MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Evaluation of the Discrete Time Feedback Particle Filter for IMU-Driven Systems Configured on SE2

Greiff, M.; Berntorp, K.

TR2018-087 July 13, 2018

Abstract

This paper evaluates the utility of the feedback particle filter (FPF) for state estimation of SE(2)-configured dynamics in a real-time context. The filter is implemented in discrete time to fuse gyroscopic- and accelerometer measurements with Ultra-Wideband (UWB) and camera measurements. With this state information, the FPF is compared to other common filters in terms of the estimate mean square error (MSE) and robustness to initial conditions. An analysis is done on how these metrics scale with utilization of computational resources, concluding that the FPF should be considered for embedded applications with CPUs on par with the Cortex M4 processor.

American Control Conference (ACC)

Evaluation of the Discrete Time Feedback Particle Filter for IMU-Driven Systems Configured on $SE(2)^*$

Marcus Greiff[1] and Karl Berntorp[2]

***Abstract*— This paper evaluates the utility of the feedback particle filter (FPF) for state estimation of $SE(2)$-configured dynamics in a real-time context. The filter is implemented in discrete time to fuse gyroscopic- and accelerometer measurements with Ultra-Wideband (UWB) and camera measurements. With this state information, the FPF is compared to other common filters in terms of the estimate mean square error (MSE) and robustness to initial conditions. An analysis is done on how these metrics scale with utilization of computational resources, concluding that the FPF should be considered for embedded applications with CPUs on par with the Cortex M4 processor.**

I. INTRODUCTION

In this paper, the feedback particle filter (FPF) is analyzed in a real-time context for $SE(2)$-configured dynamics with measurements from an inertial measurement unit (IMU). The FPF was first developed for simpler continuous-discrete filtering problems in [1], [2], where it was shown to outperform the bootstrap particle filter (PF). It has since been the subject of various simulation studies [3]–[5], compared to methods such as the extended Kalman filter (EKF) [6], [7], unscented Kalman filter (UKF) [8], and marginalized particle filter (RBPF) [9]–[11]. With encouraging results, the FPF was further investigated in the context of matrix Lie groups and used for continuous time attitude estimation of $SE(3)$ dynamics [12]. However, by the computational nature of particle filters and their poor scaling with state-space dimensionality, the FPF is rarely considered for real-time implementations in general rigid-body robotics.

Consequently, it is of great interest to study (i) how the FPF complexity scales with state-space and measurement equation dimensionality, and particularly how it compares to the above mentioned estimators in terms of (ii) accuracy and (iii) robustness when subjected to computational constraints entailed by a real-time implementation. The appeal of studying the IMU-driven system on $SE(2)$ lies its low state-space dimensionality, relative independence of parameters and subsequent generality. As such, the presented results apply to a wide class of systems, including hovercraft vehicles (HV), surface vessels (SV) and ground vehicles (GV).

The dynamical GV model and considered measurement equations are presented in Section II, before giving a review of the above mentioned filters and presenting the equations of the discrete-time FPF in Section III. Our main contribution is then to answer the questions (i)-(iii) in Section IV.

To enable this analysis, we consider each algorithm as a set of components $\mathcal{A} = \{\mathcal{A}_i\}$, with a corresponding O_i operations on average taking $\bar{t}_i$ seconds to execute. Letting f_i Hz denote the rates at which these components are run, a metric of utilization and total complexity is defined as

$$\mathcal{U} = \sum_i^{|\mathcal{A}|} \bar{t}_i f_i \in [0, 1], \qquad \mathcal{C} = \sum_i^{|\mathcal{A}|} O_i f_i > 0. \tag{1}$$

The utilization metric allows for fair comparison of the filters on specific processors, while the complexity metric indicates how when filter may implemented on other CPUs. Similarly to [2], [5], performance in estimating a sequence of $M+1$ states $\{\hat{\mathbf{x}}\}_{i=0}^{M}$ from a set of underlying true states $\{\mathbf{x}\}_{i=0}^{M}$, is done in a mean-square error (MSE) metric,

$$MSE = \frac{1}{M+1} \sum_{i=0}^{M} (\mathbf{x}_i - \hat{\mathbf{x}}_i)^T (\mathbf{x}_i - \hat{\mathbf{x}}_i). \tag{2}$$

II. MODELS

In this section, a continuous time IMU-driven $SE(2)$ model is presented with its discrete time equivalent. We also give the measurement models for linear camera-based measurements and nonlinear radio-based UWB measurements.

A. Torque-Controlled SE(2) Configured GV

Consider a differentially torque-driven two-wheeled GV, defined by an axis length $2h$ m, a wheel radius r m, a moment of inertia J kg·m^2, and a mass m kg. The system is controlled by two torques $\tau_1(t), \tau_2(t) \in \mathbb{R}$ generated by two wheels. The rigid-body is configured on $\mathbf{g}(\mathbf{p}, \theta) \in SE(2)$, where $\mathbf{p} = [p_{\mathcal{G}}^x, p_{\mathcal{G}}^y]^T$ defines the position of the body frame origin $\{\mathcal{B}\}$ in the global frame $\{\mathcal{G}\}$, and the rotation $\mathbf{R}\{\theta\} \in SO(2)$ describes the rotation of $\{\mathcal{B}\}$ relative to the $\{\mathcal{G}\}$ [13] (see Figure 1). The corresponding one-parametric group, $\boldsymbol{\xi} = [\omega_{\mathcal{B}}, v_{\mathcal{B}}^x, v_{\mathcal{B}}^y]^T \in \mathbb{R}^3$, is found as the body velocities, uniquely defining the Lie algebra $\mathfrak{se}(2)$. The governing equations are

$$\dot{\theta}(t) = \omega_{\mathcal{B}}(t) \tag{3a}$$
$$\dot{p}_{\mathcal{G}}^x(t) = v_{\mathcal{B}}^x(t)\cos(\theta(t)) - v_{\mathcal{B}}^y(t)\sin(\theta(t)) \tag{3b}$$
$$\dot{p}_{\mathcal{G}}^y(t) = v_{\mathcal{B}}^x(t)\sin(\theta(t)) + v_{\mathcal{B}}^y(t)\cos(\theta(t)) \tag{3c}$$
$$\dot{\omega}(t) = (h/(Jr))(\tau_1(t) - \tau_2(t)) \tag{3d}$$
$$\dot{v}_{\mathcal{B}}^x(t) = \omega(t) v_{\mathcal{B}}^y(t) + (r/m)(\tau_1(t) + \tau_2(t)) \tag{3e}$$
$$\dot{v}_{\mathcal{B}}^y(t) = -\omega(t) v_{\mathcal{B}}^x(t) \tag{3f}$$

*This work has received funding from the Swedish Science Foundation (SSF) project "Semantic mapping and visual navigation for smart robots" (RIT15-0038), and has benefitted from prior work at. Mitsubishi Electric Research Labs.

[1]Marcus Greiff is with the Department of Automatic Control, Lund University, Lund, Sweden marcus.greiff@control.lth.se

[2]Karl Berntorp is with Mitsubishi Electric Research Laboratories, Cambridge, MA karl.o.berntorp@ieee.org

as derived from first principles by the Newton-Euler equations. Note that by altering the actuation, the dynamics can similarly be used to model the SV and HV systems.

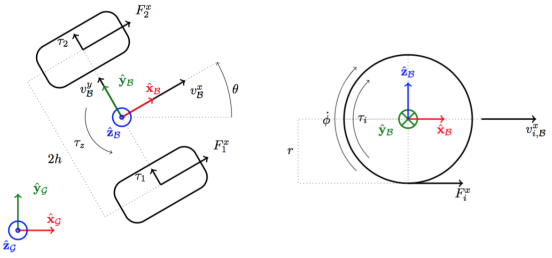

Fig. 1. *Left:* An $SE(2)$ rigid-body. *Right:* Wheel forces.

B. IMU-Driven Dynamics

Provided the system has an inertial measurement unit (IMU) measuring body frame accelerations and angular rates, $\{\dot{v}^x_{\mathcal{B}}(t), \dot{v}^y_{\mathcal{B}}(t), \dot{\theta}(t)\}$, we consider an equivalent system whose inputs are these IMU measurements. The advantage of this approach is its eliminatation of any parameter dependence in (3), making the model applicable to the GV, SV and HV systems regardless of their actuation. Clearly, if we know the true body accelerations and rotational rate, $\{\mathbf{a}_t(t), \omega_t(t)\}$, the true system evolves according to

$$\dot{\mathbf{p}}_t(t) = \mathbf{R}\{\theta_t(t)\}\mathbf{v}_t(t), \quad \dot{\mathbf{v}}_t(t) = \mathbf{a}_t(t), \quad \dot{\theta}_t(t) = \omega_t(t) \tag{4}$$

Let $\{a^x_m(t), a^y_m(t), \omega_m(t)\}$ denote the IMU measurements, corrupted by time-varying biases $\{a^x_b(t), a^y_b(t), \omega_b(t)\}$ and noise $\mathbf{w}_{mn}(t) \triangleq [a^x_{mn}(t), a^y_{mn}(t), \omega_{mn}(t)]^T$, experimentally verified as correlated Gaussian noise (see Appendix VI-B). Modeling the bias as first order Markov processes driven by Gaussian noise $\mathbf{w}_{bn}(t) \triangleq [a^x_{bn}(t), a^y_{bn}(t), \omega_{bn}(t)]^T$ with time constants T_a for the acceleration and T_ω for the angular rates, the continuous time model may be discretized using an Explicit-Euler at a time step of $t = hk$, yielding

$$\begin{aligned}
\mathbf{p}_t(k+1) =& \mathbf{p}_t(k) + h\mathbf{R}\{\theta_t(k)\}\mathbf{v}_t(k) + (h^2/2)\mathbf{R}\{\theta_t(k)\} \\
&\cdot (\mathbf{a}_m(k) - \mathbf{a}_b(k) + \mathbf{a}_{mn}(k)) && \text{(5a)} \\
\mathbf{v}_t(k+1) =& \mathbf{v}_t(k) + h(\mathbf{a}_m(k) - \mathbf{a}_b(k) + \mathbf{a}_{mn}(k)) && \text{(5b)} \\
\theta_t(k+1) =& \theta_t(k) + h(\omega_m(k) - \omega_b(k) + \omega_{mn}(k)) && \text{(5c)} \\
\mathbf{a}_b(k+1) =& \mathbf{a}_b(k) + h(T_a^{-1}\mathbf{a}_b(k) + \mathbf{a}_{bn}(k)) && \text{(5d)} \\
\omega_b(k+1) =& \omega_b(k) + h(T_\omega^{-1}\omega_b(k) + \omega_{bn}(k)) && \text{(5e)}
\end{aligned}$$

where $\mathbf{a}_X(k) = [a^x_X(k), a^y_X(k)]^T$ for $X = \{m, mn, b, bn\}$.

C. Measurement Models

In many practical applications, the position of the system is measured directly by a rigidly mounted camera system, such as the VICON, Qualisys or Optitrack products. Conservatively, we assume centimetre-range precision and let

$$\mathbf{y}_c(k) = \mathbf{h}_c(\mathbf{x}(k)) + \mathbf{e}_c(k) = \mathbf{p}_t(k) + \mathbf{e}_c(k) \in \mathbb{R}^2, \tag{6}$$

where the noise is assumed to be Gaussian and zero mean, $\mathbf{e}_c(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_c)$ with $\mathbf{R}_c = \text{diag}(0.001, 0.001)$.

If Ultra-Wideband (UWB) measurements are available, the vehicle position is determined with respect to a set of N_d anchors at positions $\mathbf{p}_i \in \mathbb{R}^3$ in space. With the common time-of-arrival approach and conventional SDS-TWR protocol [14], a vector of transit times of data-packets being sent between the UGV and the anchors is measured. This vector of time-stamps is defined as $[t_i(k)] = \mathbf{t}(k) \in \mathbb{R}^N_{>0}$. Multiplication by the speed of light, $c \approx 3 \cdot 10^8$ m/s, yields a distance between the UGV and the i^{th} anchor, $d_i(k) = ||\mathbf{p}_t(k) - \mathbf{p}_i||_2$, and the measurement equations are

$$\mathbf{y}_d(k) = \begin{bmatrix} ||\mathbf{p}_t(k) - \mathbf{p}_1||_2 \\ \vdots \\ ||\mathbf{p}_t(k) - \mathbf{p}_{N_d}||_2 \end{bmatrix} + \mathbf{e}_d(k). \tag{7}$$

The noise at each time step is assumed to be uncorrelated between each anchor and approximately gaussian, such that $\mathbf{e}_d(k) \sim \mathcal{N}(0, \mathbf{R}_d)$ where $\mathbf{R}_d = c^2\sigma_t^2\mathbf{I}$, with $\mathbf{I}$ denoting the $N_d \times N_d$ identity matrix. For the Decawave UWB chip used in the real-time implementation [15], the standard deviation of the measurement noise has been experimentally verified to be approximately $\sigma_t \approx 0.17$ ns [16].

D. Modelling summary

To summarise, the dynamical IMU-driven model is written

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) \qquad \in \mathbb{R}^n \tag{8a}$$
$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{e}_k \qquad \in \mathbb{R}^m \tag{8b}$$

where the assumptions on the process noise is given by

$$\mathbf{w}_k \triangleq \begin{bmatrix} \mathbf{w}_{mn}(k) \\ \mathbf{w}_{bn}(k) \end{bmatrix} \sim \mathcal{N}\Big(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{Q}_{nn} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{bb} \end{bmatrix} \Big) \in \mathbb{R}^6. \tag{9}$$

While $\dim(\mathbf{x}_k) = 8$ is fixed, the measurement space dimensionality, $\dim(\mathbf{y}_k) = m$, may vary, combining camera- and UWB measurements (6) (7) with $\mathbf{e}_k$ containing the corresponding measurement noise $\mathbf{e}_c(k)$ and $\mathbf{e}_d(k)$ respectively.

III. ESTIMATORS

Various filters may be considered in estimating the system states given a sequence of $\mathcal{Z}_k = \{\mathbf{u}_i, \mathbf{y}_i\}_{i=0}^k$. A common approach is to implement a recursive form of Bayes' rule,

$$p(\mathbf{x}_k|\mathcal{Z}_k) \propto p(\mathbf{x}_k|\mathbf{z}_k, \mathcal{Z}_{k-1}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathcal{Z}_{k-1})}{p(\mathbf{z}_k|\mathcal{Z}_{k-1})}, \tag{10}$$

to update a density function of the state distribution conditioned by all prior measurements and control signals. This is done comprehensively in the referenced literature [1]–[11], with a brief summary provided here to aid the reader.

A. Nonlinear Kalman Filters

By assuming unbiasedness, $\mathbb{E}[\mathbf{x}_k - \hat{\mathbf{x}}_k] = 0$, and that the posterior, $p(\mathbf{x}_k|Z_k)$, can be approximated with a Gaussian,

$$p(\mathbf{x}_k|Z_k) \approx \mathcal{N}(\mathbb{E}[\mathbf{x}_k], \mathbb{E}[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T]), \tag{11}$$

several Kalman-based filters may be derived from (10). Two standard and widely used variants are the EKF [6], [7] and the UKF proposed in [8]. The former linearizes

the nonlinear dynamics by a first order Taylor expansion, computing the posterior so as to minimize the MSE of the state estimate. The UKF instead computes a set of sigma-points, propagating the estimate with an unscented transform to directly approximate the conditional mean and covariance of the posterior distribution. Neither the EKF nor the UKF guarantee convergence and both suffer from the assumption of a gaussian state error distribution, which is violated when considering the nonlinear UWB measurements. However, the assumptions make both algorithms computationally tractable, and the filters serve as established benchmarks for the FPF implementation. For IMU-driven systems in particular, sequential scalar gain approximations can be leveraged to reduce computational complexity [17], here referred to as the scalar update extended Kalman filter (SUEKF). See [18] for explicit definitions of the associated Jacobians.

B. Particle Filters

Common particle filters (PF) differ greatly from the Kalman filters in that they are sampling based [19]. In the PF, a set of hypothesis $\{\mathbf{x}_k^i\}_{i=1}^N$ with corresponding weights $\{w_{k|k-1}^i\}_{i=1}^N$ approximate the true posterior by

$$p(\mathbf{x}_k|Z_k) \approx \hat{p}(\mathbf{x}_k|Z_k) = \sum_{i=1}^{N} w_{k|k-1}^i \mathbf{x}_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i). \quad (12)$$

The weight, w_k^i, is a measure of the likelihood of each particle, and as such the coarseness of the approximation (12) scales poorly with the state-space dimensionality $\dim(\mathbf{x}_k)$. PFs are consistent [20], with $p(\mathbf{x}_k|Z_k) = \hat{p}(\mathbf{x}_k|Z_k)$ as $N \to \infty$ provided $p(\mathbf{x}_0|Z_0) = \hat{p}(\mathbf{x}_0|Z_0)$, but performance depends greatly on the magnitude of N due to the curse of dimensionality. With $\dim(\mathbf{x}_k) = 8$ in (8), the PFs become computationally infeasible in an embedded systems context.

As a remedy, Rao-Blackwellized particle-filters (RBPF) may be considered. Assuming a mixed Gaussian state-space (MGSS) partitioning, the RBPF runs an optimal Kalman-filter for the linear states and a PF for the nonlinear states [9]–[11]. If only considering camera measurements and the IMU-driven dynamics, this partition is may be done with angle of rotation, $\theta_t(t)$, as the only nonlinear state. However, when introducing the UWB measurements, the position must also be included as a nonlinear state, see [18] for the explicit definitions. Just as the standard particle filters, the RBPF suffers from weight degeneracy, which in our implementation is handled by systematic resampling [21].

C. The Feedback Particle Filter

The final considered filter is the feedback particle filter [1]–[5], where in contrast to standard PFs, the posterior is represented with unweighted samples $w_{k|k-1}^i = N^{-1} \ \forall k$ in (12). Another fundamental difference is that the FPF implements Bayes' rule (10) by simulating the the particles as a controlled system on pseudo time, $\lambda \in [0, 1]$, each time a measurement, $\mathbf{y}_k$, is received. Letting

$$d\mathbf{x}^i = \mathbf{f}(\mathbf{x}^i(t), t)dt + d\boldsymbol{\beta} + \mathbf{U}_k^i \quad (13)$$

and assuming a Wiener distribution $\boldsymbol{\beta}$, the feedback $\mathbf{U}_k^i$ can be chosen so as to minimize the Kullback-Leibler divergence, $KL(p||\hat{p})$ at $\lambda = 1$ [4]. In practice, this is done by defining a flow of particles, $\mathbf{S}_k^i(\lambda)$, with $\mathbf{S}_k^i(0) = \mathbf{x}_{k-1}^i$. Letting

$$\frac{d\mathbf{S}_k^i(\lambda)}{d\lambda} = \mathbf{U}_k^i(\lambda), \quad (14)$$

the inclusion of measurements is reduced to an optimal control problem of synthesising a feedback law, done by solving an associated Euler-Lagrange boundary value problem. By the solution presented in [2], an innovation process is defined

$$\mathbf{I}_k^i := \mathbf{y}_k - \frac{1}{2}\Big(\mathbf{h}(\mathbf{x}_k^i) + \mathbb{E}[\mathbf{h}(\mathbf{x}_k)]\Big) \in \mathbb{R}^{m \times N}, \quad (15)$$

where

$$\mathbb{E}[\mathbf{h}(\mathbf{x})] := \int p(\mathbf{x}_k|\mathcal{Z}_k)h(\mathbf{x}_k)d\mathbf{x} \approx \frac{1}{N}\sum_{i=1}^{N} \mathbf{h}(\mathbf{x}_k^i) \quad (16)$$

due to the equal weights. The feedback law is formed as

$$\mathbf{U}_k^i(\lambda) := \mathbf{K}_k^i(\mathbf{S}_k^i, \lambda)\mathbf{I}_k^i(\mathbf{S}_k^i) + \frac{1}{2}\boldsymbol{\Omega}_k^i(\mathbf{S}_k^i, \lambda) \quad (17)$$

with $\mathbf{K}_k = [\boldsymbol{\kappa}_1 \cdots \boldsymbol{\kappa}_m]$ being a time-varying gain, and $\boldsymbol{\Omega}_k^i$ denoting the Wong-Zakai correction term [4]. The problem is then to compute the gains $\boldsymbol{\kappa}_i$, which in it's simplest form can implemented with a constant gain approximation [3], where

$$\boldsymbol{\epsilon}^i := \mathbf{h}(\mathbf{x}_k^i) - \mathbb{E}[\mathbf{h}(\mathbf{x}_k)] \in \mathbb{R}^m, \quad (18a)$$

$$\mathbf{c}_j := \frac{1}{N}\sum_{i=1}^{N} \epsilon_j^i \mathbf{S}_k^i \qquad \in \mathbb{R}^n, \quad (18b)$$

$$\mathbf{K}_k \approx [\mathbf{c}_1 \cdots \mathbf{c}_m]\mathbf{R}^{-1} \qquad \in \mathbb{R}^{n \times m}, \quad (18c)$$

With this approximation, the two parameters defining filter performance and computational effort in the FPF are the number of particles N and the discretisation of λ in simulating the time evolution of the particle flow, here done in $\lambda/\Delta\lambda$ equidistant steps. While retaining the property of consistency [2] as $N \to \infty$, the FPF requires significantly less particles to achieve results comparable to the PF [5], [12], making it a strong candidate for a real-time implementation.

IV. Evaluation

In this section, we consider the two main algorithmic components in each estimator, the prediction step, $\mathcal{A}_p$, and the measurement update step, $\mathcal{A}_u$. These components are run at rates $f_p \in (0, 500]$ Hz and $f_u \in (0, 100]$ Hz respectively, constrained by limitations of sampling rates in the IMU sensor and UWB systems respectively. Yet another constraint in any the real-time implementation is the computational power. The CPU clock frequency, instruction set and number of cores give hard constraints in a complexity, C, in terms of a number of FLOP per second. This implies a constraint in terms of the utilization metric (1), with $\mathcal{U} < \mathcal{U}_{max} \in [0, 1]$ depending on the CPU. To relate results on asymptotic complexity to computational time and the metric of utilization, real-time implementations of selected algorithms are done on a Cortex M4, commonly used for embedded applications. All

algorithms are optimized to minimize computational time, running FreeRTOS in combination with the standardized CMSIS DSP ARM math library [22]. Evaluation of the filters, $\{\mathcal{A}_p, \mathcal{A}_u\}$, is done on a set of problems, $\mathcal{P} = \{\mathcal{P}_i\}$, defined in the next section, using Monte-Carlo simulations at rates approximately solving

$$(f_p^*, f_u^*) = \min_{(f_p, f_u)} \sum_{i=1}^{|\mathcal{P}|} MSE(f_p, f_u, \mathcal{A}_p, \mathcal{A}_u, \mathcal{P}_i) \quad (19)$$

such that

$$\mathcal{U} < \mathcal{U}_{max}, f_p < 500, f_u < 100. \quad (20)$$

A. Problem Definitions

The considered problems $\mathcal{P}_i$ are defined by periodic sequences of time-varying torques $\boldsymbol{\tau}(t) = \left[\tau_1(t), \tau_2(t)\right]^T$ to the GV system (3) with parameters and anchor placement given in Appendix VI-B. The first input sequence (A),

$$\tau_1^A(t) = 0.15\sin(4t) + 0.1\sin(0.1t)$$
$$\tau_2^A(t) = 0.3\sin(2t) + 0.1\sin(0.1t)$$

causes an excitation of all states in (3) and movement outside of the convex hull of the UWB anchors where the Cramer-Rao lower bound of the position estimate increases significantly [23]. The second sequence (B) is defined with the Heaviside function $H(t)$, with a pulse $s(a, b, t) = H(t - a) - H(t - b)$, for some scalar $a, b \in \mathbb{R}$. The wheel torques

$$\tau_1^B(t) = 0.5[s(0, 1, t) - s(1, 2, t)] + [s(4, 5, t) - s(5, 6, t)]$$
$$\tau_2^B(t) = 0.5[s(0, 1, t) - s(1, 2, t)] - [s(4, 5, t) - s(5, 6, t)]$$

with $t \in [0, 6)$ are repeated over the simulation time t_s. This second sequence causes positional movement within the convex hull of the anchors, but the symmetry makes the GV stand completely still on $t \in [2, 4]$ yielding periodic deadlock drift of the estimates. Both inputs give rise to a volatile velocity response on $(\mathbf{v}_\mathcal{B}^T \mathbf{v}_\mathcal{B})^{1/2} \lesssim 1\ m/s$ and multiple revolutions in $\theta_t(t)$ over the simulation time. Combining the wheel torque sequences with, applied biases, dimensionality of the UWB measurements, $\dim(\mathbf{y}_d)$, and camera measurements, $\dim(\mathbf{y}_c)$, six problems of interest are defined in Table IV-A.

TABLE I
MATRIX OPERATIONS AND ASSOCIATED COMPLEXITY

Problem	Input	$\dim(\mathbf{y}_d)$	$\dim(\mathbf{y}_c)$	t_s	$(a_{bt}^x, a_{bt}^y, \omega_{bt})$
$\mathcal{P}_1$	$\boldsymbol{\tau}^A(t)$	0	2	30	0
$\mathcal{P}_2$	$\boldsymbol{\tau}^B(t)$	0	2	30	0
$\mathcal{P}_3$	$\boldsymbol{\tau}^A(t)$	6	0	30	0
$\mathcal{P}_4$	$\boldsymbol{\tau}^B(t)$	6	0	30	0
$\mathcal{P}_5$	$\boldsymbol{\tau}^A(t)$	6	2	60	(0.2,-0.2,0.1)
$\mathcal{P}_6$	$\boldsymbol{\tau}^B(t)$	6	2	60	(0.2,-0.2,0.1)

B. MSE Performance at Fixed Frequencies

To motivate the use of the FPF based on estimator accuracy, a simulation study is done with 100 Monte-Carlo executions for the problem $\mathcal{P}_3$ where $\dim(\mathbf{y}_k) = 6$ and $\boldsymbol{\tau}^A(t)$ is applied to the GV system (3). The utilization constraint is disregarded in (19), with prediction and update steps executed at $f_p = f_u = 100$ Hz. The MSE of the state estimate is plotted as a function of the number of particles with a 99% confidence interval (see Figure 2).

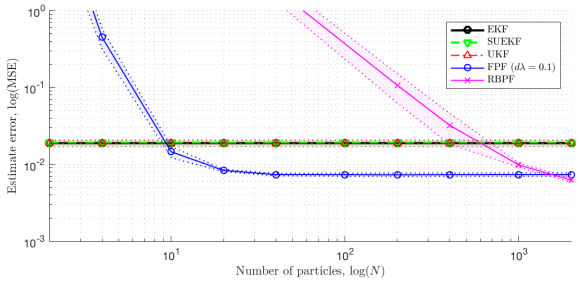

Fig. 2. State estimate MSE as a function of the number of particles in 100 Monte-Carlo simulations with a 99 % confidence interval on problem $\mathcal{P}_3$.

Clearly, the Monte-Carlo simulations yield a relatively low and repeatable MSE for the Kalman filters, with the SUEKF performing marginally worse than the regular EKF and UKF. The UKF propagates a total of 17 samples in the unscented transform, but despite this, the performance of the FPF in this experiment is unparalleled already at $N = 10$ particles for low particle counts. The RBPF gives satisfactory performance first at $N > 400$, with performance rivalling the FPF first at $N \approx 3 \cdot 10^3$. The reason for not showing higher particle counts in Figure 2 is that the problem becomes computationally intractable, with single FPF/RBPF runs at $N = 4000$ taking several hours to complete. An explanation for this given in the complexity analysis in the next section.

C. Complexity and Measurement Dimensionality

Consider next how the complexity of the two algorithmic components, $\mathcal{O}_p$ and $\mathcal{O}_u$, scale with measurement dimensionality in the considered filters. With $\dim(\mathbf{y_k}) \in [2, 10]$ UWB measurements, the complexity is shown in Figure 3.

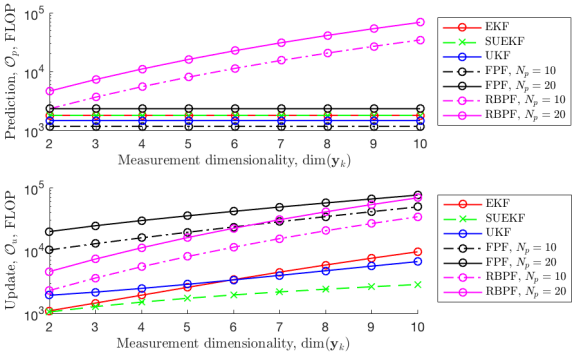

Fig. 3. Complexity of the update and prediction steps in the EKF, SUEKF, UKF, FPF and RBPF as a function of the number of UWB measurements.

From this result we note some fundamental differences in the filters. The complexity of the prediction per particle is small in the FPF, which with a number of particles $N = 10$ is less complex than the UKF prediction, but still

less so than the EKF predictions due to its dependence on matrix operations. While scaling of complexity in the FPF prediction step, $\mathcal{A}_p$, is linear with $\dim(\mathbf{y}_k)$, the complexity of the update step, $\mathcal{A}_u$, scales super-linearly. The complexity of the update step of the FPF is greater than the Kalman-type filters at a particle count of $N = 3$, increasing with both N and the number of measurements. To rival and supersede the FPF at $N \approx 20$ in terms of accuracy, the RBPF has to be run $N \approx 3 \cdot 10^3$ as shown in the previous section. However, the complexity per particle in both RBPF is slightly higher than that of the FPF, due to the high cost of the prediction step. As such, the a computational complexity of the RBPF is large enough to disqualify it from further analysis on the embedded system, while the FPF at $N \approx 20$ may be implementable.

D. Computational Time

The general trends in complexity of the algorithm components is seen in their mean execution time (see Figure 4), with some notable deviations. Firstly, the predictive step in the FPF requires more computational effort than indicated by the complexity analysis, largely due to the need for realising the Gaussian noise on each iteration, here done by the Multiply-With-Carry algorithm and a Box-Muller transform [24]. Secondly, there is a significant difference in computational time between the UKF and EKF implementations, but both can still be run at the maximum rates of each algorithmic component, with $(f_p, f_u) = (500, 100)$ $[Hz]$.

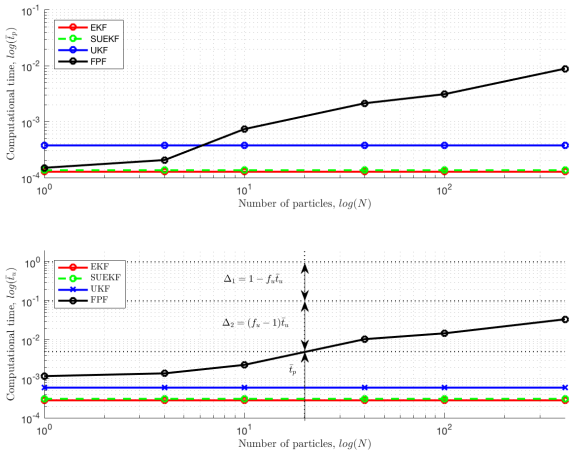

Fig. 4. Mean computational time of the prediction step (top) and the update step (bottom) as a function of the number of particles with $\dim(\mathbf{y}_k) = 6$.

The margins for f_p and f_u are clearly visible if the CPU utilization $\mathcal{U} \in [0, 1]$ is known (see Figure 4). For instance, we wish to run the measurement update with in the FPF $N = 20$ particles and utilise approximately $f_p \bar{t}_p = 0.1$ of the CPU, the update rate should be set to $f_p \approx \Delta_2 / \bar{t}_p + 1 \approx 21$ Hz. Note that the EKF update step can be run a factor 10 faster at this utilisation. In general, there are many algorithmic components at work in the real-time system, such as communication tasks and sensory control. In total, these components take up a utilization corresponding to $\mathcal{U}_{other} \approx 0.06$. If some margin is allocated to an idle task to prevent watchdog timeouts, a maximum utilization of the state estimator is set to 0.8. The prediction that $(f_p^*, f_u^*) = (500, 100)$ $[Hz]$ can then be used for the EKF/SUEKF/UKF, constrained only by the sensor sampling rates. For FPF run with $N = 20$ particles and $d\lambda = 0.1$, the rates approximately solving (19) are found as $(f_p^*, f_u^*) = (210, 100)$. With these rates, the task load of the CPU (see Figure 5) makes it clear that the Kalman filters reach their potential far before all computational resources have been utilised on the processor.

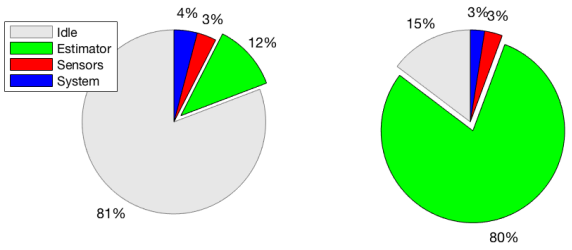

Fig. 5. CPU utilization $\mathcal{U}$ of the Cortex M4 with $\dim(\mathbf{y}_k) = 6$ running the EKF at $(f_p^*, f_u^*) = (500, 100)$ $[Hz]$ (left) and the FPF with $N = 20$ particles and $d\lambda = 0.1$ at $(f_p^*, f_u^*) = (210, 100)$ Hz (right).

E. MSE with Utilization Constraints

Next, we consider the estimate MSE when running the filters at rates found as the approximate solution to (19) on the defined problems $\mathcal{P}$ in 100 Monte-Carlo runs with

$$\mathbf{x}_0 - \mathbb{E}[\hat{\mathbf{x}}_0] = \mathbf{0}. \tag{21}$$

This is done for the EKF/SUEKF/UKF/FPF, as these are the only filters implemented on the Cortex M4, and therefore the only filters for which (f_p^*, f_u^*) are known (see Table IV-E).

TABLE II

MSE OF THE ESTIMATE AT (f_p^*, f_u^*), WHERE ENTRIES MARKED † CONVERGE TO AN MSE>1 IN 95% OF THE MC-EXECUTIONS.

	EKF	SUEKF	UKF	FPF
$\mathcal{P}_1$	$4.14 \cdot 10^{-3}$	$4.15 \cdot 10^{-3}$	$4.89 \cdot 10^{-3}$	$2.50 \cdot 10^{-3}$
$\mathcal{P}_2$	$1.55 \cdot 10^{-2}$	$1.55 \cdot 10^{-2}$	$1.63 \cdot 10^{-2}$	$5.62 \cdot 10^{-3}$
$\mathcal{P}_3$	$1.53 \cdot 10^{-2}$	$1.54 \cdot 10^{-2}$	$1.52 \cdot 10^{-2}$	$5.69 \cdot 10^{-3}$
$\mathcal{P}_4$	$\mathbf{3.15}$ †	$\mathbf{3.16}$ †	$\mathbf{1.07}$ †	$9.12 \cdot 10^{-3}$
$\mathcal{P}_5$	$4.17 \cdot 10^{-3}$	$4.18 \cdot 10^{-3}$	$4.83 \cdot 10^{-3}$	$2.40 \cdot 10^{-3}$
$\mathcal{P}_6$	$1.52 \cdot 10^{-2}$	$1.52 \cdot 10^{-2}$	$1.68 \cdot 10^{-2}$	$5.67 \cdot 10^{-3}$

Note that the EKF and SUEKF show no significant difference on any problem, as the sequential update mainly affects computational time and not the estimate accuracy [17]. The result in the highlighted column conclusively shows that the FPF with $N = 20$ and $d\lambda = 0.1$ can and should be implemented on the Cortex M4 for the considered dynamics (5) based on its superior MSE performance. The FPF is reliable across all tested combinations of camera measurements and UWB measurements, and it notably outperforms the Kalman filters for the highly difficult problem $\mathcal{P}_4$ with the torque sequence causing intermediary deadlock drift, for which at least 5% of the Kalman filter MC-executions diverge in the

sense that the total estimate $MSE > 1$ at $t = t_s$. This divergence in the Kalman filters motivates further tests of filter robustness to poor initial estimates.

F. Robustness To Initial Conditions

As shown in the previous section, the nonlinear Kalman filters suffer from potential robustness issues for problems reminiscent of $\mathcal{P}_4$. This is due to the inaccurate assumptions of a Gaussian state distribution when using UWB measurements, and the approximation nonlinear dynamics using the first order Taylor expansion in the EKF and the unscented transform in the UKF. As such, the estimates may diverge if the state distribution is volatile and multimodal, and especially if the nonlinear states estimates such as the rotation at any time differ greatly from the true system states.

When considering a real-time implementation, diverging estimates will cause a filter reset from which the filter must recover and converge. To study the robustness of the filters in this respect, let $\mathbf{U} \sim \mathcal{U}(-1, 1) \in \mathbb{R}^8$ denote a uniformly distributed random variable and introduce an estimate error

$$\mathbf{x}_0 - \mathbb{E}[\hat{\mathbf{x}}_0] = [1,\ 1,\ 1,\ 1,\ \pi,\ 1,\ 1,\ 0.1]^T \cdot \mathbf{U}.$$

In this example, problems $\mathcal{P}_3$ and $\mathcal{P}_4$ are studied with 100 MC-executions with the filters run at their corresponding rates (f_p^*, f_u^*) (see Table IV-F). An estimate is considered to have converged in $t = t_s = hN_s$ if $||\mathbf{x}_{N_s} - \hat{\mathbf{x}}_{N_s}||_2 < 10^{-1}$, with the limit set based on MSE results with perfect estimate initialization in Section IV-E.

TABLE III
PERCENTAGE OF CONVERGED MONTE-CARLO EXECUTIONS

Problem	EKF	SUEKF	UKF	FPF
$\mathcal{P}_3$	46%	45%	48%	56%
$\mathcal{P}_4$	33%	33%	37%	70%

This demonstrates that the FPF is more robust to perturbations in the initial estimate in the considered problems. Upon closer inspection, the divergence in both filters depend greatly on the error in the initial rotational estimate (see Figure 6). When visualizing the FPF and EKF run on $\mathcal{P}_3$, both filters converge from an initial error of $|\theta(0) - \hat{\theta}(0)| < 0.4\ rad$ in all MC-executions, and convergence is highly improbable when $\cos(\theta(0) - \hat{\theta}(0)) < 0$. The skew seen in the interval of $\hat{\theta}(0)$ from which convergence is more likely may be caused by the shape state trajectory and the geometry of the anchor placement (see Appendix VI-B).

A typical simulation of divergence in problem $\mathcal{P}_3$ is demonstrated in Figure 7, where the FPF state estimate converges and the EKF diverges due to the rotational estimate error increasing linearly with time, by consequence of the gyroscopic bias having drifted to approximately $-1\ rad/s$.

V. CONCLUSIONS

In this paper, the utility of the FPF has been analyzed for embedded real-time implementations in IMU-driven dynamics configured on $SE(2)$. Constraining the systems computational power and sampling rates of the IMU, camera and

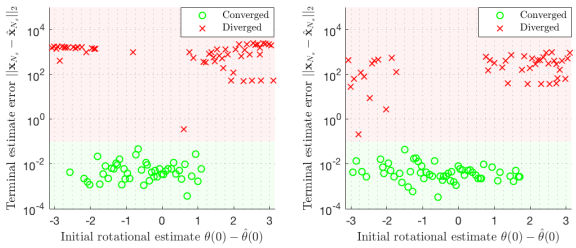

Fig. 6. Terminal estimation error for problem $\mathcal{P}_3$ as a function of the initial rotational estimate in the EKF (left) and the FPF (right).

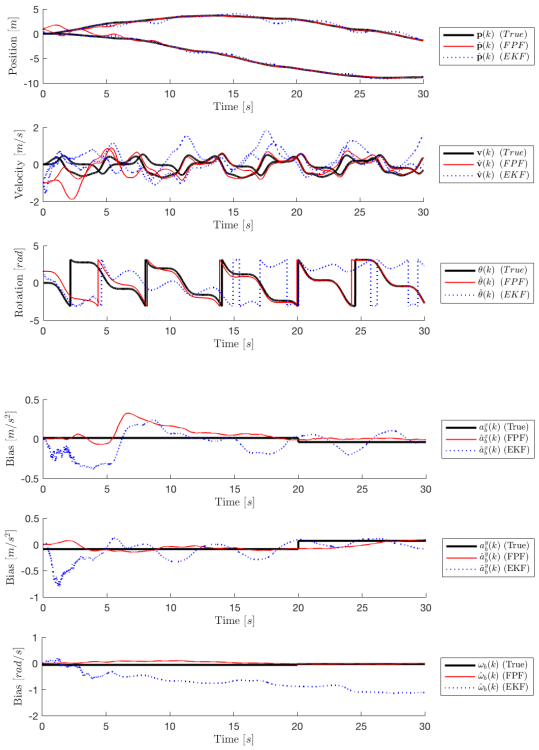

Fig. 7. True and estimated states in one of the MC-executions of problem $\mathcal{P}_3$ using the EKF (blue) and FPF (red) with a poor initial state estimate with an added bias step change at $t = 20\ s$ for demonstrative purposes.

UWB sensors, a comparative analysis was done with the EKF, SUEKF, UKF, RBPF and FPF to determine the scaling of performance with computational complexity for a set of standard problems (see Table IV-A). An initial analysis of estimate MSE for problem $\mathcal{P}_3$ indicated the FPFs utility, where it outperformed the Kalman based filters with $N \gtrsim 10$, rivalled by the RBPF only at $N \gtrsim 1000$ particles (see Figure 2). In terms of complexity, the FPF introduces a significant computational effort which scales super-linearly with the number of measurements and particles (see Figure 3), but the estimator was nonetheless run efficiently on the Cortex M4 with $N = 20$ and $d\lambda = 0.1$ using $(f_p^*, f_u^*) = (210, 100)$ (see Figure 5). At these rates, the FPF was compared to the EKF, SUEKF and UKF for a wide range of problems. As indicated in the initial MSE analysis, the FPF outperformed the rivals with any combination of 6 UWB measurements

and 2 camera measurements for realistic input torque and bias sequences (see Table IV-E). Furthermore, the robustness of the FPF to perturbations in the initial filter estimate was shown to be superior to that of the Kalman filters (see Table 7). This analysis also showed that all considered estimators work reliably if initialised with a rotation such that $\cos(\theta(0) - \hat{\theta}(0)) \gtrsim 0$ for the considered problems.

In conclusion, the FPF is a good and robust alternative to standard non-linear Kalman filters and readily implementable in an embedded applications for IMU-driven $SE(2)$-configured dynamics (5). In addition, it is likely a good candidate of systems with a similar state-space dimensionality. If the CPU has less processing power than the Cortex M4 or if the utilization constraint is greatly decreased, the SUEKF should be considered as an alternative due to its performance being similar to the EKF and UKF with a slightly reduced complexity in the update step. However, if computational resources are is significantly increased, the RBPF should be reconsidered.

VI. APPENDIX

A. Asymptotic Complexity

Conservative asymptotic complexity of floating point operations (FLOP) are given in the Table VI-A. The † indicates the number of FLOP required to evaluate an operation assuming nominal state-space dimensionality of $\dim(\mathbf{x}_k) = 8$, $\dim(\mathbf{y}_k) = 6$. Furthermore, the ‡ indicates FLOPs required to evaluate the Jacobians of the prediction and update steps.

TABLE IV
MATRIX OPERATIONS AND ASSOCIATED COMPLEXITY

Operation	Input	Complexity
Matrix add.	$\mathbb{R}^{nm}, \mathbb{R}^{nm}$	nm
Matrix mult.	$\mathbb{R}^{nm}, \mathbb{R}^{mp}$	nmp
QR	$\mathbb{R}^{nn}$	$2n^3$
LU	$\mathbb{R}^{nn}$	$(2/3)n^3$
Cholesky	$\mathbb{R}^{nn}$	$(1/3)n^3$
$\mathbf{Ax} = \mathbf{b}$ (QR)	$\mathbb{R}^{nn}, \mathbb{R}^{n}$	$2n^3 + 3n^2$
$\mathbf{Ax} = \mathbf{b}$ (LU)	$\mathbb{R}^{nn}, \mathbb{R}^{n}$	$(2/3)n^3 + 2n^2$
$\mathbf{f}(\mathbf{x}(k))^{\dagger}$	$\mathbb{R}^{8}, \mathbb{R}^{3}$	44
$\mathbf{y}_c(\mathbf{x}(k))^{\dagger}$	$\mathbb{R}^{8}$	0
$\mathbf{y}_d(\mathbf{x}(k), N_d)^{\dagger}$	$\mathbb{R}^{8}, N_d$	$8N_d$
Pred. Jac.‡	$\mathbb{R}^{8}, \mathbb{R}^{3}$	42
Update Jac.‡	$\mathbb{R}^{8}$	$8m + 4$

B. Nominal Parameters

The model parameters used in all simulations are $h = 0.1$ m, $r = 0.1$ m, $J = 0.1$, kg/m^2 and $m = 0.1$ kg, with

$$\mathbf{P} = [\mathbf{p}_1, \cdots, \mathbf{p}_6] = \begin{bmatrix} 1.5 & 1.5 & 1.5 & -1.5 & -1.5 & -1.5 \\ 3 & 0 & -3 & -3 & 0 & 3 \\ 3 & 0 & 3 & 0 & 3 & 0 \end{bmatrix}$$

at all times, with the zero-mean Gaussian process noise terms $\mathbf{w}_{mn}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{nn})$ and $\mathbf{w}_{bn}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{bb})$ defined by

$$\mathbf{Q}_{nn} = 0.1 \cdot \begin{bmatrix} 1 & 0.05 & 0.01 \\ 0.05 & 1 & 0.01 \\ 0.01 & 0.01 & 0.1 \end{bmatrix}, \ \mathbf{Q}_{bb} = 10^{-2} \cdot \begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{bmatrix}.$$

REFERENCES

[1] T. Yang, P. G. Mehta, and S. P. Meyn, "A mean-field control-oriented approach to particle filtering," in *American Control Conference (ACC), 2011*. IEEE, 2011, pp. 2037–2043.

[2] ——, "Feedback particle filter with mean-field coupling," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*. IEEE, 2011, pp. 7909–7916.

[3] T. Yang, R. S. Laugesen, P. G. Mehta, and S. P. Meyn, "Multivariable feedback particle filter," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*. IEEE, 2012, pp. 4063–4070.

[4] T. Yang, P. G. Mehta, and S. P. Meyn, "Feedback particle filter," *IEEE transactions on Automatic control*, vol. 58, no. 10, pp. 2465–2480, 2013.

[5] K. Berntorp, "Feedback particle filter: Application and evaluation," in *Information Fusion (Fusion), 2015 18th International Conference on*. IEEE, 2015, pp. 1633–1640.

[6] A. Romanenko and J. A. Castro, "The unscented filter as an alternative to the EKF for nonlinear state estimation: a simulation case study," *Computers & chemical engineering*, vol. 28, no. 3, pp. 347–355, 2004.

[7] A. H. Jazwinski, *Stochastic processes and filtering theory*. Courier Corporation, 2007.

[8] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*. Ieee, 2000, pp. 153–158, (visited on 08-07-2016).

[9] T. Schon, F. Gustafsson, and P.-J. Nordlund, "Marginalized particle filters for mixed linear/nonlinear state-space models," *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2279–2289, 2005.

[10] R. Karlsson, T. Schon, and F. Gustafsson, "Complexity analysis of the marginalized particle filter," *IEEE Transactions on Signal Processing*, vol. 53, no. 11, pp. 4408–4411, 2005.

[11] T. B. Schon, R. Karlsson, and F. Gustafsson, "The marginalized particle filter in practice," in *Aerospace Conference, 2006 IEEE*. IEEE, 2006, pp. 11–pp.

[12] C. Zhang, A. Taghvaei, and P. G. Mehta, "Feedback particle filter on matrix lie groups," in *American Control Conference (ACC), 2016*. IEEE, 2016, pp. 2723–2728.

[13] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[14] R. Dalce, T. Val, and A. V. Bossche, "Comparison of indoor localization systems based on wireless communications," 2011.

[15] Decawave, "Dw1000 user manual," 2014. [Online]. Available: http://thetoolchain.com/mirror/dw1000/dw1000_user_manual_v2.05.pdf

[16] M. W. Mueller, M. Hamer, and R. D'Andrea, "Fusing ultra-wideband range measurements with accelerometers and rate gyroscopes for quadrocopter state estimation," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1730–1736.

[17] P. J. Huxel and R. H. Bishop, "Navigation algorithms and observability analysis for formation flying missions," *Journal of guidance, control, and dynamics*, vol. 32, no. 4, p. 1218, 2009.

[18] M. Greiff, "Models and definitions for IMU-driven filtering on SE(2)," 2017. [Online]. Available: https://github.com/mgreiff/preprints/blob/master/preprintA.pdf

[19] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on signal processing*, vol. 50, no. 2, pp. 425–437, 2002.

[20] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on signal processing*, vol. 50, no. 2, pp. 174–188, 2002.

[21] J. D. Hol, T. B. Schon, and F. Gustafsson, "On resampling algorithms for particle filters," in *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*. IEEE, 2006, pp. 79–82.

[22] V. Nikolskiy and V. Stegailov, "Floating-point performance of arm cores and their efficiency in classical molecular dynamics," in *Journal of Physics: Conference Series*, vol. 681, no. 1. IOP Publishing, 2016, p. 012049.

[23] A. Ledergerber, M. Hamer, and R. D'Andrea, "A robot self-localization system using one-way ultra-wideband communication," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 3131–3137.

[24] H. R. Neave, "On using the box-muller transformation with multiplicative congruential pseudo-random number generators," *Applied Statistics*, pp. 92–97, 1973.