P(x|y, \lambda_k) \quad (1)
\end{aligned}$$

where $\{\Theta, \Lambda\}$ denotes the set of all parameters, with $\Theta = \{\theta_k, k = 1 \dots K\}$ the set of gating parameters and $\Lambda = \{\lambda_k, k = 1 \dots K\}$ the set of experts parameters.

ME implementations then differ in 3 main points: the experts model, the gating model, and the inference method. More information about the different implementations can be found in [YWG12]. Our model for the gating framework will be justified in section 2, and expert models are set as Kalman filter in our case. In this paper we will make use of a common learning method based on the maximum likelihood principle. This method is quickly described hereafter.

Given a training set $\{\mathbf{x}, \mathbf{y}\}$ we try to maximize the likelihood \mathcal{L} of the data set with respect to the model parameters. If samples are considered identically independently distributed, this is equivalent to maximize:

$$\mathcal{L} = \prod_n p(x^n, y^n)$$

We then define the usual cost function \mathcal{C} as the negative log of the likelihood function, such that maximizing likelihood is now equivalent to minimize \mathcal{C} :

$$\mathcal{C} = - \sum_n \ln(p(x^n|y^n))$$

Different methods for determining max likelihood have been developed. The standard gradient descent methods can be applied. More recently, sampling, variational inference and several Expectation Maximization (EM) algorithms have emerged [YWG12] and have shown good performances.

2.2 Adaptive Kalman filtering for sensor selection

We implicitly solve the sensor selection problem through the filter bank approach. In classical implementations, the bank is composed of a finite number of filters differing in transition model, transition noise

and observation noise. A weighting function then assigns weight factors to the output of each individual filter, giving highest weight value to the best performing filter. In our case, models differ only in observation matrices, acting as a selector on the sensors, each of which being associated with the corresponding observation noise matrix (See Fig. 2). By this mean we implicitly select the most appropriate subset of sensors through the filter selection process.

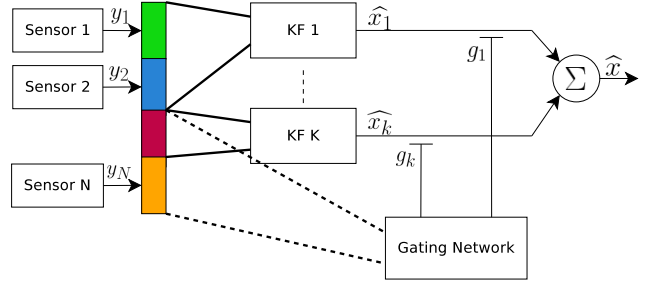


Figure 2: Mixture of experts framework for sensor selection. Each expert (filter) can be wired to one or a subset of sensors. The gating network can either share experts inputs, or use any useful additional information for context assessment.

It is also important to emphasize that, as opposed to classical bank of filters techniques, the ME approach does not assume that the optimal estimator can be obtained by switching between models according to a known transition matrix. From a practical point of view, the ME approach is also computationally faster.

3 ME framework for sensor selection

3.1 Motivations

A micro-UAV is often brought to deal with changing environment. The simple take-off and landing phase of a micro-UAV already reveals many different regimes in term of sensor performance for estimating altitude. Ultrasonic sensors are for example quite reliable and accurate until they reach a given maximum range. They also easily provide outliers measurements due to wrong reflection or when emitted signal is lost. Vision may start to provide information after reaching an unknown altitude, depending on the camera characteristics like the angle of view, and on the ground texture. Barometric pressure sensors provide wide measuring range with quite constant accuracy but also require to estimate a bias due to changing atmospheric conditions,

while GPS provide signal dependant precision, and is more likely to be reliable for high altitudes, also depending on environment characteristics (if navigation occurs in urban environment or open environment for example). These specificities raise the need to create decision rules for selecting an active sensor subset given a specific context.

In [TSL⁺12] the authors report that environment transition results in outliers the classical methods can not reject. Therefore, a mechanism for sensor selection is proposed, giving ability to the system to switch to the sensor that works well in the current environment. This selection mechanism relies on the strong assumption that the sensor with the smallest measurement variance is the more reliable in the current environment. Even if this selection method turns to be efficient enough in the specific context of outdoor/indoor transition with a particular sensor setup, we can understand that this self-contained decision system is not reliable for different types of environment transitions, or for different sensors. In the case of altitude estimation, it would inevitably favour ultra-sonic sensor when maximum range is reached, the sensor providing constant but erroneous altitude value with small variance.

The motivation for this work is to benefit from the ME gating network ability either to share the experts inputs, or to wire additional connections (See Fig. 2). This easily allows us to base decision on context-dependent information of any kind. Under the assumption that our training set contains enough samples, the learned gating network then ensures adaptation to the different environments, providing a partial assessment of sensor performances.

The learning step also provides modularity in terms of sensor configuration as the model automatically learns adapted decision boundaries when sensors are added or removed, hence decreasing the engineering needs of hardware configuration modification.

3.2 Using Localized gating network to encode context-dependent decision rules

Besides adapting the perception modalities, we also aim at switching smoothly between experts. This requirement especially makes sense in flight context, where hard transitions between sensors (consequently between estimates) are not admissible, as it directly impacts the flight stability as well as the robot safety in cluttered environment. In the standard model, the gating network is a single layer linear network, hence the decision boundaries consist of 'soft' hyperplanes and

inevitably creates overlapping regions [RG98], within which only one expert may be needed (*i.e.* only one sensor subset is effective).

Consequently we adopt a specific model for the gating network, known as localized ME [XJH95]. This modified gating network consists of normalized Gaussian kernels (or any density function from the exponential family) :

$$g_k(y, \theta_k) = P(k|y) = \frac{\alpha_k P(y|\theta_k)}{\sum_{j=1}^K \alpha_j P(y|\theta_j)} \quad (2)$$

with

$$P(y|\theta_k) = \frac{1}{(2\pi)^{d/2} |\Sigma_k|^{1/2}} \exp \left(-\frac{(y - m_k)^T \Sigma_k^{-1} (y - m_k)}{2} \right)$$

where $\theta_k = \{m_k, \Sigma_k\}$ the mean and variance of the Gaussian kernel distribution.

The Gaussian kernels now allow us to divide the input space into soft hyper-ellipsoids. These ellipsoids can overlap, or create localized regions of expertise where a single sensor subset is trustworthy. The choice of Gaussian kernels also have consequences on the learning step, as it conducts to a one-pass maximization step when using the EM algorithm. The convergence rate of EM algorithm is also empirically and theoretically proven to be faster than gradient ascent methods [JX93]. Associated to the Gaussian kernels, it provides guaranteed convergence due to the single loop maximization step (as opposed to the method developed in [Jor93] which consists in a double-loop EM).

For this reason we decide to learn the gating parameters with EM algorithm.

3.3 Learning the mixture parameters

The basic idea of the EM algorithm is to make the assumption that some variables are hidden, in our case the probability that the n^{th} target sample x^n was generated by expert k . Hence we introduce an indicator variable z :

$$z_j^n = \begin{cases} 1 & \text{if target sample } x^n \text{ is generated by expert } j \\ 0 & \text{otherwise} \end{cases}$$

This hidden variable induces mutual competition among experts. It also models the existence of unknown operating contexts which for different subsets of experts are reliable. We complete maximum likelihood estimation on the joint density $p(x, y)$. Rewriting equation (1) with the new gating function and noting the k^{th} expert output conditional density function $\phi_k(x|y)$:

$$p(x|y, \Theta, \Lambda) = \sum_{k=1}^K \frac{\alpha_k P(y|\theta_k)}{\sum_{j=1}^K \alpha_j P(y|\theta_j)} \phi_k(x|y) \quad (3)$$

we obtain joint density

$$p(x, y) = \sum_{k=1}^K \alpha_k P(y|\theta_k) \phi_k(x|y) \quad (4)$$

by making use of Baye's rule on equation (2) to obtain $p(y) = \sum_{j=1}^K \alpha_j P(y|\theta_j)$.

Finally, introducing the indicator variable z to mediate mutual exclusive experts, the joint distribution over hidden and observed variables take the form

$$p(x, y, z) = \prod_{k=1}^K (\alpha_k P(y|\theta_k) \phi_k(x|y))^{z_k} \quad (5)$$

which, by maximum likelihood, leads to the cost function:

$$\mathcal{C} = - \sum_n \sum_{k=1}^K z_k^n \ln(\alpha_k P(y|\theta_k) \phi_k(x|y)) \quad (6)$$

Now the specificity of EM algorithm enters. In the first step we replace the hidden variable z by its expected value. This is the **Expectation step**:

$$\begin{aligned} \mathcal{E}(z_k^n) &:= p(z_k^n = 1 | x^n, y^n) \\ &= \frac{p(x^n | z_k^n = 1, y^n) p(z_k^n = 1 | y^n)}{p(x^n | y^n)} \\ &= \frac{\alpha_k P(y^n | \theta_k) \phi_k(x^n | y^n)}{\sum_{j=1}^K \alpha_j P(y^n | \theta_j) \phi_j(x^n | y^n)} = h_k(x^n, y^n) \end{aligned} \quad (7)$$

Then we maximize the expectation of the cost function by substituting z_k by its expectation $h_k(x, y)$. This is the **Maximization step**:

$$\begin{aligned} \mathcal{E}(\mathcal{C}) &= - \sum_n \sum_{k=1}^K h_k(x^n, y^n) \ln(\alpha_k P(y^n | \theta_k) \phi_k(x^n | y^n)) \\ &= - \sum_n \sum_{k=1}^K h_k(x^n, y^n) \ln(\alpha_k P(y^n | \theta_k)) \\ &\quad - \sum_n \sum_{k=1}^K h_k(x^n, y^n) \ln(\phi_k(x^n | y^n)) \end{aligned} \quad (8)$$

These two terms can be minimized separately. The first one corresponds to minimizing the cost function relative to gating parameters and the last term corresponds to the expert network parameters.

3.4 Achieving mixture of kalman filters

In our context each expert is a particular Kalman filter providing its own estimation based on observation

input y_k^n and parameters λ_k describing specific sensor observation noise and observation selection matrix. Hence $\phi_k(x^n | y^n)$ is obtained by evaluating the output distribution of the k^{th} Kalman filter at point x^n . The maximization step then consists only in minimizing the first term of result (8). Setting partial derivatives relative to α_k (and using Lagrangian multiplier to introduce constraint $\sum_k \alpha_k = 1$), m_k and Σ_k to zero, we obtain new estimates [RG98]:

$$\alpha_k = \frac{1}{N} \sum_n h_k(x^n, y^n) \quad (9)$$

$$m_k = \frac{\sum_n h_k(x^n, y^n) y^n}{\sum_n h_k(x^n, y^n)} \quad (10)$$

$$\Sigma_k = \frac{1}{d} \frac{\sum_n h_k(x^n, y^n) \|y^n - m_k\|^2}{\sum_n h_k(x^n, y^n)} \quad (11)$$

Using these new parameters, we then repeat the expectation and maximization step until convergence, *i.e.* change in the parameter values becomes insignificant.

As we can see, introducing the hidden variable z has two benefits. It models the existence of different regimes our system is likely to encounter, and at the same time it fosters competition among experts, based on the assumption that only one expert is responsible for a sample $\{x^n, y^n\}$.

One common problem with mixture of Kalman filters is that the exact belief state grows exponentially in time. If we consider a set of K filters and an observation length T , then the exact distribution of the state is a mixture of K^T Gaussian distributions. To deal with this exponential growth we use the GPB collapsing method of order 1 (GPB1), and approximate the mixture of filters output distribution with a single Gaussian distribution. At step n , if each filter k provides an output distribution of mean μ_k and variance σ_k we obtain the mixture distribution mean μ_{mix} and variance σ_{mix} [BSLK01]:

$$\begin{aligned} \mu_{mix} &= \sum_{k=1}^K g_k \mu_k \\ \sigma_{mix} &= \sum_{k=1}^K g_k [\sigma_k + (\mu_k - \mu_{mix})(\mu_k - \mu_{mix})^T] \end{aligned}$$

The next transition step is then based on this mixture output, hence accumulating the error introduced by the approximation at each time step. However, it has been shown in [BK98] that the process error remains bounded indefinitely, avoiding the mixture output to become irrelevant.

At runtime, for each time step, we first compute the shared transition step and evaluate weights g_k . As observed in section 4 the weights locally converge to binary values, giving us the opportunity to save computation time by evaluating the update steps for activated experts only. Finally, as usually a single expert is activated, the computation time required by this approach is similar to a basic Kalman filter, with extra computations due to gating weights and collapsing evaluation.

Some drawbacks of the approach are now discussed. First of all, we saw that the gating network inputs are directly wired to the sensor network. For many sensors, including vision and laser scan, the dimension of raw data (for example the number of pixels for a camera) would require high dimension Gaussian kernels, conducing to infeasible gating weights computations. In this case, information should be resumed in pre-extracted features. These features can be specifically extracted for the gating function, or the features already required by the filtering layer.

An other constraint, directly imposed by the nature of the Gaussian kernels, is the unimodal distribution of the regions of expertise. However, some specific sensors may need to be active in separate regions of input space. This would require to model gating probabilities with more complex models, like Gaussian mixture models or Gaussian processes [YWG12]. In the case of altitude estimation, we will however notice that the localized Gaussian kernels provide sufficient clustering capabilities.

In its original implementation the ME framework inputs are synchronized, and the gating network bases its decision on a joint set of observations. Some investigation on the ability for our approach to handle asynchronous data will be led in future work. Meanwhile, for experiments, we simulated synchronous observations by forcing sensors to provide measures at a defined frequency. For the slowest sensors that means we feed the framework with the last measure several times. As we will see this approach doesn't affect the framework ability to make decisions, mainly because difference between inputs frequencies stays low. However, if the difference in frequencies become too important, the system wouldn't provide relevant decisional capabilities.

4 Experiments

4.1 Simulation

This first test case scenario illustrates the system ability to learn decision rules according to sensors charac-

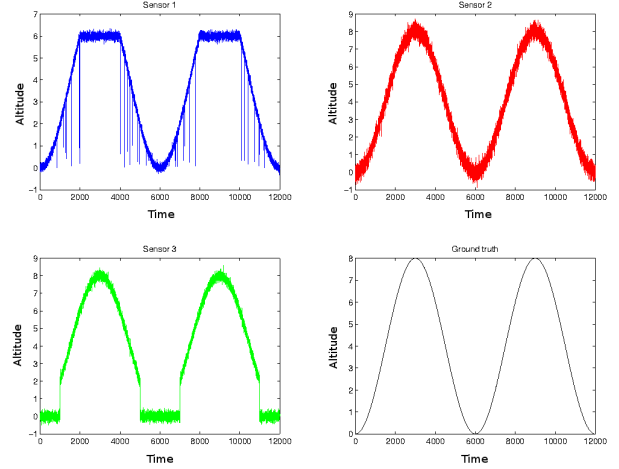


Figure 3: Altitude measures (in meters). Sensor 1 reproduces typical ultrasonic measures, low observation noise, strong outliers occurrences and maximum range threshold. Sensor 2 permanently provides measures with high observation noise. Sensor 3 does not provide relevant measures before reaching 2 meters.

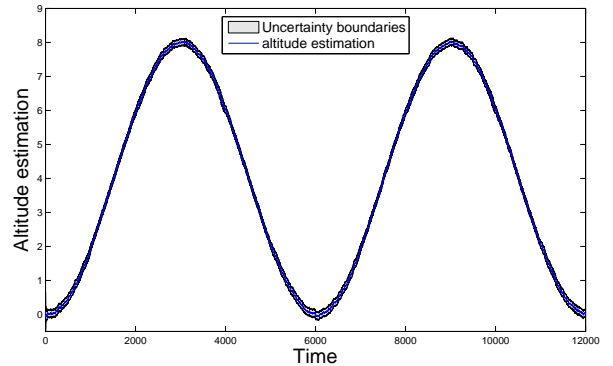


Figure 4: Estimated altitude and uncertainty boundaries on validation dataset. Uncertainty magnitude evolves according to selected sensor.

teristics. This simple example reproduces the take-off and landing phases of a quadrotor UAV. Three sensors are providing direct measures of the altitude with different characteristics, such as observation noise, outliers occurrences and measurement range thresholds (Fig. 3). Each sensor feeds one filter, and all filters share a common transition model (constant velocity). These filters do not implement any rejection method, and provide altitude estimation based on raw data. We train the gating network on a dataset of 12000

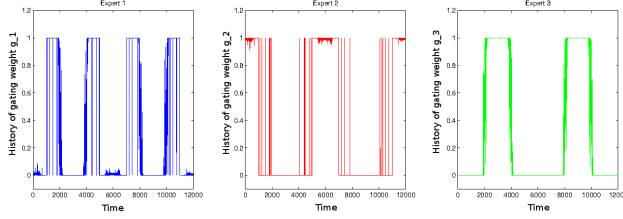


Figure 5: Gating weights history on validation dataset (Expert i is connected to sensor i).

samples reproducing two subsequent take-off/landing sequences. The EM algorithm takes 58 iterations to converge with a convergence threshold of 10^{-6} .

The final estimate and associated uncertainty boundaries for the validation set is shown in Fig. 4. As we can see, the gating network learned to reject outliers and to take into consideration each sensor measurement range. By switching smoothly between sensors, the framework then manages to provide a cleaned estimation output. We note that, as expected from mutual competition between experts introduced during the learning step, the gating network tends to assign binary weights. Hence, mixing only operates during transition periods. As a consequence, the system provides consistent estimation but does not benefit from uncertainty reduction that direct measure fusion would provide (*i.e.* by feeding a filter with multiple sensors).

4.2 Real Data

We now use datasets acquired on a *Paparazzi* quadrotor UAV [BDG⁺06]. Datasets consists of 50Hz synchronized altitude measures provided by an ultrasonic sensor and a barometer as well as accelerations on 3 axis provided by the embedded IMU. Altitude truth is given by a motion capture system. As we can see in Fig. 6, ultrasonic sensor presents strong and frequent outliers we know to be related to thrust level. We also suppose that an external filter gives us an estimation of the barometer offset.

Without additional understanding of the perturbations generated on ultrasonic sensor measures, we apply the mixture of experts framework to show its ability to learn to filter these outliers, and improve the estimation accuracy. For this application we use 3 different experts: one expert based on ultrasonic measures, an other based on barometer measures, and a last one based on both ultrasonic and barometer measures. As we know the presence of outliers in ultrasonic observations is correlated to the thrust, we provide 3 inputs to the gating network: both sensor measures

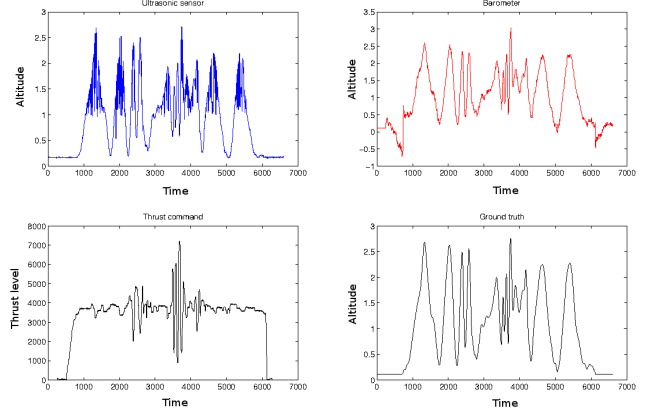


Figure 6: Gating network inputs and altitude truth for validation dataset.

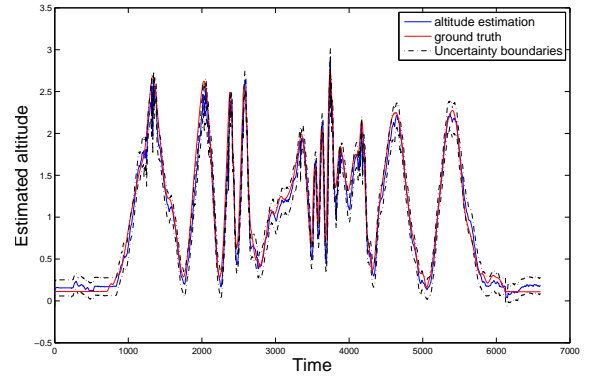


Figure 7: Altitude estimation and uncertainty boundaries using mixture of Kalman filters on validation set.

and the current thrust command. We compare this method to a naive Kalman filter implementation using both barometer and ultrasonic sensor, and a similar filter using *3sigma* rejection scheme on ultrasonic measures (Hence using only barometer information for the update when an ultrasonic measure is rejected). All these filters share the same constant velocity transition model and observation noise. We train the system on a dataset of 5000 samples. After 50 iterations the EM algorithm reaches the convergence threshold of 10^{-6} .

Experience shows that the learned parameters generalize well on different validation sets, always providing similar performances. As we can see on Fig. 7, some outliers are not perfectly filtered. These outliers presumably present unknown characteristics for the gating network, implying that rejection capability could

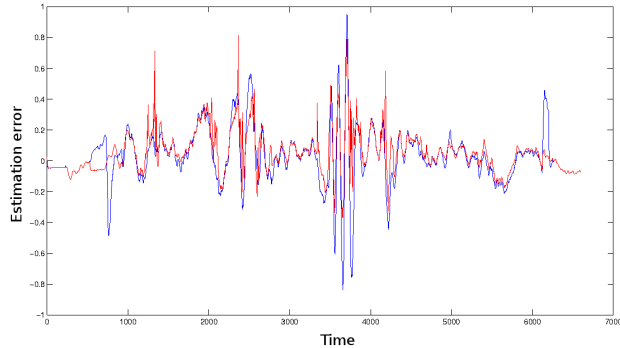


Figure 8: Estimation error relative to altitude truth for mixture of Kalman filter (in red) and Kalman filter with 3-sigma rejection (in blue).

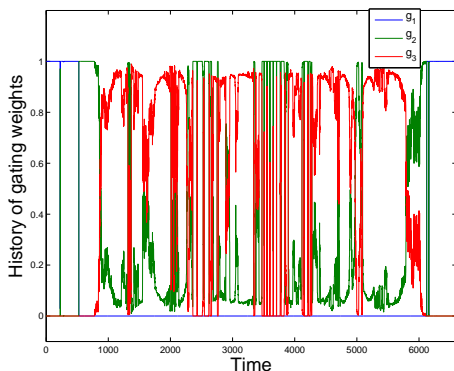


Figure 9: Weights history for expert 1 (ultrasonic sensor + barometer), expert 2 (ultrasonic sensor) and expert 3 (barometer).

be improved by using a larger training set. For all validation sets, the mixture of filter still provides the best RMS estimation error. On the validation set corresponding to figure 6 it provides an RMS estimation error value of 0.142. Naive Kalman filter provides an RMS error of 0.221, and the filter with rejection provides better performance with an RMS error of 0.185. With a gating framework basing its decision on sensor measures only, we found an RMS error of 0.170. This result attests of thrust impact on ultrasonic measures, and of the ability for the framework to take it into consideration as well.

The estimation error improvement provided by the mixture approach (shown in Fig. 8) can be explained by the sensor selection process. We notice that the change in the air flow creates a strong disturbance on barometer measures when we turn motors on and off (between samples 0 and 1000 or after sample 6000).

At this point, and for low altitudes in general, our model learned to promote the ultrasonic sensor, where the Kalman filter can't incorporate this specific knowledge, unless we implement a decision rule by hand. Meanwhile, due to associated offset and its estimation latency, barometer measures are more relevant for higher altitudes and low velocity. Once again, because the ME chooses ultrasonic measures in priority when the UAV is close to the ground it reduces overestimation of the altitude during the fast transition phase between sample 3000 and 4000. The ME approach also decreases underestimation of altitude in this case as it filters ultrasonic outliers based on the strong thrust command value. In comparison, the Kalman filter rejection scheme suffers from barometer latency, punctually becoming coherent with low outliers measures. This corresponds to the highest peak in Fig. 8

This experiment demonstrates the ability of our method to learn an outlier rejection scheme and to learn the different contexts our robot encounters. In this scenario, a context intuitively corresponds to an altitude range and to a thrust command level, *i.e.* the current dynamic of the UAV. The proposed approach outperforms classical filtering methods and, from a practical point of view, appears to be more robust toward filter parameters imprecision.

5 Conclusion

We demonstrated that the mixture of expert framework can be applied to the sensor selection problem. The gating network discovers the different operating contexts and encodes knowledge about sensor reliability through the gating probability distributions parameters. This feature enables the system to automatically select the best suited estimation output, improving robustness regarding filter parameters inaccuracies and inherent sensor characteristics. The modular nature of the system shows it can be extended to more complex configurations by increasing the gating network input space and sensor network as well.

An interesting direction for future work would consist in using more complex models for decision boundaries and extending the method to richer information sources like laser scan and image features. In current implementation the mixture process ignores previous values of the gating parameters. We believe that, through the application of switching state space model, extending the mixture method to its dynamical version would also improve capabilities of the approach.

References

- [BBS88] H.A.P. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with markovian switching coefficients. *Automatic Control, IEEE Transactions on*, 33(8):780–783, aug 1988.
- [BDG⁺06] P. Brisset, A. Drouin, M. Gorraz, P.S. Huard, and J. Tyler. The paparazzi solution. 2006.
- [BK98] Xavier Boyen and Daphne Koller. Tractable inference for complex stochastic processes. In *In Proc. UAI*, pages 33–42, 1998.
- [BSLK01] Yaakov Bar-Shalom, X. Rong Li, and Thiagalingam Kirubarajan. *Estimation with Applications to Tracking and Navigation*. Wiley-Interscience, 1 edition, 2001.
- [CBG97] Wassim S. Chaer, Robert H. Bishop, and Joydeep Ghosh. A mixture-of-experts framework for adaptive Kalman filtering. *IEEE Transactions on Systems, Man, and Cybernetics*, 27, 1997.
- [CBG98] W.S. Chaer, R.H. Bishop, and J. Ghosh. Hierarchical adaptive kalman filtering for interplanetary orbit determination. *Aerospace and Electronic Systems, IEEE Transactions on*, 34(3):883–896, Jul. 1998.
- [HL97] D. L. Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, January 1997.
- [JJNH91] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Comput.*, 3(1):79–87, March 1991.
- [Jor93] Michael I. Jordan. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6:181–214, 1993.
- [JX93] Michael I. Jordan and Lei Xu. Convergence results for the em approach to mixtures of experts architectures, 1993.
- [Mag65] D. Magill. Optimal adaptive estimation of sampled stochastic processes. *Automatic Control, IEEE Transactions on*, 10(4):434–439, Oct. 1965.
- [RG98] Viswanath Ramamurti and Joydeep Ghosh. On the use of localized gating in mixture of experts networks, 1998.
- [SBP⁺07] Sreenivas R. Sukumar, Hamparsum Bozdogan, David L. Page, Andreas Koschan, and Mongi A. Abidi. Sensor selection using information complexity for multi-sensor mobile robot localization. In *ICRA*, pages 4158–4163. IEEE, 2007.
- [SW09] R. Schubert and G. Wanielik. Unifying bayesian networks and imm filtering for improved multiple model estimation. In *Information Fusion, 2009. FUSION '09. 12th International Conference on*, pages 810 – 817, july 2009.
- [TBF05] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.
- [TSL⁺12] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I.L. Grix, F. Ruess, M. Suppa, and D. Burschka. Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue. *Robotics Automation Magazine, IEEE*, 19(3):46–56, sept. 2012.
- [XJH95] Lei Xu, Michael I. Jordan, and Geoffrey E. Hinton. An alternative model for mixtures of experts. 1995.
- [YWG12] Seniha Esen Yuksel, Joseph N. Wilson, and Paul D. Gader. Twenty years of mixture of experts. *IEEE Trans. Neural Netw. Learning Syst.*, 23(8):1177–1193, 2012.