35th Annual International Conference of the IEEE EMBS
Osaka, Japan, 3 - 7 July, 2013

Modification and fixed-point analysis of a Kalman filter for orientation estimation based on 9D inertial measurement unit data

Hans-Peter Brückner, Christian Spindeldreier, Holger Blume, *Senior Member, IEEE*
Institute of Microelectronic Systems, Leibniz Universität Hannover, Hannover, Germany
{brueckner, spindeldreier, blume}@ims.uni-hannover.de

***Abstract*— A common approach for high accuracy sensor fusion based on 9D inertial measurement unit data is Kalman filtering. State of the art floating-point filter algorithms differ in their computational complexity nevertheless, real-time operation on a low-power microcontroller at high sampling rates is not possible. This work presents algorithmic modifications to reduce the computational demands of a two-step minimum order Kalman filter. Furthermore, the required bit-width of a fixed-point filter version is explored. For evaluation real-world data captured using an Xsens MTx inertial sensor is used. Changes in computational latency and orientation estimation accuracy due to the proposed algorithmic modifications and fixed-point number representation are evaluated in detail on a variety of processing platforms enabling on-board processing on wearable sensor platforms.**

I. INTRODUCTION

Inertial measurement units (IMU) usually consist of triads of gyroscopes, accelerometers and magnetometers. In order to determine orientation based on these sensors common algorithmic approaches are either gyroscope integration, vector observation, complementary filters or Kalman filtering [1]. Modern highly integrated, small-sized, and lightweight [2], [3] IMUs, commonly comprise triaxial inertial sensors and a microcontroller (MCU) for interfacing the sensors and communication with the data processing platform. In wireless IMUs the orientation estimation is performed on the host processing platform due to performance limitations of the low-power MCUs.

On-board computation of the orientation is desired to lower the transmission latency by reducing the amount of data to be transmitted. Processed orientation data is represented by a four element quaternion instead of the RAW data comprising three data elements for each of the three tri-axial sensors (e.g. gyroscope, accelerometer and magnetometer). The transmission data rate is often a limiting factor in motion capturing using multiple IMUs.

A sensor network comprising multiple wearable IMUs allows motion capturing in rehabilitation sessions. An application relying on the movement data acquired is movement sonification. Thereby, an audio feedback dependent on the captured movement [4] is generated. Recent research showed a remarkable benefit from sonification of movements for patients in stroke rehabilitation [5]. Wearable IMUs fixed at the patient's body are used to capture complex upper body movements. Based on a connected rigid chain body model, parameters like angles between body segments, positions or velocities can be computed.

Important criteria for the usage of inertial sensors in rehabilitation sessions are long term usability in home based environments. Therefore, gyroscope drift and susceptibility to ferromagnetic materials should not affect motion capturing. Compensation of magnetic disturbance is a highly computational demanding algorithmic problem. Therefore, the algorithms cannot be performed in real-time on low power hardware platforms.

Based on a previous study [6] evaluating computational latency and orientation estimation accuracy of eight inertial sensor fusion algorithms the Kalman filter according to Lee and Park 2009 [7] was chosen for sensor fusion, as this algorithm constitutes the best tradeoff between orientation estimation accuracy and latency considering the movement sonification application demands. This two stage Kalman filter comprises a four element state vector and accelerometer and magnetometer data preprocessing utilizing the O2OQ (optimal two-observation quaternion estimation method) algorithm further reducing the computational demands compared to a unique Kalman filter with larger state vector dimensions. The structure of the algorithm is shown in Figure 1.

In this work algorithmic modifications reducing the computational demands while preserving the orientation estimation accuracy for the Kalman filter algorithm are proposed and evaluated. Second, a fixed-point version of the sensor fusion algorithm is presented and orientation estimation degradation and the platform dependent influence on the execution time is evaluated. This analysis allows to choose the preferred implementation for each hardware architecture. Orientation estimation accuracy is evaluated based on datasets captured using a Xsens MTx [3] inertial sensor and a Qualisys optical capturing system as golden reference.

The paper is organized as follows: Section II presents related work highlighting wireless, wearable IMUs and sensor fusion algorithms. Section III provides details about the Kalman filter optimization and the generation of a fixed-point version. Section IV highlights the accuracy and latency evaluation results of the proposed algorithmic modifications. Conclusions are presented in section V.

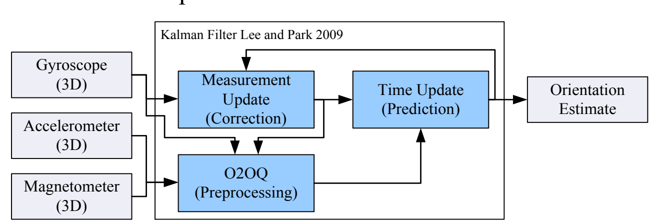

Figure 1. Minimum order two step Kalman filter structure

II. RELATED WORK

In literature there are multiple algorithms for inertial sensor data fusion [7], [8], [9]. It is common to benchmark those algorithms using artificial datasets. Therefore, a fair comparison of different approaches is hampered. Real-world data-sets used in this work were proposed in [6] and comprise a long-term and a magnetic disturbance data-set.

Complementary filter and Kalman filter algorithms for inertial sensor fusion were evaluated based on a captured walking trial in [1]. Since this work further optimized vector observation techniques [9], Kalman filters with minimized computational costs [7] and magnetic disturbance compensation [10] have been developed. For this reason, the computational latency and orientation estimation accuracy of eight algorithms was evaluated in a previous study [6].

Complementary filters were not considered for sensor fusion as all of the Kalman filters evaluated in [6] provide considerably better results in contrast to the algorithms benchmarked in [1]. Complementary filters fuse orientation estimates based on high-pass filtered gyroscope data and low pass filtered accelerometer data. According to [1] these filters provide only slightly improved results compared to vector observation methods while increasing computational costs.

A simplified Kalman filter algorithm is proposed in [11] using pre-computed a priori and a posteriori error covariance matrices. This reduces the computational demands by about 25 %. However, the authors do not consider an additional magnetometer, as well as testing multi axis movements.

In [12] a two stage Kalman with reduced computational complexity is presented. The preprocessing relies on the QUEST / FQA algorithm, which shows a worse accuracy compared to the O2OQ algorithm used in this work. A custom ASIP was designed to speed up computations and allow real-time operation at a 1 kHz sampling rate. The paper focuses in ASIP design and lacks an orientation estimation accuracy analysis and a detailed fixed-point analysis.

III. KALMAN FILTER MODIFICATION

To reduce the computational demands of the chosen Kalman filter algorithm two strategies are presented. First of all, algorithmic modifications are implemented to reduce the number of operations required for the computation of a Kalman filter update. Applying predefined values for several matrices is enabled due prior knowledge of constant sensor and process noise characteristic. Second, a fixed-point version of the algorithm is designed to substitute the high latency floating-point operations by lower latency fixed-point operations and thus achieve a speedup on certain processors.

A. Predefinition of Error Covariance Matrices

The authors in [11], [12] propose the reduction of computational complexity by using pre-computed a posteriori and a priori error covariance matrices on their proposed Kalman filter. It has been shown that this modification has negligible influence on the orientation estimation accuracy.

This concept is applied to the Kalman filter according to Lee and Park [7]. The influence on the filter structure is shown in Figure 2. , highlighting the reduction of required computation steps.

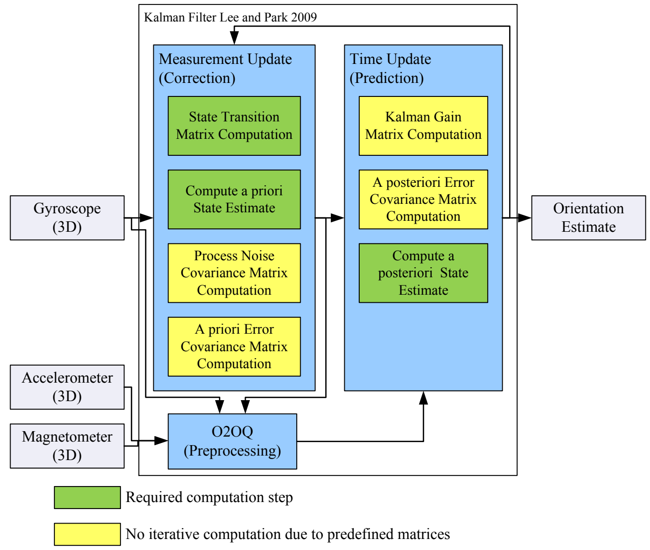

Figure 2. Proposed modified two step Kalman filter structure

Using pre-computed matrices results in less required operations, enabled by knowledge about constant sensor noise characteristics. The Kalman gain matrix also becomes independent from the actual input data and can be pre-computed due to the pre-computed error covariance matrices The pre-computation of the Kalman gain matrix achieves a high reduction of the computational demands, as this eliminates the computation of a 4x4 matrix inverse in each filter step.

A comparison of floating-point operations required for a single Kalman filter step of the modified Kalman filter compared to the initial version is given in TABLE I. The modification significantly reduces the number of operations.

TABLE I. NUMBER OF OPERATIONS PER FILTER STEP

Operation		*'+', '-'*	*'*'*	*'/'*	*Arc Cos*
O2OQ		147	197	31	1
Kalman Filter	Original	579	524	46	0
	Modified	60	37	20	0

B. Transformation from floating-point to fixed-point

To determine required minimal total bit-width and the number of integer and fraction bits for each Kalman filter variable a template based C++ framework, described and applied in [13], is used. The framework enables code reusage at data-type level by abstracting the data-type.

For analysis using this framework the floating-point data type (e.g. float or double) has to be replaced by the frameworks template based data type. Setting the template data-type to float or double results in a floating-point reference implementation. For fixed-point analysis the template data-type has to be replaced by a hybrid floating-point or integer type (e.g. int32_t, int64_t) and parameters specifying total bit-width and fixed-point position.

An iterative design space exploration is performed, varying total bit-width and fixed-point position. An additional data-type implemented in the framework allows accuracy assessment compared to floating-point and overflow detection at bit-level.

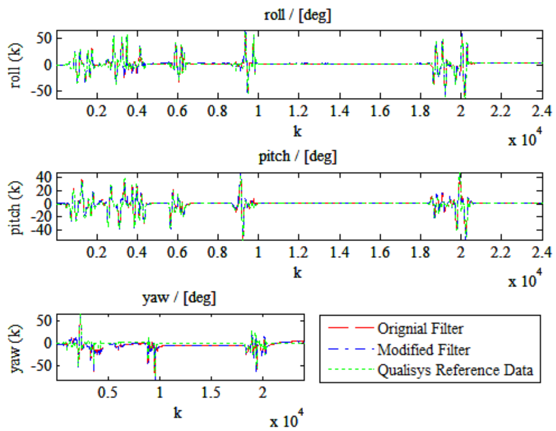

Figure 3. Orientation estimation variation due to algorithmic modification demonstrating negligible effect of utilizing predefined matrices

IV. Evaluation Of Kalman Filter Modifications

Orientation estimation accuracy of the original and modified Kalman filter was determined based on a 24,000 samples arbitrary rotation data-set [6]. The data-set comprises Xsens MTx [3] inertial sensor data and Qualisys optical motion tracking system data as golden reference. The MTx sensor and the optical system are both sampled at 120 Hz. Within the data-set the inertial sensor is rotated about +/- 90° along each axis in between the sensor stayed in the starting position. The data-set is also used to determine the effect of bit-width reduction in evaluating the fixed-point Kalman filter.

A. Evaluation of Kalman filter modification

To evaluate the influence of the Kalman filter modification the estimation of the original and the modified filter algorithm are compared to the golden reference data-set.

The differences in the computed orientation estimation are shown in Figure 3. The difference of the orientation estimation is 0.08 °RMS, 0.01 °RMS and 0.19 °RMS (roll, pitch, and yaw) compared to the original algorithm. Providing an absolute accuracy of 2.36 °RMS, 2.50 °RMS and 6.43 °RMS (roll, pitch, and yaw) compared to the golden reference. Therefore, the influence on accuracy due to the algorithmic modification is negligible while highly reducing the number of operations required. Evaluating a further data-set from [6] involving partial external magnetic disturbance to show data independence performance the gap is 0.19 °RMS, 0.09 °RMS and 0.37 °RMS (roll, pitch, and yaw).

B. Influence of total bit-width on estimation accuracy

First of all, the required bit-width of a hybrid floating-point version of the Kalman filter is evaluated, as a starting point for the later fixed-point analysis. This number representation is suitable for ASIP designs, while on other architectures the number of operations required highly increases due to the additional value dependent shift operations.

Second, the relation between total bit-width and orientation estimation accuracy of the fixed-point Kalman filter algorithm is analyzed. The optimal bit-position of the static fixed-point is determined via test runs using the arbitrary rotation reference data-set [6]. Due to the non optimal bit-width utilization accuracy will decrease compared to a floating or hybrid floating-point number representation.

TABLE II. shows the relation between bit-width and orientation estimation accuracy. A detailed analysis of the filter structure shows a high impact of the state propagation step accuracy on the overall Kalman filter performance. Therefore, the influence of a larger bit-width for the computation of this step is evaluated. According to TABLE II. this approach enables a further reduction of the bit-width for all remaining operations. When further reducing the bit-width, based on the performed evaluation, spikes in the orientation estimation occur making the results unusable for further processing.

TABLE II. Relation Between Bit Width And Orientation Error Using Hybrid Floating-Point Number Representation

Kalman filter version	Bit-width		Orientation estimation error			
	State prediction	*Other filter steps*	*Roll / °RMS*	*Pitch / °RMS*	*Yaw / °RMS*	*Median / °RMS*
Original	30	30	1.7	2.3	2.0	2.0
	52	26	1.9	2.0	2.2	2.0
Modified	27	27	1.3	3.0	3.2	3.9
	32	19	1.5	2.3	3.2	2.3

TABLE III. presents the relation between bit-width and orientation estimation accuracy for the Kalman filter using a fixed-point number representation. In contrast to the hybrid floating-point version there is a larger influence of the state prediction step on the overall orientation estimation accuracy. Therefore, an equal scaling of the bit-width for all filter steps results in an inacceptable orientation error. The results show that the modified Kalman filter enables a further reduced bit-width while achieving more reliable orientation estimation, by avoiding the fault-prone matrix inverse computation due to the static Kalman gain matrix. On programmable platforms multiples of the register-width is used for representing the filter variables.

TABLE III. Relation Between Bit Width And Orientation Error Using Fixed-Point Number Representation

Kalman filter version	Bit-width		Orientation estimation error			
	State prediction	*Other filter steps*	*Roll / °RMS*	*Pitch / °RMS*	*Yaw / °RMS*	*Median / °RMS*
Original	62	30	4.7	3.2	16.4	8.1
Modified	51	30	2.0	3.8	6.9	4.2

The relation between °RMS estimation error and total bit-width of the state estimation computation for the fixed-point filter version is shown in Figure 4. The bit-width for all other computation steps is set to 30 bit. Spikes occurring in the orientation are masked due to mean value computation.

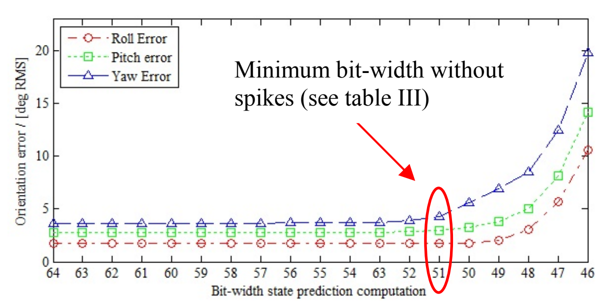

Figure 4. Orientation estimation degradation due to bit-widht scaling

C. Hardware platform dependent latency analysis of the number representation formats

The number representation (e.g. floating-point, hybrid floating-point or fixed-point) achieving lowest latency is platform dependent. Therefore, an analysis considering a widespread number of hardware platforms is performed to figure out the best implementation for each architecture. Factors influencing the optimal number representation are the processors data-width, the presence of a floating-point unit, and the performance of the floating-point operations emulation.

Hardware platforms considered in the evaluation are:

- ATMEGA1281: 8-bit low cost MCU @ 8 MHz
- AT32UC3A0128: 32-bit MCU @ 64 MHz
- Nios II (fast core): Altera soft-core processor with FPU @ 50 MHz
- ARM 1176: 32-bit RISC core with coprocessor FPU @ 700 MHz
- ARM Cortex A8: 32-bit RISC core with fixed-point and floating-point SIMD unit @ 1.0 GHz
- Core i5 760: 64-bit general purpose processor with floating point SIMD unit enabling vector operations @ 2.8 GHz

A detailed comparison of the of the Kalman filter execution times on the different processor cores is presented in TABLE IV. The platform dependent lowest computational latency is highlighted. The bit-width of fixed-point version is set to the required multiples of the processors data-width.

TABLE IV. ORIGINAL AND MODIFIED KALMAN FILTER LATENCY

Execution time on the processing core in µs	C++ reference	C hybrid floating-point	C fixed-point
	Original / Modified Kalman filter latency / [µs]		
Core i5 760	**1.05 / 0.65**	41.22 / 18.77	5.13 / 1.41
ARM Cortex A8	43.74 / 21.77	238.32 / 142.02	**31.46 / 8.41**
ARM 1176	**22.43 / 8.66**	550.56 / 339.39	136.86 / 16.29
AT32UC3A0128[a]	1,899.81 / 853.6	8,046.92 / 6,927.23	**464.08 / 140.47**
ATMEGA 1281[a]	**24,061.50 / 9,430.25**	314,919.88 / 284,640.50	88,002.38 / 31,483.63
Nios II (fast core)	21,662.00 / 8,400.00	8,338.00 / 5,365.00	**2,189.00 / 682.00**

a. Reference algorithm written in C due to compiler

V. CONCLUSIONS

Reduced computational demands of the Kalman filter due to predefined Kalman gain and a priori and a posteriori error covariance matrices achieves an accurate orientation estimation based on the Kalman filter according to [7]. Dependent on the computation platform and the number representation a speedup between 1.1 and 8.4 is achieved. The evaluation points out the optimal implementation for each architecture.

Influencing aspects are the latency of floating-point operations, the latency of emulated floating-point operations and the latency of shift operations due to the transformation into the fixed-point number representation. Dependent on the hardware platform a fixed-point or a floating-point version of the algorithm achieves lowest latency.

A negligible accuracy degradation caused by the Kalman filter modification is shown using a reference data-set with a duration of 3.3 minutes and a sampling rate of 120 Hz.

The bit-width reduction of the hybrid floating-point and fixed-point filter version is mainly limited by spikes in the computed orientation estimation corrupting a further processing of the orientation data. Therefore, the results presented in TABLE II. and TABLE III. are the bit-width limits regarding an orientation estimation without spikes.

ACKNOWLEDGMENT

The authors thank Rochus Nowosielski and Martin Pfitzner from the Institute of Microelectronic Systems for supporting the framework for fixed-point analysis

REFERENCES

[1] A. Young, Comparison of Orientation Filter Algorithms for Realtime Wireless Inertial Posture Tracking, Wearable and Implantable Body Sensor Networks, 2009. BSN 2009. Sixth International Workshop on, pp. 59 -64, 2009.

[2] B. C. Till, M. S. Benning und N. Livingston, Wireless inertial sensor package (WISP), Proceedings of the 7th international conference on New interfaces for musical expression, 2007.

[3] Xsens Technologies BV, [Online]. Available: www.xsens.com.

[4] H.-P. Brückner, M. Wielage und H. Blume, „Intuitive and Interactive Movement Sonification on a Heterogeneous RISC / DSP Platform," *The 18th International Conference on Auditory Display,* 2012.

[5] I. Wallis, T. Ingalls, T. Rikakis, L. Olson, Y. Chen, W. Xu und H. Sundaram, „Real-Time Sonification of Movement for an Immersive Stroke Rehabilitation Environment," *The 13th International Conference on Auditory Display,* 2007.

[6] H.-P. Brückner, C. Spindeldreier, H. Blume, E. Schoonderwaldt und E. Altenmüller, „Evaluation of Inertial Sensor Fusion Algorithms in Grasping Tasks Using Real Input Data," *Wearable and Implantable Body Sensor Networks (BSN), Ninth International Conference on,* pp. 189-194, 2012.

[7] J. K. Lee und E. Park, A minimum-order kalman filter for ambulatory real-time human body orientation tracking, Robotics and Automation, 2009. ICRA '09. IEEE International Conference on, 2009.

[8] H. Black, A passive system for determining the attitude of a satellite, AIAA Journal, vol. 2, no. 7, pp.1350–1351, 1964.

[9] F. Markley, Attitude determination from vector observations: A fast optimal matrix algorithm, Journal of Astronautical Sciences, vol. 41, no. 2, pp. 261-280, 1993.

[10] Y. S. Suh, Orientation Estimation Using a Quaternion-Based Indirect Kalman Filter With Adaptive Estimation of External Acceleration, Instrumentation and Measurement, IEEE Transactions on, 2010.

[11] S. Sabatelli, F. Sechi und L. &. R. A. Fanucci, „A sensor fusion algorithm for an integrated angular position estimation with inertial measurement units," *Design, Automation Test in Europe Conference Exhibition (DATE),* 2011.

[12] S. Sabatelli, M. Galgani und L. &. R. A. Fanucci, „A double stage Kalman filter for sensor fusion and orientation tracking in 9D IMU," *Sensors Applications Symposium (SAS), 2012 IEEE,* 2012.

[13] M. Pfitzner, F. Cholewa, P. Pirsch und H. Blume, „Close-to-hardware error analysis for real-time wavenumber domain processing," *RADAR 2012, 7th International Conference on Radar,* 2012.