



HAL
open science

Speedometer Fault Detection and GNSS Fusion using Kalman Filters

Hugo Kerhascoet, Johann Laurent, Eric Senn, Frédéric Hauville

► **To cite this version:**

Hugo Kerhascoet, Johann Laurent, Eric Senn, Frédéric Hauville. Speedometer Fault Detection and GNSS Fusion using Kalman Filters. OCEANS'16 MTS/IEEE, Sep 2016, Monterey, United States. pp.8. hal-02141279

HAL Id: hal-02141279

<https://hal.science/hal-02141279>

Submitted on 27 May 2019

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.

Speedometer Fault Detection and GNSS Fusion using Kalman Filters

Hugo Kerhascoët
Lab-STICC and IRENav
Université de Bretagne Sud
nke Marine Electronics
Email: hkerhascoet@nke.fr

Johann Laurent and Eric Senn
Lab-STICC
Université de Bretagne Sud
Email: surname.name@univ-ubs.fr

Frédéric Hauville
M2EN
Institut de Recherche de l'Ecole Navale
Email: frederic.hauville@ecole-navale.fr

Abstract—Navigation systems used in racing boats require sensors to be more and more sophisticated in order to obtain accurate information in real time. To meet the need for accuracy of the surface speed measurement, the mechanical sensor paddle wheel has been replaced by the ultrasonic sensor. This ultrasonic sensor measures the water speed precisely and with very good linearity. Furthermore, by its principle of operation, it measures the water flow several centimetres from the sensor, which puts it outside the boundary layer, the region close to the hull where the flow is disturbed. However, this sensor has several drawbacks: it is quite sensitive and if the flow contains too many air bubbles, the sensor picks them up, which can happen quite frequently on boat with a planing hull. Another limitation of this sensor is its low frequency measurement rate. In this paper we explain the techniques used based on Kalman filters to address these shortcomings, firstly by identifying the inaccurate measurements caused by inadvertent dropouts, then by improving the useful sensor frequency with GNSS data fusion.

NOMENCLATURE

SOG	Speed Over Ground (knots).
SOW	Speed Over Water (knots).
$GNSS$	Global Navigation Satellite System.
X_k	State vector.
y_k	Measurement vector.
F_k	State transition matrix.
H_k	Observation matrix.
w_k	Process noise.
v_k	Observation noise.
Q_k	Process noise covariance matrix.
R_k	Measurement noise covariance matrix.
K_k	Kalman gain matrix.
P_k	State covariance matrix.
σ_{SOG}	GPS speed standard deviation.
σ_{SOW}	Speedometer standard deviation.
ε_{AccSOW}	Model noise.
r	Fault detection threshold.
Te	Integration time.
res_k	Measurement residual.
S_k	Measurement covariance.

I. INTRODUCTION

Surface speed corresponds to boat speed relative to the water. It is a major element in the wind measurement chain, because it is one of the basis for calculations to determine,

among others, the true wind speed and angle. As it is data that occurs at the basis of the reconstruction of the true wind, the measurement error is transmitted and amplified onto the ensuing results. Sensors commonly used are the paddle wheel, the wheel is driven by the water flow. The boat speed is then determined by measuring the rotation speed of this wheel. This kind of sensor presents two major problems: the first one is that the mobile part is immersed in sea water, so it is quickly covered by seaweed that leads to variations in its behaviour. The second is that it measures the flow along the hull, in the boundary layer; where the flow is disturbed and not representative of the actual boat speed. In conclusion, the sensor measurements are neither precise nor linear.

To address the shortcomings of the paddle wheel sensor, a new kind of sensor has been developed: an ultrasonic sensor. Its main advantage is that it has no moving parts, so it is not sensitive to the marine environment.¹ The second one is that it measures the flow velocity several centimetres away, outside the boundary layer; where the speed measurement is less disturbed by the hull. The measurements are thus highly accurate and linear over the full speed range. However, new problems occur: the sensor is inaccurate when it is near the limit of immersion. This is often the case with planing hulls or when the boat is heeled over too far. Indeed in this case, the sensor is no longer able to acquire the real data so it drops out. This phenomenon is shown in Figure 1.

We can see that surface speed drops to 6kts whereas the ground speed remains stable. The sensor is therefore in an error state. The main problem in this situation is not that the sensor is no longer able to provide a valid speed, although the fact of no longer having surface speed is an issue, it is even more problematic to rely on completely erroneous information while thinking it is accurate because all the information and rules that are using this data are deceived.

Currently, the method performed by the electronic sensor to detect these dropouts is not efficient enough because this state is detected after 2 or 3 erroneous samples (as shown in Figure

¹It is a flush sensor that can be covered with gel coat or anti-fouling as the rest of the hull.

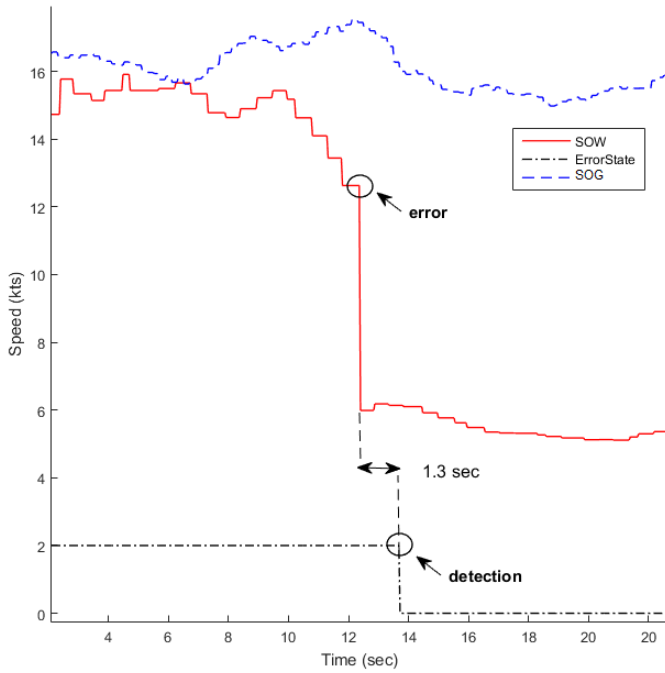


Fig. 1. Dropout example. Sea recording on a 60" Imoca. (ErrorState 2 is for valid data, 0 for errors).

1). This delay leads to some errors in the computation of the true wind angle for instance, which is a concern for security when the boat is under autopilot. Although this period is very short, the autopilot will make decisions based on incorrect information which can for example, lead to unwanted gybes. The objective of this paper is to propose a methodology to detect these errors as soon as they appear, and not after a delay as is currently the case. The present paper does not seek to compare the performance and accuracy of the sensor but focuses on the error detection.

II. FAULT DETECTION METHODOLOGY

In this section we will present our methodology in order to detect the sensor's errors. First, we will present the ultrasonic sensor.

A. The ultrasonic sensor

The purpose of this section is not to describe how this sensor works, but to understand its basic principle of operation.

As shown in Figure 2, the sensor monitors the flow of particles at 8 centimetres with its ultrasonic beam.

This means that in this area, the water must be homogeneous, without large air bubbles, otherwise the sensor cannot follow the signal, as shown in Figure 1. As already stated, it is not so much that sometimes the sensor picks up that is constraining, because this is rarely the case in normal conditions of use, but the fact that there is a delay in the actual error detection. The goal is therefore to provide a solution that detects errors before they are transmitted to the rest of the navigation system.

Our approach is presented below.

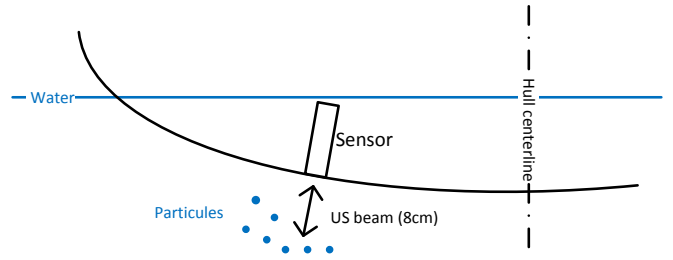


Fig. 2. Ultrasonic sensor working principle.

B. Methodology of detection

With the aim of detecting errors at their earliest stage, we will move towards a filter that features predictive capabilities. By observing the correlation between the predicted and the measured value, we could then estimate the validity of the latter. Prediction is one of the main characteristics of the well-known Kalman filter [1], which is why it was chosen. Mehra and Peschon [2] were the first to apply Kalman filters for fault detection.

Generally the practical implementation of a Kalman filter is quite complicated, but in this case, the one used is relatively straightforward, firstly because the processed variables are linear; this enables us to avoid linearization steps. And secondly because of the observation (13) and transition (14) matrices which are of a limited order.

The filter is characterized by two equations: the state equation and the measurement equation (1). y_k represents the measurement vector, in this case the surface speed and X_k its estimate.

$$\begin{cases} X_k = F_k X_{k-1} + w_{k-1} \\ y_k = H_k X_k + v_k \end{cases} \quad (1)$$

Measurement equation (1) refers to H_k the observation or measurement matrix which associates the measurement vector to the state estimate.

F_k is the state transition matrix which models the transition between two estimates.

w_k and v_k respectively represent, model noise and measurement noise which are assumed to have a normal probability distribution.

$$p(w) \sim N(0, Q_k) \quad (2)$$

$$p(v) \sim N(0, R_k) \quad (3)$$

(2) and (3) state that w_k and v_k are not biased with variance Q_k and R_k .

The state of the filter is represented by two variables:

- $\hat{X}_{k|k}$ the a posteriori state estimate at time k giving observations up to and including at time k .
- $P_{k|k}$ the a posteriori error covariance matrix.

We compute the filter in two recursive steps, prediction and estimation.

In the context of error detection, it is the predictive ability of the filter that we are going to use in order to detect the sensor errors.

1) *Prediction*: This step predicts the a priori estimate of the state vector equation (4) and the covariance of the estimation error equation (5) of the system at time k from the previous state.

$$\hat{X}_{k|k-1} = F_k \hat{X}_{k-1|k-1} \quad (4)$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \quad (5)$$

After the prediction comes the update step.

2) *Update*: The update step calculates from the a priori error covariance $P_{k|k-1}$ (5), the Kalman gain (6) which minimizes the covariance of the estimated error. A posteriori estimate of the state vector (7) and a posterior estimate covariance (8) are both updated with the new gain (6) and measurement vector y_k .

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T - R_k)^{-1} \quad (6)$$

$$\hat{X}_{k|k} = \hat{X}_{k|k-1} + K_k (y_k - H_k \hat{X}_{k|k-1}) \quad (7)$$

$$P_{k|k} = P_{k|k-1} - K_k H_k P_{k|k-1} \quad (8)$$

C. Implementation

The measurement vector (9) is limited to one dimension: the raw signal. As it has been observed previously, the aim of this work is to detect sensor errors only by observing its evolutions, without external information which could also be incorrect.

$$y = (SOW) \quad (9)$$

We restrict the state vector to a first order system, i.e. we use one derivative to model the system. As we try to model the speed, this first order is adapted to a constant acceleration model. This approximation must be taken into account in the process noise covariance matrix (15).

$$X = \begin{pmatrix} \widehat{SOW} \\ \widehat{AccSOW} \end{pmatrix} \quad (10)$$

Using (1), (9) and (10) we can deduce both the state matrix (13) and observation matrix (14).

$$(SOW) = [1 \ 0] \times \begin{pmatrix} \widehat{SOW} \\ \widehat{AccSOW} \end{pmatrix} \quad (11)$$

$$\begin{pmatrix} \widehat{SOW} \\ \widehat{AccSOW} \end{pmatrix}_{k+1} = \begin{bmatrix} 1 & Te \\ 0 & 1 \end{bmatrix} \times \begin{pmatrix} \widehat{SOW} \\ \widehat{AccSOW} \end{pmatrix}_k \quad (12)$$

Hence:

$$H = [1 \ 0] \quad (13)$$

$$F = \begin{bmatrix} 1 & Te \\ 0 & 1 \end{bmatrix} \quad (14)$$

The transition matrix (14) assumes that acceleration is constant during the time steps. If it is not the case, we will have to adjust the noise model applied to acceleration ε_{AccSOW} , depending on the boat's acceleration / deceleration capability, brought back onto the iterative time Te .

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & \varepsilon_{AccSOW}^2 \end{bmatrix} \quad (15)$$

The Q matrix design is one of the trickiest aspects of the Kalman filter. If Q is too small then the filter will be overconfident and if it is too large the estimate will be too affected by measurements. Normally this matrix can be more complex, but in order to limit the computing cost and as the iterative time is small, we have restricted it to (15). Measurement noise σ_{SOW} depends on sensor characteristics, and more precisely on its inaccuracy. Techniques for a good variance estimate are proposed in [3].

$$R = [\sigma_{SOW}^2] \quad (16)$$

The measurement noise covariance matrix (16) will influence the weight given to a new measurements during the prediction step (6). The more its value increases, the less the sensor data will be taken into account. In the case of our application, model and sensors noises are considered to be static, which is why subscripts k of their respective matrices are no longer present in the equations.

We have seen the implementation of the Kalman filter; now let us look at its error detection abilities.

1) *Fault condition*: The filter's predictive ability enables us to know if we are dealing with bad data, by comparing the difference between the measured y_k value and the estimated \hat{X}_k value. The measurement residual is computed with equation (17):

$$res_k = y_k - H_k \hat{X}_{k|k-1} \quad (17)$$

Measurement residual is also called innovation because once weighted by the Kalman gain during the update step (7), it determines the evolution of the prediction compared to the last estimate.

In Figure 3 we can observe the evolution of the \widehat{SOW} residual with respect to the uncertainty drawn from the state covariance (8). In this experiment we are in the operating conditions of the sensor, so it is normal to observe that most of the data is in the 1σ standard deviation bounds (68%).

Now we have to determine if the last data from the sensor is valid or not by using the residual behaviour. Several methods

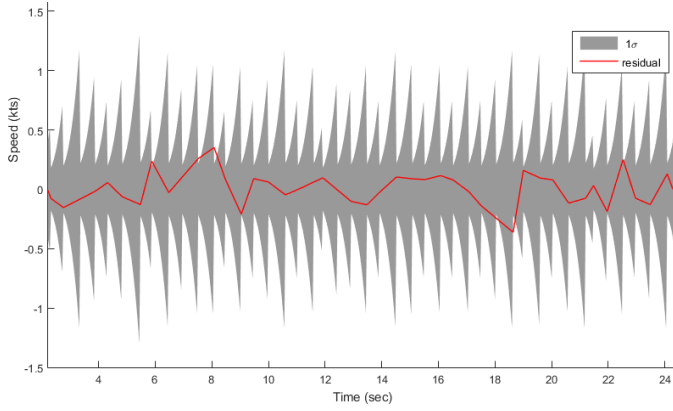


Fig. 3. Residual vs 1σ uncertainty.

from the literature can be used as Zarchan [4] or Salehfar [5]. Zarchan compare the absolute of the residual with a multiple of the standard deviation of the innovation covariance \sqrt{S} .

$$S_k = H_k P_k H_k^T + R \quad (18)$$

This method requires huge computational resources and as we use an embedded system it is not suitable.

The Salehfar method compares the square of the residual with a pre-specified detection threshold r :

$$res_k^2 > r^2 \quad (19)$$

If the condition (19) is satisfied then the sensor is reported faulty.

It is the correct setting of r which allows bad-data detection. Its determination is performed empirically but the good value is usually close to 10 times the sensor standard deviation [6] $r = 10 \times \sigma_{SOW}$

Figure 4 shows an error condition detected with the filter that has been set up.

While the boat was under way without making any particular manoeuvre, the raw speed suddenly drops to 5 knots.

This is clearly impossible but the error state communicated by the sensor (2 is for normal operation and 0 for error) occurs 960 milliseconds too late. For about one second, the entire measurement chain which uses this information was incorrect, this is not acceptable.

By using the fault detection rule (19) based on the prediction of the filter, it is able to identify bad measurements from their very first iteration.

The results from Figure 4 show that we are able to detect if the data provided by the sensor are valid or not. In this example if the values are not valid then no speed is communicated to the system, however a speed measurement is required.

In the next section, we will present a solution in order to

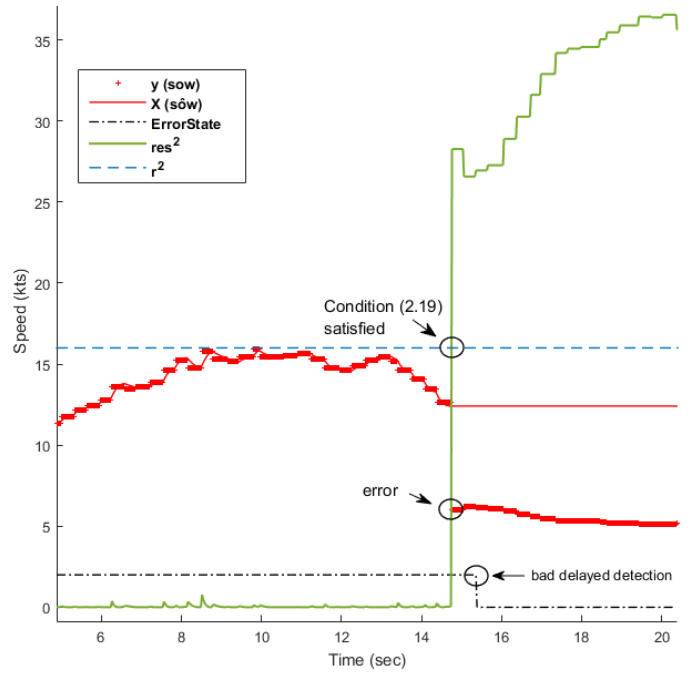


Fig. 4. Fault detection. Sea recording on a 60' Imoca.

provide a speed value when the one provided by the transducer is no longer valid.

III. SOW SUBSTITUTION

In order to provide an alternative for surface speed when the sensor is unable to provide a valid one, we propose to rely on another sensor: the GNSS receiver. Firstly, because of its ready availability and secondly because it relies on a different technology. Both sensors therefore have few common causes of breakdowns.

However the GNSS receiver provides speed information which is not in the same referential as the one from the speedometer. Indeed the speedometer measures the flow velocity of the water, therefore the boats moving speed relative to the water surface. Unlike the GNSS receiver which measures its speed in the Earth referential, relative to the ground.

A. Assumptions

Surface and ground speed differ with the effect of current and leeway. To simplify the calculations we consider that the current and leeway are constant over the iteration time. So we can say that surface and ground speed differ from an almost constant offset.

This may also be written as:

The derivative of the speed relative to the water and the ground are identical.

Compared to the previous implementation, measurement (20) and state (22) vectors are enhanced by ground speed.

$$y = \begin{pmatrix} SOW \\ SOG \end{pmatrix} \quad (20)$$

We retain that the boat acceleration is constant between two iterative steps so:

$$\widehat{SOG}_k = \widehat{SOG}_{k-1} + Acc\widehat{SOG}_{k-1} \times Te \quad (21)$$

$$X = \begin{pmatrix} \widehat{SOG} \\ Acc\widehat{SOW} \\ \widehat{SOG} \\ Acc\widehat{SOG} \end{pmatrix} \quad (22)$$

Adding *SOG* to the state vector (22) aims to take advantage of the GNSS receiver information when surface speed is no longer available. We must now define the method that takes advantage of this new sensor.

B. Tolerant fault estimator

Unlike the previous implementation, transition and observation matrices are no longer constant.

We will switch between two models with their respective matrices based on the state of the sensor, if the sensor is detected as being in default or not. We use subscript $.N$ for normal operation mode and $.F$ when the sensor has been identified as faulty.

1) *Normal operating mode*: When the transducer is operational, the model is fed by each sensor, the observation matrix (23) includes two inputs, the whole measurement vector.

$$H_N = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (23)$$

We find the same pattern (14) in the transition matrix (24) since it uses the same model and same filter order as for fault detection, with the additional dimensions for speed over ground.

$$F_N = \begin{bmatrix} 1 & Te & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & Te \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (24)$$

2) *Substitution mode*: When condition (19) is satisfied, information returned by the faulty sensor must be rejected. Compared to (23) the substitution observation matrix (25) no longer takes into account information from the sensor to feed the model.

H_F prevents the use of bad data during the a posteriori estimate (7).

$$H_F = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (25)$$

It is the previous assumption which suggests that the derivative of the speed relative to the water and the ground are identical that allows us to maintain an evolution of the surface speed based on the variations of the other. The transition matrix (26) no longer uses the *SOW* accelerations to update the estimate but uses those from the *SOG*.

$$F_N = \begin{bmatrix} 1 & 0 & 0 & Te \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & Te \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (26)$$

The a priori estimate \widehat{SOW} is now supplied by the *SOG* derivative. The evolutions of \widehat{SOW} are then based on those from the speed over ground estimate.

Assumptions and approximations made must now be taken into account.

C. Noises matrices

The sensor's characteristics do not change whether they are in fault or not. There is no reason to change their noise, σ_{SOW} is therefore the same. Although the GPS receptor used is very accurate, the order of magnitude of σ_{SOG} is two times higher than σ_{SOW} .

$$R = \begin{bmatrix} \sigma_{SOW}^2 & 0 \\ 0 & \sigma_{SOG}^2 \end{bmatrix} \quad (27)$$

Concerning noise models, when the transducer is estimated to be valid, the noise process matrix is similar to (15).

$$Q_N = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \varepsilon_{AccSOW}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \varepsilon_{AccSOG}^2 \end{bmatrix} \quad (28)$$

However Q_F does not refer to the same model, therefore it as to be adapted:

$\varepsilon_{\Delta Acc}$ represents the modelling error introduced by the assumption that the ground and surface acceleration are identical. As ε_{AccSOG} and $\varepsilon_{\Delta Acc}$ are two independent variables the standard deviation of their sum is equal to: $\sqrt{\varepsilon_{AccSOG}^2 + \varepsilon_{\Delta Acc}^2}$

ε_{SOW} is no longer equal to zero because it would mean that the model is absolutely correct : $\varepsilon_{SOW} = \varepsilon_{\Delta Acc} \times Te$

During substitution, Q becomes:

$$Q_F = \begin{bmatrix} \varepsilon_{SOW}^2 & 0 & 0 & 0 \\ 0 & \varepsilon_{AccSOG}^2 + \varepsilon_{\Delta Acc}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \varepsilon_{AccSOG}^2 \end{bmatrix} \quad (29)$$

D. Results

We have applied our tolerant fault estimator to the same application as in (19). Results are shown in Figure 5.

The first interesting thing to notice in this case is the fact that the ErrorState indicates that the sensor status remains at 2. This means the sensor is not aware of its own fault, despite the relatively long duration (4 sec).

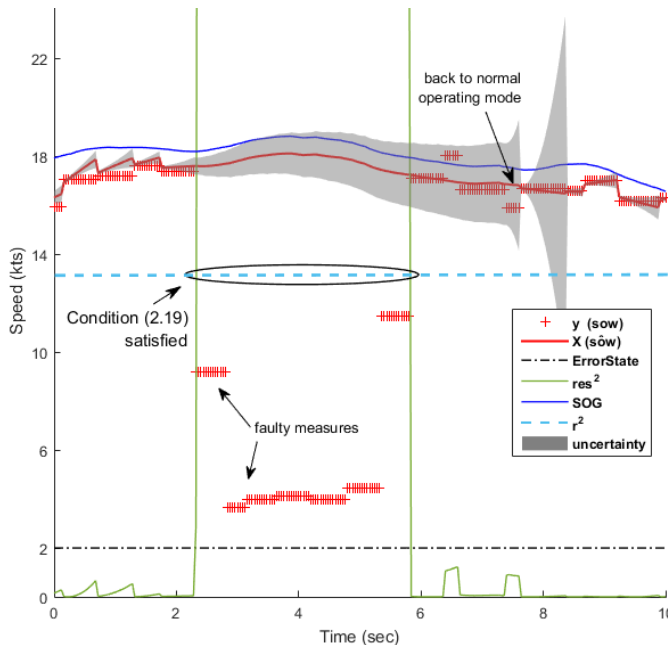


Fig. 5. *SOW* substitution during a fault.

The Fault detection is performed at the right moment as soon as the condition (19) is satisfied. At this moment the filter switches to the substitution model. *SOW* estimate X starts to follow *SOG* variations during all the substitution phases and we can see that the false measurements are correctly rejected.

Throughout this phase, the filter uncertainty increases, due to ε_{SOW} which is added to each iteration of (5) and which cannot be reduced during (8) since $H_F(1,1)$ is equal to zero.

A return to normal operating mode does not occur as soon as the condition (19) is no longer met because we want to ensure that the sensor has recovered the signal for a few iterations. The artefact of uncertainty which appears at the hang-up is due to the switching of the covariance matrices which have lost their continuity. It is not of great concern because we want to give more importance to the measurements during this transition phase.

Our system now enables the detection of errors and proposes the use of substitution data so that the rest of the measurement chain continues to operate properly. Furthermore our approach also helps to improve signal quality when the sensor is not in an error state. The following section will present this aspect.

IV. SIGNAL IMPROVEMENT

The signal processing functions developed in the previous sections allow possibilities other than the fault detection. As the tools are in place, we will use them to improve the signal quality.

The ultrasonic sensor is used in a navigation system that calculates and provides commands to the autopilot 25 times

per second, but the sensor provides new data only 2 to 4 times per second, so there are two issues with this situation. The first is that the data refresh rate is not fixed, so it must be taken into account when applying the Kalman filter.

The second is that the autopilot computes its algorithms at a higher frequency than the one provided by the sensor. To be efficient the sensor should provide its values as fast as the autopilot computes its own.

A. Time varying filter

The information is coming from the sensor at random intervals. It is necessary to take this into account during the predicting step to propagate the state covariance (5). The transition matrix (14) needs to be reassessed to take into account the exact time since the last iteration: Te_k .

$$F = \begin{bmatrix} 1 & Te_k \\ 0 & 1 \end{bmatrix} \quad (30)$$

Likewise, the process noise covariance matrix (15) must be revised as it takes into account the acceleration model noise ε_{AccSOW} itself related to Te_k . Q becomes:

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & \varepsilon_{AccSOW}^2 Te_k^2 \end{bmatrix} \quad (31)$$

The next section presents what can be done between the measurement updates.

B. Improved frequency

As the navigation system runs its algorithms at 25Hz it is preferable to do the same with the Kalman filter, or 6 to 12 times more often than the sensor refresh rate. The technique is simple, as planned by the filter:

The prediction steps (4) & (5) are performed at each iteration. Te used in the transition matrix (14) is constant and worth 1/25. However the process noise covariance matrix (15) increase with each new prediction steps because the acceleration model noise ε_{AccSOW} refers to time spent since the last update step.

When a new data from the sensor is available, update steps (6), (7) &(8) are performed.

Figure 6 shows the filter 25Hz predictions (red dots) between the measurements updates (o). As the estimate takes account of acceleration, the signal is continued until the next measurement update. When this acceleration is well modelled, prediction coincides with the next sensor measurement. The constant line corresponds to the shape of the signal that we would be forced to use without the prediction: holding the last measurement until the new one.

Note in Figure 6 that the uncertainty increases with the square of the time since the last measurement. This is in agreement with the noise covariance matrix (15), which increases with each prediction iteration without new measurement updates.

The next section will present the complete methodology flow (from fault detection to the improved signal).

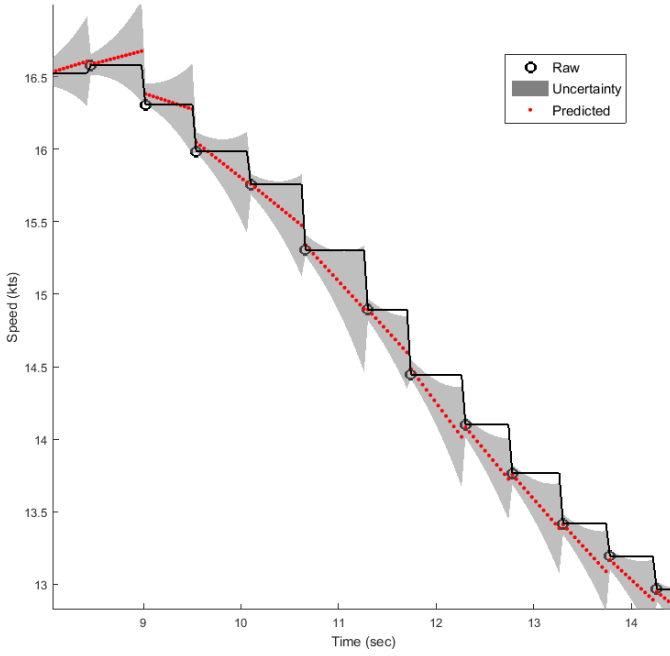


Fig. 6. Signal prediction between the measurement update.

C. Methodology flow

Figure 7 summarizes the decision process for updating the model at 25Hz.

Figure 7 shows the process used when the sensor is not at fault, as for the example in Figure 6. At each iteration, if no new sensor data is available, then the signal is constructed using the prediction. If new data is available, then we first apply the fault detection method (section II). If the outcome is positive we switch to the substitution process (section III). If the data is good then it is used to update the estimate.

D. Adjustable Process Noise

We have seen in Figure 6 that the speed prediction between measurement updates is suitable when we are in conditions described by the model, if the acceleration is close to a constant. However this is no longer the case when the boat switches from an acceleration phase to a deceleration phase, and vice versa. Indeed, in this situation the model approximation is borderline and tends to delay the signal. The first-order filter is no longer optimal.

It is the situation presented in Figure 8: where the sensor is not in fault but subjected to waves. The resulting readout speeds are very fluctuating, so estimated accelerations induce errors and add noise to the signal. Therefore it is necessary to find a law that will detect these circumstances and will then correct the behaviour of the model.

Shalom [7] described a method adapted to this situation; Adjusting the process noise Q according to criteria based on normalized residual squared:

$$\epsilon_k = res_k^T S_k^{-1} res_k \quad (32)$$

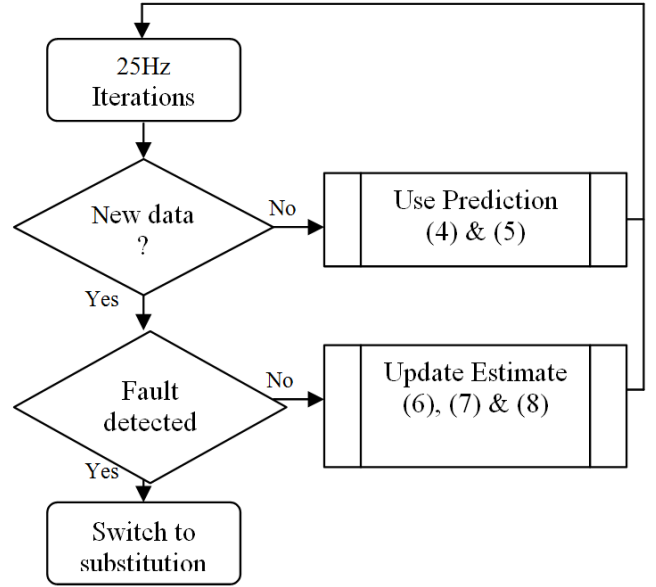


Fig. 7. Flowchart.

With S_k measurement covariance described in (18)

(32) offers two possibilities, either Q is proportional to ϵ or increases by a scaling factor whenever ϵ exceeds a threshold. The second solution adopted: switching between two noise covariance matrices Q whenever ϵ is higher or lower than a threshold.

$$\epsilon_k > \epsilon_{max} \quad (33)$$

The switching threshold ϵ_{max} is chosen such that the probability of it being reached under normal conditions is small. Normalizing the residual by the measurement covariance ensures that the residual is disassociated from measurement noise. Choosing ϵ_{max} has to be done empirically but can be from a baseline in the order of 3 to 5 times the sensor/signal standard deviation σ_{SOW} .

When condition (33) is reached, the process noise ϵ_{AccSOW} is increased which will give more weight to the measurements. So the estimate will adhere to the measurements in such situations; this case is presented in Figure 9 which uses the same signal as Figure 8 but adjusting Q when condition (33) is met. The estimates do not add more noise; the behaviour of the filter is comparable to a low pass filter.

V. CONCLUSION

In this paper, we have shown that even if the ultrasonic sensor is intrinsically accurate, sometimes errors may appear. In this case, all computations derived from these data also generate errors which can be harmful to the system's performance (here on a boat). In order to detect these faults, we proposed the use of a Kalman filter that allows us not only, due to its

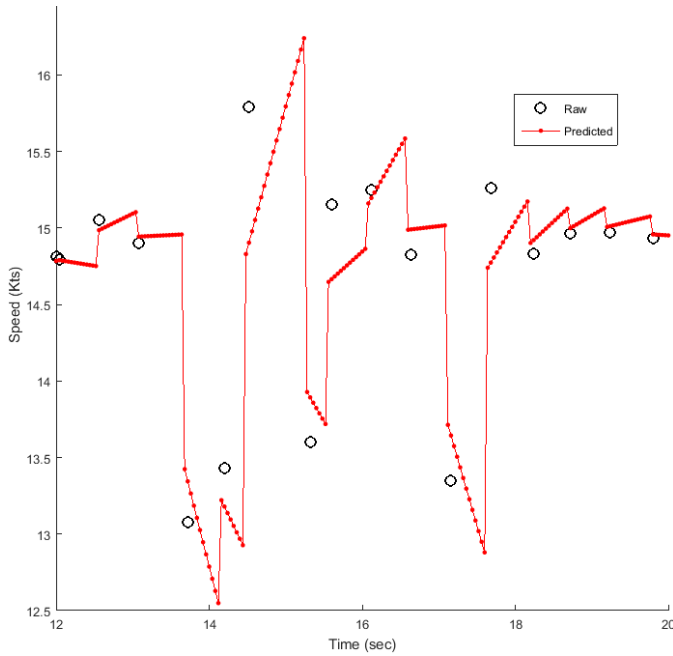


Fig. 8. Poor acceleration estimate.

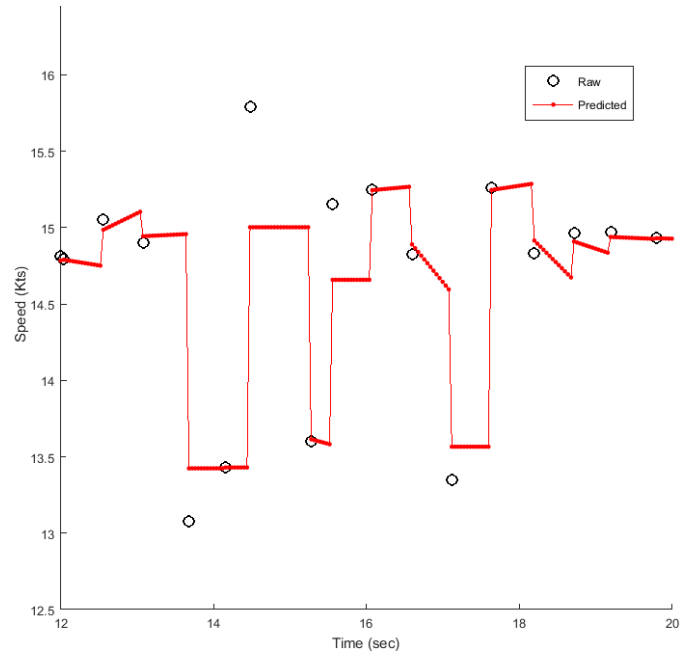


Fig. 9. Here, see adjusted process noise.

predictive property, to detect the sensor's errors (cf. section II) but also to provide an alternative way to obtain a boat speed measurement when the sensor is in fault (cf. section III). The implementation of this filter also improves the signal quality when the sensor operates correctly, indeed we are able to use the predictive property of the Kalman filter in order to predict the sensor data between two measurement cycles (cf. section IV). However, particular attention to the noise model must be given in order to adjust it when the transition matrix no longer correctly represents the behaviour.

We have shown that with our proposed methodology, the ultrasonic sensor is now reliable and even more accurate.

REFERENCES

- [1] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME - Journal of Basic Engineering*, no. 82 (Series D), pp. 35–45, 1960.
- [2] R. Mehra, "On the identification of variances and adaptive kalman filtering," *IEEE Transactions on Automatic Control*, vol. 15, no. 2, pp. 175–184, Apr 1970.
- [3] R. Mehra and J. Peschon, "An innovations approach to fault detection and diagnosis in dynamic systems," *Automatica*, vol. 7, no. 5, pp. 637 – 640, 1971.
- [4] P. ZARCHAN and M. HOWARD, *Fundamentals of Kalman Filtering: A Practical Approach*, 209.
- [5] H. Salehfar and R. Zhao, "A neural network preestimation filter for bad-data detection and identification in power system state estimation," *Electric Power Systems Research*, vol. 34, pp. 127–134, 1995.
- [6] C.-h. Huang, K.-k. Shih, C.-h. Lee, and Y.-j. Wang, "Application of Kalman Filter to Bad-Data Detection in Power System," no. May, 2010.
- [7] Y. Bar-Shalom and X. Li, *Estimation with Applications to Tracking and Navigation*, 2001.