The Two-State Implicit Filter Recursive Estimation for Mobile Robots

Michael Bloesch [ID], Michael Burri [ID], Hannes Sommer, Roland Siegwart [ID], and Marco Hutter [ID]

Abstract—This letter deals with recursive filtering for dynamic systems where an explicit process model is not easily devisable. Most Bayesian filters assume the availability of such an explicit process model, and thus may require additional assumptions or fail to properly leverage all available information. In contrast, we propose a filter that employs a purely residual-based modeling of the available information and thus achieves higher modeling flexibility. While this letter is related to the descriptor Kalman filter, it also represents a step toward batch optimization and allows the integration of further techniques, such as robust weighting for outlier rejection. We derive recursive filter equations that exhibit similar computational complexity when compared to their Kalman filter counterpart—the extended information filter. The applicability of the proposed approach is experimentally confirmed on two different real mobile robotic state estimation problems.

Index Terms—Sensor fusion, probability and statistical methods, localization.

I. INTRODUCTION

STATE estimation represents an essential aspect in many engineering problems. Especially within the fast growing and progressing field of mobile robotics, a reliable and easily deployable state estimation is indispensable. In order to improve reliability, a combination of different sensor modalities is often desirable as it provides redundancy and avoids the use of self-contained but expensive sensor devices. However, the use of appropriate sensor fusion algorithms is a prerequisite in order to obtain a consistent estimation output. This entails a correct modeling of the individual sensors and the combination into a unified problem formulation.

A majority of the work on Bayesian filters assume the availability of an *explicit* dynamic system model with *process* and *measurement* models. Formulated as a time-discrete dynamic system this has the following form:

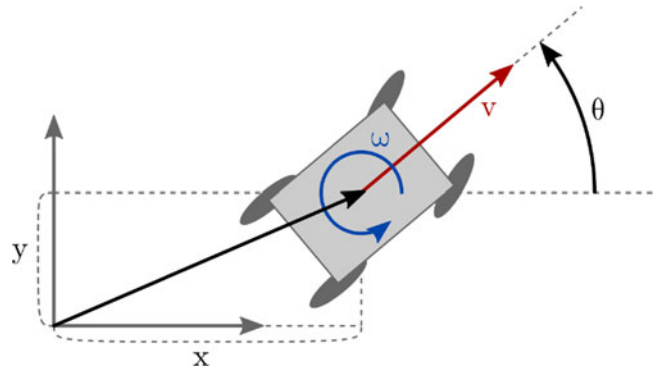$$x_k = f(x_{k-1}, u_k, w_k), \tag{1}$$

$$z_k = h(x_k, n_k), \tag{2}$$

Fig. 1. Toy example: 2D robot with global position coordinates x and y, global orientation θ, forward velocity v and rotational rate ω. Given measurements of the position (x and y), the velocity v, and the rotational rate ω, a simple EKF can be used to estimate all the states (including the heading θ). But what if no measurement is available for the rotational rate ω? Or what if we have two independent velocity measurements?

where w_k and n_k are noise terms and where, in the context of state estimation, both u_k and z_k are observed quantities. For linear and Gaussian models the well-known Kalman filter [1] represents the optimal solution to the associated estimation problem. Various extensions have been proposed in order to improve different aspects. Examples include the extended Kalman filter (EKF) [2], the iterated extended Kalman filter (IEKF) [3], the unscented Kalman filter (UKF) [4], the information filter [5], or square-root filters [6].

Whether a specific sensor modality is associated with the *process* (1) or *measurement* (2) model depends on the choice of filter states. In many mobile robotic problems a combination of both is often employed. For instance, for the toy example depicted in Fig. 1, a typical solution would integrate the velocity and rotational rate measurements in an odometry-driven *process* model and subsequently update the state with the location measurements. Classical Kalman filters require the formulation of an *explicit process* model as described in (1) and cannot deal with under- or over-constrained situations. For under-constrained cases (e.g. the rotational rate measurement is not available) it would require additional assumptions on the dynamics of the heading (e.g. a random walk (RW)) and thus potentially introduce a bias effect. For over-constrained cases (e.g. two independent velocity measurements are available) either a pre-processing step is required, or the filter state can be augmented with differential quantities such as velocities or accelerations. The later approach would enable to include all measurements by means of *measurement* models but would in turn

require the modeling of the dynamics of the introduced differential states. In summary, the requirement for an *explicit process* model restricts the flexibility during modeling and makes the design of an estimation framework more difficult.

In the case of multiple *process* models, solutions have been investigated in the field of track-to-track fusion. The goal is to combine locally computed estimates within a decentralized scenario where multiple sensors exchange information by following a given communication pattern. In order to obtain good performance it is important to consider the cross-correlations between the estimates [7], [8]. In some cases different state representations may be involved as well [9]. However, the obtained results are typically sub-optimal when compared to a centralized solution [10], and thus are less relevant in the present case.

A more general treatment of under- or over-constrained processes can be attained when employing descriptor systems, which is a concept better known in the control community [11]. A descriptor system typically has the following time-discrete linear form, where in essence an *implicit* formulation of the *process* model is employed:

$$\bar{E}x_k = \bar{A}x_{k-1} + \bar{B}u_k + w_k, \tag{3}$$

$$z_k = \bar{C}x_k + n_k. \tag{4}$$

Skliar and Ramirez [12] show that, although the matrix $\bar{E}$ is often assumed to be square and non-singular and thus classical Kalman filtering could be applied, a numerically more stable and fast solution can be employed when directly leveraging the descriptor system form. For general matrices, the descriptor system yields a very high modeling flexibility. Since, in state estimation, the control input u_k and the measured output z_k are both observed quantities, the descriptor system can be interpreted as a single *implicit process* model:

$$\begin{bmatrix} \bar{E} \\ \bar{C} \end{bmatrix} x_k = \begin{bmatrix} \bar{A} \\ 0 \end{bmatrix} x_{k-1} + \begin{bmatrix} \bar{B} & 0 \\ 0 & I \end{bmatrix} \begin{pmatrix} u_k \\ z_k \end{pmatrix} + \begin{pmatrix} w_k \\ -n_k \end{pmatrix}. \tag{5}$$

The filter proposed in this letter employs a residual interpretation of this *implicit process* model and is closely related to the descriptor Kalman filter introduced by Nikoukhah *et al.* [13]. In their work, they derive a so-called "3-block" form for descriptor systems with arbitrary $\bar{E}$ matrix and can thus handle under- or over-constrained problems. They also investigate detectability and discuss the issue related to non-causal behavior that can arise in descriptor systems.

Batch optimization [14], [15] also allows for more general *process* models and may improve the handling of non-linear and non-Gaussian settings. However, recursive filters have their own set of advantages which are mainly related to their real-time nature and a lower implementation complexity. This also explains why, nowadays, they are still widely used in mobile robotics. Also, for well-behaved problems (moderate non-linearities and good state observability) the difference in accuracy between filters and batch optimization becomes negligible.

In contrast to the descriptor Kalman filter [13], the proposed approach emphasizes a residual based perspective of the *implicit process* model (5) and takes a further step into the direction of batch optimization. The employed residuals depend on two consecutive states and we consequently entitle the proposed filter "Two-State Implicit Filter" (TSIF). A probabilistic derivation of the maximum-likelihood problem is described and a set of recursive filtering equations is presented. Furthermore, we exploit the link to batch optimization in order to discuss iterative schemes as well as a robust weighting method for a better handling of non-linear systems and outliers.

In comparison to classical Bayesian filtering such as the EKF, the residual based modeling allows a more flexible integration of sensor data and may thereby improve modeling accuracy. The proposed approach is also not equivalent to a Kalman filter augmented with the previous state which does not allow to circumvent an *explicit process* model. In order to demonstrate the relevance for mobile robotics the proposed technique is first illustrated on a toy example. Then, we solve two different real state estimation problems: Two novel filters for fusing inertial measurement unit (IMU) data with model based dynamic/kinematic cues are implemented and evaluated – one for micro aerial vehicles (MAVs) and one for legged robots.

II. PREREQUISITES

A multivariate Gaussian random variable with mean μ and covariance P has a probability density function of the following form:

$$p(x) = \frac{1}{\sqrt{2\pi \det(P)}} \exp\left(\frac{-(x-\mu)^T P^{-1}(x-\mu)}{2}\right). \tag{6}$$

The corresponding negative log-likelihood yields:

$$-\log(p(x)) = \log\left(\sqrt{2\pi \det(P)}\right) + \frac{1}{2}\|x-\mu\|_P^2, \tag{7}$$

where $\|\cdot\|_P$ is the Mahalanobis distance with covariance P. In the context of recursive estimation, we will denote the estimated mean and covariance of the state at timestep k considering the information until timestep l by x_k^l and P_k^l, respectively. We also employ the abbreviation $P_k = P_k^k$.

Two useful properties can be derived from the above negative log-likelihood. First, the mean μ is equivalent to the minimum of the negative log-likelihood, and second, the covariance P can be retrieved by inverting the Hessian of the negative log-likelihood:

$$\mu = \arg\min_x \left(-\log(p(x))\right). \tag{8}$$

$$P^{-1} = -\frac{\partial^2}{\partial x^2}\log(p(x)). \tag{9}$$

III. TWO-STATE IMPLICIT FILTER

The keywords contained in the term "two-state implicit filter" refer to the way measurements (or constraints) are modeled in the proposed filter. In order to circumvent the classical prediction-update paradigm of Kalman filters, an *implicit* residual modeling scheme is employed which is related to the *implicit process* model in descriptor systems [11]. Measurements or constraints are always associated with a residual depending on a consecutive pair of states (hence *two-state*), even if the derived residuals depend on a single state only. This improves flexibility and allows covering a broader set of estimation problems. For
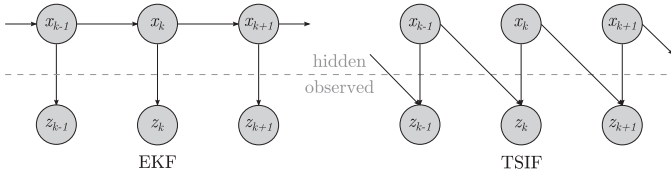
Fig. 2. Bayes network for the classical EKF and the proposed TSIF. In comparison to the EKF, the TSIF does not assume any process model but rather extends the dependency of the observed quantities to two hidden states, i.e., the measurement z_k depends on the states x_{k-1} and x_k.

instance auxiliary states can be introduced without necessarily having to provide corresponding dynamics (e.g. an unknown velocity dimension). Or various process models with overlapping states (e.g. multiple odometry models) can be merged without the need for resolving the overlap.

A. Probabilistic View

We begin with a probabilistic description of the approach. While still meeting a constraint related to the Markov property (see (11)), the employed discrete stochastic model deviates from the classical hidden Markov model by avoiding the need for an *explicit process* model. In other terms, the states of the system at different timesteps are assumed to be independent if no further information is provided (i.e. not considering any measurements or constraints). To compensate for this assumption, we employ a measurement model which depends on two consecutive states: Instead of employing a probability density function of the form $p(z_k|x_k)$, where x_k is the (hidden) state at time t_k and z_k the measurements at the same time instant, we make use of

$$p(z_k|x_{k-1}, x_k), \tag{10}$$

which represents a conditional density function on both the previous and the current state. An illustration of the underlying Bayesian network is given in Fig. 2, and compared to the hidden Markov model employed in classical Bayesian filters. We require that the system fulfills the following property:

$$p(z_k|x_0, \dots, x_k) = p(z_k|x_{k-1}, x_k). \tag{11}$$

This is similar to the well-known Markov property but exhibits a "double support" for the conditional probability and is expressed w.r.t. the measurements. Please note that this can also be inverted in time which can be recognized when contemplating the symmetry w.r.t. time of the system in Fig. 2, i.e.:

$$p(z_k|x_{k-1}, \dots, x_N) = p(z_k|x_{k-1}, x_k). \tag{12}$$

In the context of filtering, the goal is to find the probability distribution of the current state x_k given all available information up to the time t_k, i.e., we want to find $p(x_k|z_1, \dots, z_k)$, which we will also write as $p(x_k|Z_k)$. To this end, we assume to know the conditional distribution of the current measurement z_k (10) as well as the distribution of the previous state $p(x_{k-1}|Z_{k-1})$. We begin by investigating the joint distribution of x_{k-1} and x_k given the available measurements Z_k. Using Bayes' rule we get:

$$p(x_{k-1}, x_k|Z_k) = \frac{p(Z_k|x_{k-1}, x_k)p(x_{k-1}, x_k)}{p(Z_k)}. \tag{13}$$

Similar to other Bayesian filtering approaches we assume conditional independence of the measurements (given the separating states). In our setup we also have the independence of the states and can thus factorize the above term into:

$$p(x_{k-1}, x_k|Z_k)$$
$$= \frac{p(Z_{k-1}|x_{k-1}, x_k)p(z_k|x_{k-1}, x_k)p(x_{k-1})p(x_k)}{p(Z_k)}. \tag{14}$$

Using the property $p(Z_{k-1}|x_{k-1}, x_k) = p(Z_{k-1}|x_{k-1})$ (derived from (12)) and reapplying Bayes' rule we get:

$$p(x_{k-1}, x_k|Z_k)$$
$$= \underbrace{\frac{p(Z_{k-1})}{p(Z_k)}}_{\text{normalization}} \underbrace{p(x_k)}_{\text{prior}} \underbrace{p(x_{k-1}|Z_{k-1})}_{\text{previous step}} \underbrace{p(z_k|x_{k-1}, x_k)}_{\text{measurement model}}. \tag{15}$$

We drop the prior as it can be integrated in the measurement model and marginalize the previous state x_{k-1} to get a recursive filtering framework:

$$p(x_k|Z_k) \sim \int p(x_{k-1}|Z_{k-1})p(z_k|x_{k-1}, x_k)dx_{k-1}. \tag{16}$$

In the case where a process model $p(x_k|x_{k-1})$ (or a prior $p(x_k)$) is available we can leverage the contained information by introducing a virtual measurement $\tilde{z}_k$ and associating it with the following measurement model:

$$p(\tilde{z}_k|x_{k-1}, x_k) \sim p(x_k|x_{k-1}). \tag{17}$$

If we employ this within (16) we obtain

$$p(x_k|Z_{k-1}, \tilde{z}_k) \sim \int p(x_k|x_{k-1})p(x_{k-1}|Z_{k-1})dx_{k-1}, \tag{18}$$

which is equivalent to what is obtained from the prediction step of a Bayesian filter with a regular hidden Markov model. More notes on this matter are provided in the next section.

B. Gaussian Case and Negative Log-Likelihood

So far we have not made many assumptions on the type of involved distributions. However, for computational reasons and in analogy to many other stochastic inference techniques, we assume multivariate Gaussian distributions. Since all employed operations (e.g. multiplication of distributions, marginalization) preserve Gaussian distributions, it is sufficient if the prior $p(x_0)$ as well as the employed measurement model $p(z_k|x_{k-1}, x_k)$ are Gaussians. As important design choice we make use of an *implicit* measurement model

$$p(z_k|x_{k-1}, x_k) \sim \exp\left(-\frac{1}{2}\|r(x_{k-1}, x_k, z_k)\|^2\right), \tag{19}$$

using the Euclidean norm $\|\cdot\|$ with the linear residual

$$r(x_{k-1}, x_k, z_k) = a + Ax_{k-1} + Bx_k + Cz_k. \tag{20}$$

This form allows a more general and flexible encoding of the available information and may go beyond mere sensor based measurements. For instance, it may be used to represent other forms of information such as stationarity assumptions or priors

which are not directly linked to any physical sensor. Please note that the induced conditional probability density function on z_k may be non-integrable (similar to an improper prior). However, this does not represent an issue for the proposed filter as z_k is an observed quantity and does not need to be estimated. For the aforementioned case where a process model of the form

$$p(x_k|x_{k-1}) \sim \exp\left(-\frac{1}{2}\|x_k - Fx_{k-1}\|_W^2\right) \quad (21)$$

is available, a virtual measurement can be introduced as:

$$r(x_{k-1}, x_k, z_k) = W^{-1/2}(x_x - Fx_{k-1}). \quad (22)$$

Because this residual is independent of z_k, this induces a non-integrable Gaussian but nonetheless allows a neat inclusion of the process model. This concept is also employed in batch optimization based approaches [14], [15].

The relations in Section II motivate the use of the negative log-likelihood for computing mean and covariance of the joint distribution $p(x_{k-1}, x_k|Z_k)$ before applying a marginalization step in order to obtain $p(x_k|Z_k)$. To this end, we contemplate the involved terms in (15). The first term is constant and can be ignored. The second term can be modeled as part of the last term and is therefore also dropped. The third term represents the solution of the previous filter step and thus has a Gaussian distribution with mean x_{k-1}^{k-1} and covariance P_{k-1}. The last term represents our measurement model (19) where we embed the measurement z_k into the term $b_k = a + Cz_k$ since it is not part of the subsequent minimization. We thus obtain the following negative log-likelihood (ignoring the additive constant):

$$L(x_{k-1}^k, x_k^k) = \underbrace{\|x_{k-1}^k - x_{k-1}^{k-1}\|_{P_{k-1}}^2}_{\text{pervious step}}$$

$$+ \underbrace{\|b_k + Ax_{k-1}^k + Bx_k^k\|^2}_{\text{residuals}}, \quad (23)$$

where the state is n dimensional and the residual is m dimensional.

Please note that a similar derivation scheme can be used to derive the Kalman filter equations, whereas the negative log-likelihood has the following form:

$$L(x_{k-1}^k, x_k^k) = \underbrace{\|x_{k-1}^k - x_{k-1}^{k-1}\|_{P_{k-1}}^2}_{\text{pervious step}} + \underbrace{\|Fx_{k-1}^k - x_k^k\|_Q^2}_{\text{prediction}}$$

$$+ \underbrace{\|Hx_k^k - z_k\|_R^2}_{\text{update}}. \quad (24)$$

By choosing an appropriate residual vector b_k and matrices A and B, the Kalman filter case can be shown to be covered by the proposed approach.

C. Filter Equations

Based on the previously derived negative log-likelihood the goal is to derive a set of recursive filter equations. The posterior of the state is associated with the minimum of the negative log-likelihood and consequently we differentiate (23) w.r.t. x_k^k and x_{k-1}^k and equate it to zero. This yields the following system of equations:

$$\underbrace{\begin{bmatrix} P_{k-1}^{-1} + A^T A & A^T B \\ B^T A & B^T B \end{bmatrix}}_{\text{Hessian}} \begin{pmatrix} x_{k-1}^k \\ x_k^k \end{pmatrix} = \begin{pmatrix} P_{k-1}^{-1} x_{k-1}^{k-1} - A^T b_k \\ -B^T b_k \end{pmatrix}.$$

$$(25)$$

The posterior covariance of the joint distribution can be retrieved by inverting the Hessian of the negative log-likelihood (see (9)). However, the present Hessian is associated with the *joint* posterior over the previous and current state and thus we have to marginalize the previous step in order to retrieve the covariance of the current state. This can be achieved by employing Schur's complement:

$$P_k^{-1} = B^T B - B^T A (P_{k-1}^{-1} + A^T A)^{-1} A^T B. \quad (26)$$

Similarly, this can also be used to find the mean of the posterior of the current state:

$$P_k^{-1} x_k^k$$

$$= B^T \left(A(P_{k-1}^{-1} + A^T A)^{-1}\left(A^T b_k - P_{k-1}^{-1} x_{k-1}^{k-1}\right) - b_k\right)$$

$$= -B^T(I - A(P_{k-1}^{-1} + A^T A)^{-1} A^T)(A\, x_{k-1}^{k-1} + b_k). \quad (27)$$

By introducing some intermediary terms the following recursive filter equations can be derived:

$$D_k = Y_{k-1} + A^T A, \quad (28)$$

$$S_k = B^T(I - AD_k^{-1}A^T), \quad (29)$$

$$Y_k = S_k B, \quad (30)$$

$$Y_k x_k^k = -S_k(A\, x_{k-1}^{k-1} + b_k), \quad (31)$$

where $Y_k = P_k^{-1}$ is the information matrix. Other forms of recursive filter equations may be derived but are not in the scope of this letter. Overall, the proposed information form is relatively efficient and is comparable to the classical information filter [5]. It scales linearly with the number of residuals. Also, due to the usually sparse Jacobians, the costs are dictated by the matrix inversion which can (theoretically) be solved in $\mathcal{O}(n^{2.4})$. The computational costs of the algorithm depend on the selected order of the matrix multiplication and on the employed decompositions/solvers.

Note that the matrix B needs to have full column rank for the equations to be solvable. A detailed discussion of the stability and convergence of the filter is not in the scope of this letter and interested readers are referred to the related detectability and stabilizability discussion in [13].

D. Non-Linear Case

For non-linear residuals $r(x_{k-1}, x_k, z_k)$, and consequently non-Gaussian measurement models, a simple linearization of the residual provides us with a Gaussian approximation. This is in analogy to the linearization step employed by the EKF. However, we may encounter non-vector space quantities (e.g. rotations) and thus the use of boxplus $\boxplus$ (relates to addition) and boxminus $\boxminus$ (relates to subtraction) operators for introducing a local perturbation $\delta x_k^i = x_k^i \boxminus \bar{x}_k^i$ around a linearization point

$\bar{x}_k^i$ becomes indispensable [16], [17]. The residual itself is assumed to be member of a vector space in order to guarantee a meaningful metric for the associated density function (19). By carrying out a change of variable and linearizing by means of Jacobians ($J_\boxminus$, $J_{x_{k-1}^k}$, $J_{x_k^k}$) we get:

$$
\begin{aligned}
L(\delta x_{k-1}^k, \delta x_k^k) = \| \bar{x}_{k-1}^k \boxminus x_{k-1}^{k-1} \\
+ \underbrace{J_\boxminus(\bar{x}_{k-1}^k, x_{k-1}^{k-1})}_{J_k} \delta x_{k-1}^k \|_{P_{k-1}}^2 + \| \underbrace{r(\bar{x}_{k-1}^k, \bar{x}_k^k, z_k)}_{b_k} \\
+ \underbrace{J_{x_{k-1}^k}(\bar{x}_{k-1}^k, \bar{x}_k^k, z_k)}_{A_k} \delta x_{k-1}^k + \underbrace{J_{x_k^k}(\bar{x}_{k-1}^k, \bar{x}_k^k, z_k)}_{B_k} \delta x_k^k \|^2.
\end{aligned}
$$
$$(32)$$

This results in the following adapted filter equations:

$$
D_k = J_k^T Y_{k-1} J_k + A_k^T A_k, \tag{33}
$$

$$
S_k = B_k^T (I - A_k D_k^{-1} A_k^T), \tag{34}
$$

$$
Y_k = S_k B_k, \tag{35}
$$

$$
Y_k \delta x_k^k = S_k (A J_k^{-1} (\bar{x}_{k-1}^k \boxminus x_{k-1}^{k-1}) - b_k), \tag{36}
$$

$$
x_k^k = \bar{x}_k^k \boxplus \delta x_k^k. \tag{37}
$$

In summary, when compared to the linear case, we have to compute the Jacobians and work with local perturbation in order to handle non-vector space quantities.

One important aspect is the selection of linearization points. For the linearization point of the previous state $\bar{x}_{k-1}^k$ the prior x_{k-1}^{k-1} is often a good choice. This also simplifies the filter equations since the matrix J_k reduces to the identity matrix and $\bar{x}_{k-1}^k \boxminus x_{k-1}^{k-1} = 0$. For the current state, finding a suitable linearization point $\bar{x}_k^k$ may be more involved. For cases with small incremental changes the prior x_{k-1}^{k-1} could be employed again. In other cases, scenario specific prediction models may be applied. Optionally, the chosen linearization points can also be refined iteratively, where a vanishing update $\delta x_k^k < \epsilon$ could serve as termination condition [3]. For the presented approach there are two options: Either the iteration is only performed on the current state x_k^k (*partial* form) or both, the previous state and the current state, are refined during the iterations (*full* form). In the later case, the full two-state-window batch optimization in (32) has to be solved every step. Whether an iterations scheme should be used depends on the level of nonlinearity within the range of estimation uncertainty.

One notable advantage of the present formulation is that robust cost functions can be embedded into the residuals. This allows to increase the robustness w.r.t. to bad data or constraint violations. In comparison to classical Kalman filters, this can easily be applied on *every* residual, notably also on prediction-related residuals. This means that robustness against outliers can be incorporated, both, for update-related residuals as well as for prediction-related residuals. The simplest approach would be to apply a simple weighting scheme (e.g. Huber weighting).

IV. EXPERIMENTS AND RESULTS

Within the scope of this experimental section we discuss the application of the filter to 3 estimation problems:

1) The fusion of odometry and absolute location data for the toy example in Fig. 1.
2) The fusion of IMU data with barometer based height measurements as well as rotor speed measurements on a MAV with known aerodynamic model.
3) The fusion of IMU data with model-based kinematic and dynamic cues on a quadrupedal robot.

We thereby illustrate the type of models that can be employed and show the benefits of the TSIF for mobile robotics. While for the first experiment simulated data is employed, real data is used for the experiments involving the MAV and the quadruped. The presented filters, especially the one for the quadrupedal robot, bring some novelty on their own. Due to limited space, a selected subset of the results is presented.

A. Toy Example

We first investigate the applicability of the presented approach to a simple toy example: a 2D robot with global position coordinates x_k and y_k, heading θ_k, forward velocity v_k (speed), and rotational rate ω_k (see Fig. 1). Given measurements of the speed $\tilde{v}_k$ and the rotational rate $\tilde{\omega}_k$, a simple process model can be formulated:

$$
x_k = x_{k-1} + dt \cos(\theta_{k-1}) \tilde{v}_{k-1} + w_{x,k}, \tag{38}
$$

$$
y_k = y_{k-1} + dt \sin(\theta_{k-1}) \tilde{v}_{k-1} + w_{y,k}, \tag{39}
$$

$$
\theta_k = \theta_{k-1} + dt \tilde{\omega}_{k-1} + w_{\theta,k}, \tag{40}
$$

with time increments dt and discrete Gaussian noise $w_{*,k}$. This can be completed with the following measurement model (including discrete Gaussian noise $n_{*,k}$):

$$
\tilde{x}_k = x_k + n_{x,k}, \quad \tilde{y}_k = y_k + n_{y,k}, \tag{41}
$$

and leveraged into a simple EKF in order to estimate the global position and the heading of the robot.

However, if the rotational rate is not measured the problem is less straightforward to solve since the term $\tilde{\omega}_{k-1}$ is unknown in (40). Possible solutions are:

1) Make an assumption on the distribution of $\tilde{\omega}_{k-1}$ and use an EKF based on (38) to (41). The quality of this solution strongly depends on how well the unknown rotational rates fit the chosen distribution.
2) Solve analytically for the heading θ_{k-1} using (38) and (39). This approach struggles with properly integrating the noise properties of the system.
3) Formulate a large maximum likelihood problem over N subsequent states. This becomes a full-scale batch optimization and looses any filter related properties.
4) Formulate a maximum likelihood problem over two subsequent states. This is basically the solution we propose, whereas we derived a set of recursive filter equations for doing so (including marginalization of the old state).

TABLE I
TSIF FOR TOY EXAMPLE

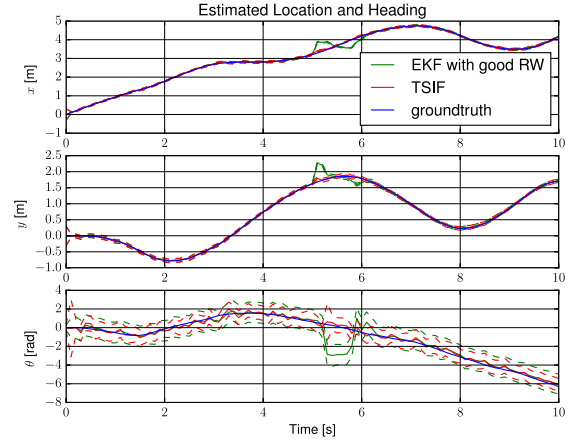| position (x_{k-1})	position (y_{k-1})	heading (θ_{k-1})	speed (v_{k-1})		position (x_k)	position (y_k)	heading (θ_k)	speed (v_k)
■				x prediction (1)	■		■	■
	■			y prediction (1)		■	■	■
				x update (1)	■			
				y update (1)		■		
				v update (1)				■

Fig. 3. Estimated location and heading for the toy example in Fig. 1 and their 3-sigma bounds. Both the TSIF and the EKF show good performance and manage to estimate the heading of the robot. However, the TSIF does so without relying on a RW prior on the heading dynamics, which is required by the EKF in order to get an explicit process model.
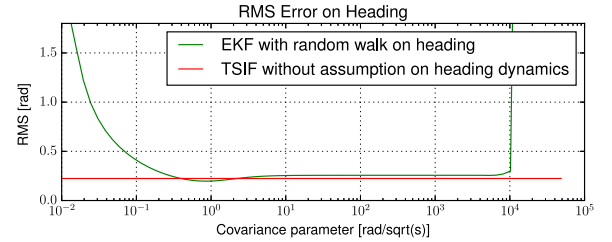

Fig. 4. RMS error of the heading estimate for the toy example in Fig. 1 for different values of covariance parameters for the heading RW. While the TSIF is agnostic to the heading RW, the EKF relies on a well tuned covariance parameter. If either the covariance is too low (over-confident) or too large (risk of numerical issues) its performance will degrade.

In the following we will compare the EKF where the heading is modeled as RW (with various noise magnitudes) against the proposed TSIF.

Table I illustrates the TSIF setup in form of a *residual-state dependency table*. The state is represented twice in the table: the previous state on the left and the current state on the right. Each row represents a single residual (with its dimension in brackets) and illustrates the sub-state dependencies: a cell is colored if the residual depends on a specific sub-state. This corresponds to the sparsity pattern of the matrices A (left) and B (right) of Section III-C. This table can be used to visualize the overall structure of the filter and obtain a qualitative impression of how the residuals and sub-states influence one-another. For the toy example, the top two rows (gray) relate to the original process equations (38) and (39), whereas a Euler-Backward discretization scheme is used in order to obtain a stronger dependency on the current state and thereby get a full column rank matrix B. The three bottom rows (green) represent the simple relation between the sub-states and their measurements. The speed v_k is included into the filter state in order to allow a proper inference from the different sources of information. The corresponding (unweighted) residual equations would be as follows:

$$r_{v_x} = x_k - dt \cos\left(\theta_k\right) v_k - x_{k-1}, \tag{42}$$

$$r_{v_y} = y_k - dt \sin\left(\theta_k\right) v_k - y_{k-1}, \tag{43}$$

$$r_x = x_k - \tilde{x}_k, \quad r_y = y_k - \tilde{y}_k, \quad r_v = v_k - \tilde{v}_k \tag{44}$$

Fig. 3 shows the obtained location and heading estimates. The TSIF is compared against the EKF with a well-tuned covariance for the heading RW. In the middle of the experiment a faulty speed measurement is injected. While both approaches yield good results and are able to estimate the not directly measured heading, the TSIF exhibits a better handling of the outlier due to a Huber weighting on speed measurements – whereas the EKF does not allow the integration of a robust kernel during prediction. Also, any change of the underlying distribution of the rotational rate could deteriorate the performance of the EKF, while the TSIF is agnostic to such a prior. To further illustrate this dependency we plotted the RMS error of the heading estimates against the covariance parameter of the heading RW in Fig. 4 in an outlier-free setting. In case this covariance is too small (overconfident) the RMS will increase for the EKF. On the other hand a very large covariance can lead to numerical issues. The TSIF does not use any prior on the heading dynamics and thus is not affected by this parameter. Whether the minimum for the EKF (around 0.9 rad$/\sqrt{s}$) achieves a lower RMS then the TSIF depends on how well the dynamics of the heading match a RW.

B. Blind MAV Filter

A common problem in MAV state estimation is the limited availability of proprioceptive sensors. In this blind MAV example we combine the data from on-board IMU and pressure sensors with the rotor speed measurements by using an aerodynamic model. The employed MAV model is based on a point mass model and a simple aerodynamic model which takes into account the generated thrust and drag forces, and the torque of the rotors. Both, the IMU measurements and the aerodynamic model relate to the accelerations of the robot body and thus represent two different *process* models which cannot easily be incorporated in a traditional filter. Burri *et al.* [18] did so within an EKF, but had to extend the filter state with rotor speeds in order to integrate the aerodynamic cues and accelerometer biases simultaneously. In comparison, the flexibility of the TSIF allows a more modular integration of both information sources.

The *residual-state dependency* is given in Table II, where "Pose FD" is the finite difference (FD) of position and attitude, "IMU" refers to the IMU measurements, and "IMU bias RW"

TABLE II
BLIND MAV FILTER

Residual	position	attitude	velocity	rot. rate	acc. bias	gyr bias	Height ref.	position	attitude	velocity	rot. rate	acc. bias	gyr bias	Height ref.
Pose FD (6)	■	■						■	■					
IMU (6)		■							■	■	■	■	■	
IMU bias RW (6)					■	■						■	■	
Barometer (1)							■	■						■
Height ref. RW (1)							■							■
Dynamics FD (6)		■	■	■					■	■				

TABLE III
BLIND QUADRUPED FILTER

Residual	position	attitude	velocity	rot. rate	acc. bias	gyr bias	footholds	dyn. par.	position	attitude	velocity	rot. rate	acc. bias	gyr bias	footholds	dyn. par.
Pose FD (6)	■	■							■	■						
IMU (6)		■								■	■	■	■	■		
IMU bias RW (6)					■	■							■	■		
Kinematics (12)															■	
Foothold FD (0-12)	■	■		■											■	
Dynamics FD (6)	■	■									■	■				■
Dyn. par. RW (4)								■								■

Estimated Attitude

(roll [rad], pitch [rad], yaw [rad] vs Time [s]; curves: Blind MAV, Groundtruth)
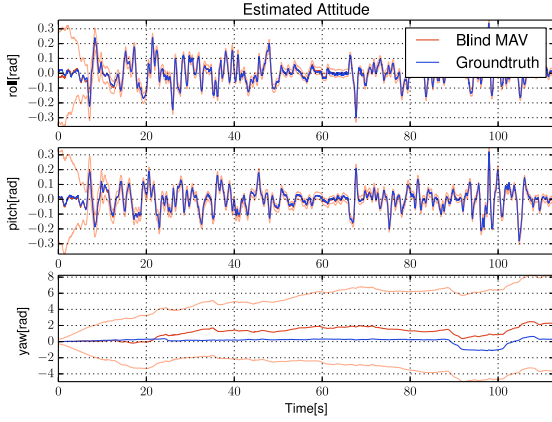
Fig. 5. Attitude estimates of the blind MAV filter and their 3-sigma bounds. Relying on IMU, barometer, and rotor speed measurements only, the filter is able to estimate the inclination angles very accurately and in a consistent manner (the oscillations are part of the robot motion). The yaw angle is unobservable and drifts.
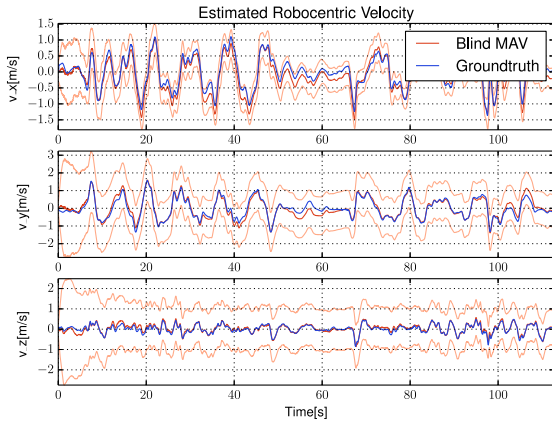
Estimated Robocentric Velocity

(v_x [m/s], v_y [m/s], v_z [m/s] vs Time [s]; curves: Blind MAV, Groundtruth)

Fig. 6. Velocity estimates of the blind MAV filter and their 3-sigma bounds. The filter successfully estimates the robocentric velocity of the MAV and could for instance be applied to bridge outages of exteroceptive sensing capabilities.

represents the RW on accelerometer and gyroscope biases. The "Dynamics FD" residual relies on the aerodynamic model to compute linear and rotational accelerations and relates them to the FD of the velocities. The inclusion of the barometer measurements requires the augmentation of the state with a reference height in order to be able to handle reference changes (modeled as RW).

Figs. 5 and 6 show the estimated attitude and linear velocities, which are both crucial quantities in MAV control. Please note that no exteroceptive sensor modality is employed and thus the position and yaw angle are intrinsically unobservable. However, as can be observed at hand of the bound uncertainty in both plots, the inclination angles and robocentric velocity are observable and track the groundtruth. This would not be achievable in an IMU-only setup without further assumption. The dynamic model is subject to large uncertainties and affected by noisy motor speeds, and thus the weighting of the corresponding residual is kept low. Still, it provides enough information to estimate inclination angles and velocities and could represent an interesting back-up solution for autonomous MAVs. The accumulated drift over 2 min amounts to 15 m.

C. Blind Quadruped Filter

In our final example we derive a filter for fusing IMU data with kinematic and dynamic cues on a torque controlled and fully-actuated quadrupedal robot [19]. In comparison to established approaches [20]–[22], the inclusion of dynamics increases the redundancy and may allow the co-estimation of certain model parameters (e.g. the mass). The IMU is integrated in the same way as in the Blind MAV Filter (Section IV-B). The kinematic measurements from the joint encoders are leveraged as relative measurements between foot contact points and main body. This can be achieved by augmenting the filter state with robocentric foot locations [21].

Instead of an aerodynamic model, we leverage the rigid body equations of motion. Dynamic quantities have been applied to legged robot state estimation previously [23], but, to the best of our knowledge, they have never been tightly fused with inertial data in a consistent manner. While the joint torques are measured, the foot-ground contact forces are unknown and are eliminated by means of a null-space projection [24]. The remaining set of equations is sufficient to predict the linear and rotational acceleration of the main body and is thus formulated as a dynamic FD residual. Additionally, the mass of the main body and its offset are included into the state. Both represent important quantities for control and may be subject to change when the payload is altered. The *residual-state dependency* is given in Table III.

Figs. 7 and 8 depict the position of the robot and its linear velocity. For moving around the rectangular shape the quadrupedal robot employed a trotting gait (with a significant amount of two-stance phases). In relation to the long trajectory (~ 75 m) the position estimates exhibit only little drift (mainly due to the drifting yaw angle). Tests have also been carried out without IMU data (again demonstrating the modularity of the approach),
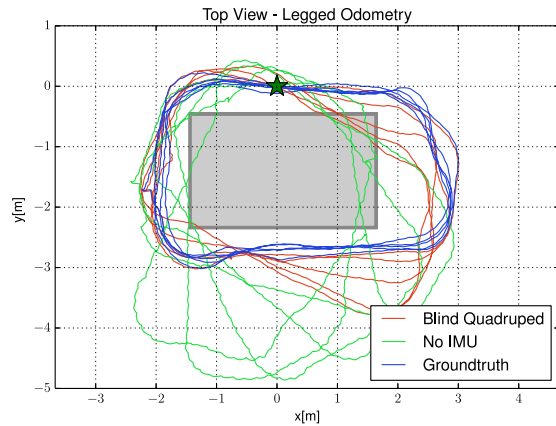
Fig. 7. Position estimates of the blind quadruped filter while the robot is trotting around a rectangular shape. For a dead-reckoning approach the achieved accuracy is good. If the IMU data is neglected the drift becomes more significant, especially around the yaw angle.
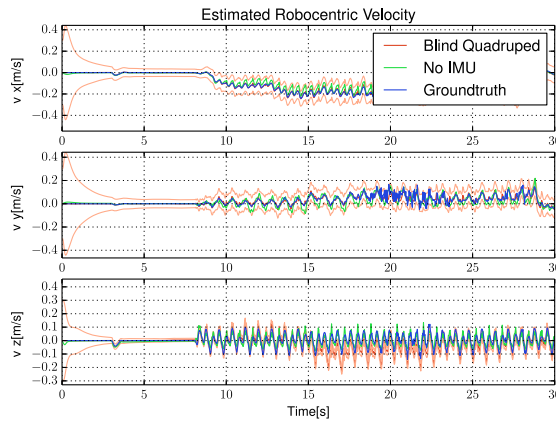

Fig. 8. Velocity estimates of the blind quadruped filter. Except for the velocity along the z-axis which exhibits a slight offset, the estimates track the motion capture derived references. When disabling the IMU the estimate get more noisy but still manage to track the reference. The 3-sigma bounds of the IMU-free case are very large and not depicted in the above plots.

where the state estimation only relied on model-based kinematics and dynamics. Even though the dynamic cues are affected by significant noise and subject to high modeling inaccuracies, the obtained estimates are still reasonable and could be used to handle IMU failures. The velocity estimates of both filters also show an accurate and consistent tracking. The mass was in both cases estimated to be around 18.2 kg (a coarse reference around 20 kg) and its center near the middle of the robot.

V. CONCLUSION

We presented a novel recursive estimation algorithm which can be seen as generalization of classical Kalman filtering in that it handles a broader range of system models. It also takes a step into the direction of batch optimization by modeling all available information as collection of residuals. This makes the filter implementation more flexible and modular and allows the use of further techniques such as robust weighting. A detailed derivation is provided for the corresponding set of recursive filter equations, which is then applied in three different mobile robotic sensor fusion problems.

The authors are currently working on an open-source filtering library which should provide researchers with a plug-n-play filter library. Consequently, future work will include implementation related topics such as automatic measurement handling, numerical stability, observability constraints, or square root forms.

REFERENCES

[1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Fluids Eng.*, vol. 82, no. 1, pp. 35–45, 1960.
[2] H. W. Sorenson, *Kalman filtering: Theory and Application.* Piscataway, NJ, USA: IEEE Press, 1985.
[3] B. Bell and F. Cathey, "The iterated Kalman filter update as a Gauss-Newton method," *IEEE Trans. Autom. Control*, vol. 38, no. 2, pp. 294–297, Feb. 1993.
[4] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *Proc. Int. Soc. Opt. Photon. AeroSense*, 1997, pp. 182–193.
[5] P. S. Maybeck, *Stochastic Models, Estimation, and Control*, vol. 1. New York, NY, USA: Academic, 1979.
[6] P. Kaminski, A. Bryson, and S. Schmidt, "Discrete square root filtering: A survey of current techniques," *IEEE Trans. Autom. Control*, vol. AC-16, no. 6, pp. 727–736, Dec. 1971.
[7] Y. Bar-Shalom and L. Campo, "The effect of the common process noise on the two-sensor fused-track covariance," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-22, no. 6, pp. 803–805, Nov. 1986.
[8] K. Chang, "On optimal track-to-track fusion," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 33, no. 4, pp. 1271–1276, Oct. 1997.
[9] T. Yuan, Y. Bar-shalom, and X. Tian, "Heterogeneous track-to-track fusion," in *Proc. Int. Conf. Inf. Fusion*, 2011, pp. 1–8.
[10] H. Chen, T. Kirubarajan, and Y. Bar-Shalom, "Performance limits of track-to-track fusion vs. centralized Estimation: Theory and application," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 2, pp. 386–400, Apr. 2003.
[11] S. L. V. Campbell, *Singular Systems of Differential Equations*. New York, NY, USA: Pitman, 1980.
[12] M. Skliar and W. F. Ramirez, "Implicit Kalman filtering," *Int. J. Control*, vol. 66, no. 3, pp. 393–412, 1997.
[13] R. Nikoukhah, A. Willsky, and B. C. Levy, "Kalman filtering and Riccati equations for descriptor systems," *IEEE Trans. Autom. Control*, vol. 37, no. 9, pp. 1325–1342, Sep. 1992.
[14] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2o: A general framework for graph optimization," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 3607–3613.
[15] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *Int. J. Robot. Res.*, vol. 31, no. 2, pp., pp. 3281–3288, 2012.
[16] C. Hertzberg, R. Wagner, U. Frese, and L. Schröder, "Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds," *Inf. Fusion*, vol. 14, no. 1, pp. 57–77, 2011.
[17] M. Bloesch et al., "A primer on the differential calculus of 3D orientations," *CoRR*, vol. abs/1606.0, 2016.
[18] M. Burri, M. Dätwiler, M. W. Achtelik, and R. Siegwart, "Robust state estimation for micro aerial vehicles based on system dynamics," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 5278–5283.
[19] M. Hutter et al., "ANYmal—A highly mobile and dynamic quadrupedal robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 38–44.
[20] A. Chilian, H. Hirschmüller, and M. Görner, "Multisensor data fusion for robust pose estimation of a six-legged walking robot," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2011, pp. 2497–2504.
[21] M. Bloesch et al., "State estimation for legged Robots—Consistent fusion of leg kinematics and IMU," in *Proc. Robot. Sci. Syst. Conf.*, 2012, pp. 1–8.
[22] M. F. Fallon, M. Antone, N. Roy, and S. Teller, "Drift-free humanoid state estimation fusing kinematic, inertial and LIDAR sensing," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2014, pp. 112–119.
[23] X. Xinjilefu, S. Feng, and C. G. Atkeson, "Dynamic state estimation using quadratic programming," in *Proc. 2014 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 989–994.
[24] M. Mistry, J. Buchli, and S. Schaal, "Inverse dynamics control of floating base systems using orthogonal decomposition," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 3406–3412.