On the noise and power performance of a shoe-mounted multi-IMU inertial positioning system

Subhojyoti Bose
oblu IoT
GT Silicon Pvt Ltd
Kanpur, India
Email: subho@oblu.io

Amit K Gupta
oblu IoT
GT Silicon Pvt Ltd
Kanpur, India
Email: amit@oblu.io

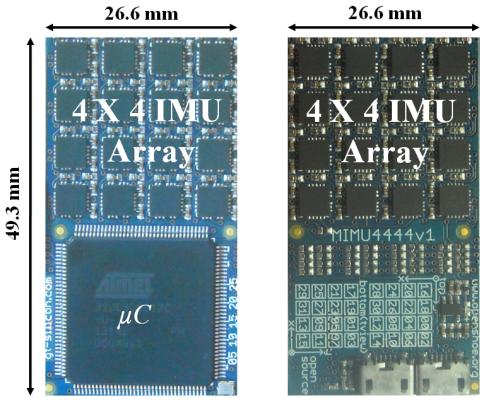
Peter Händel
Department of Information Science
and Engineering
KTH Royal Institute of Technology
Stockholm, Sweden

***Abstract*—Shoe-mounted inertial navigation systems, aka pedestrian dead reckoning or PDR sensors, are being preferred for pedestrian navigation because of the accuracy offered by them. Such shoe sensors are, for example, the obvious choice for real time location systems of first responders. The opensource platform OpenShoe has reported application of multiple IMUs in shoe-mounted PDR sensors to enhance noise performance. In this paper, we present an experimental study of the noise performance and the operating clocks based power consumption of multi-IMU platforms. The noise performances of a multi-IMU system with different combinations of IMUs are studied. It is observed that four-IMU system is best optimized for cost, area and power. Experiments with varying operating clocks frequency are performed on an in-house four-IMU shoe-mounted inertial navigation module (the Oblu module). Based on the outcome, power-optimized operating clock frequencies are obtained. Thus the overall study suggests that by selecting a well-designed operating point, a multi-IMU system can be made cost, size and power efficient without practically affecting its superior positioning performance.**

***Index Terms*—Indoor positioning; IMU; pedestrian dead reckoning; ZUPT; Allan variance; clock budgeting; power optimization.**

I. Introduction

Indoor pedestrian tracking is a subject of intense research in the scientific community. Researchers have used various technologies for tracking indoor pedestrian navigation where the Global Positioning Satellite (GPS) is highly ineffective [1-2]. Due to the rapid boom in smart phone market, the prices of the Micro Electro Mechanical Systems (MEMS) based inertial sensors have drastically reduced. This reduction in market price has motivated a number of researchers and practitioners to use the inertial sensors in indoor navigation applications. Few shoe-mounted prototypes that are available in the market are used for indoor navigations without any prior knowledge of the environment [3-8]. With the recent cost reduction of the sensor technology, sensor-array based approaches, aka multi-IMU systems, have appeared to enhance the performance [9-11]. There are many applications of the multi-IMU devices apart from indoor tracking, namely in autonomous robotics, gaming, fitness monitoring, land survey, medical treatment of movement disorders and workforce monitoring & management to name a few.



(a) Front side (b) Back side

Fig. 1. The Osmium MIMU4444: A massive multi-IMU array based inertial positioning platform.

The purpose of the paper is to find a suitable operating point for multi-IMU devices, based on their noise performance with number of IMUs, to save the cost as well as to reduce the size. Further, to make the shoe-mounted multi-IMU device more power efficient by operating the clocks at optimized frequency.

This paper is organized as follows. Section II gives an overview of the device under test, i.e. the massive multi-IMU array Osmium MIMU4444 as shown in Fig. 1. Noise performance analysis of the IMU arrays is presented in Section III. The clock budgeting based power consumption study is described in Section IV. Conclusions of the study are drawn in Section V.

II. Device Under test

Shoe-mounted inertial navigation devices like Osmium MIMU4444, contain multiple IMUs. As MEMS sensors based inertial positioning systems suffer from drift, Zero-velocity Update (ZUPT) algorithm is used to minimize the accumulation of error [12].

Normal human gait shows a momentary standstill when the shoe sole comes in contact with ground. ZUPT algorithm takes advantage of this phase of human gait by detecting the standstill moment and eliminating any non-zero velocity measurement done by the shoe-mounted device. So the

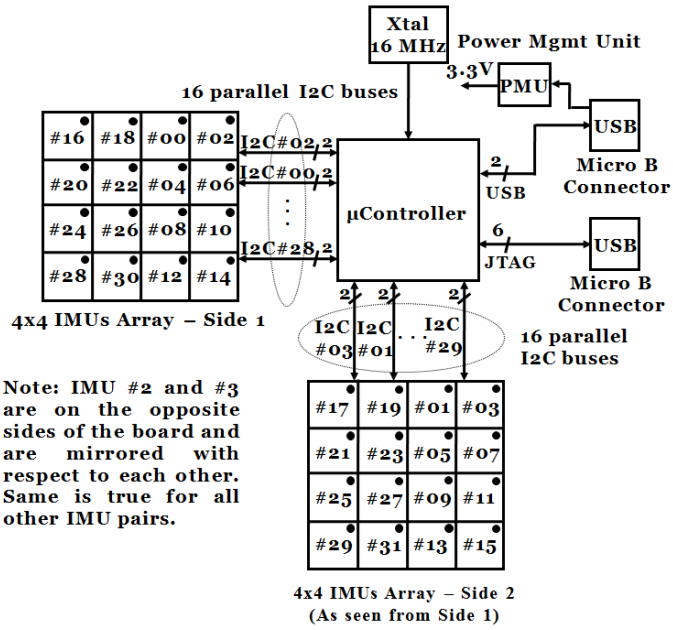


Fig. 2. Block Diagram of Osmium MIMU4444: The massive multi-IMU array platform contains thirty two IMUs on board. It contains two 4x4 IMU-arrays placed in well defined layout on either side of the board, and are mirrored with respect to each other.

MEMS based accelerometers and gyroscopes can be used for measuring the displacement and heading of each and every human step and thereby help in human tracking without any pre-installed infrastructures.

The device under test, shown in Fig. 1, contains thirty two 9-axis MPU9150 IMUs placed on the either side of the module. The IMUs contain 3-axis accelerometers and 3-axis gyroscopes and 3-axis magnetometers. It also contains a pressure sensor. The module has a powerful 32-bits floating point AT32UC3C micro-controller for onboard data acquisition and computation required for implementing the ZUPT algorithm. Other key components of the module include micro-USB connector for data communication as shown in the block diagram in Fig. 2 [13]. The module can be programmed using JTAG programmer.

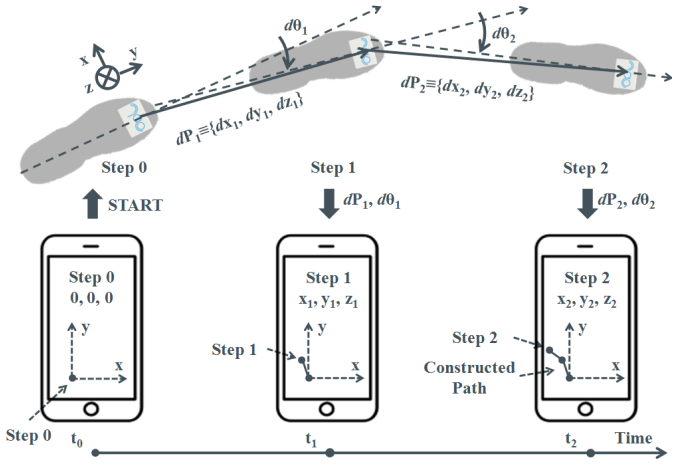


Fig. 3. PDR with shoe-sensors: PDR is simplified with the shoe-mounted multi-IMU array. The device starts transmitting location data at every step, on receiving start command from the application platform. Here dP_i and dθ_i are displacement and change in orientation at every step. Though a smartphone is shown as a user's application platform in the figure, a desktop PC or any other system could well be configured to run user's application.

The module works on the principle of PDR i.e. the process of calculating one's position by estimating the direction and displacement. The device detects steps and gives out the displacement and change in heading, using ZUPT approach. The information provided by the device is used to track the current position based on previous known position. The information is sent via USB to any application platform where one can calculate the current position. The operation based on PDR is shown in Fig. 3 [14].

The device under test is a multi-IMU inertial navigation system based on the open source OpenShoe project (www.openshoe.org) [7-8] where the hardware platform as well as the embedded software is released under the permissive open source Creative Commons Attribution 4.0 International Public License.

The 32-IMU array enables data fusion and thereby reduces independent stochastic errors and improves the navigation performance. Presence of the on-board floating point controller significantly enhances the processing capability which allows IMUs to be simultaneously sampled at maximum allowable rate and carry out data fusion and navigational computation inside the device as illustrated in Fig. 4. The device therefore becomes capable of transmitting low rate PDR data at every step, over USB interface. The device can easily be attached to the shoe, to obtain relative coordinates of the tracked path as PDR data, in the user's application platform .

III. Noise Performance of an IMU Array

The errors in the IMUs are caused by noise sources which are statistically independent. Many methods for modelling such noise are developed. The simplest and most used is the Allan variance time-domain analysis technique. It involves analysing a sequence of data in the time domain, to measure frequency stability in oscillators. This method can also be used to determine the intrinsic noise in a system as a function of the averaging time. The method is simple to compute and understand. It is one of the most popular methods today for identifying and quantifying different noise terms that exist in any inertial sensor data. The method has been adapted to characterize random-drift of a variety of devices including MEMS based IMUs [15].

The Allan deviation (AD) is a direct measurable quantity which can provide information on the types and magnitude of various noise terms. It is calculated as

$$AD = \sqrt{\frac{1}{2(n-1)} \sum_{i=1}^{n} (a_{\tau_{i+1}} - a_{\tau_i})^2}$$

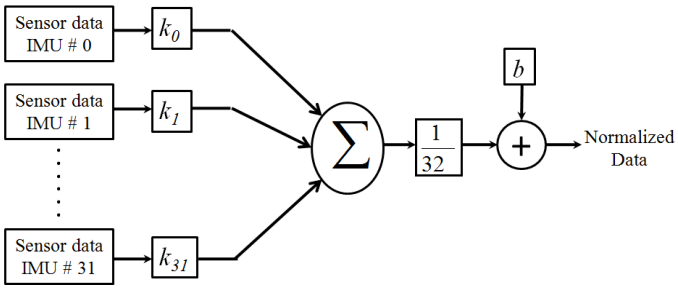


Fig. 4. Data pre-processing flow in Osmium MIMU4444: The sensors data from the IMUs are compensated with a gain factor k_i after calibration. Then, the average is taken and bias b is added to the averaged data to get the normalized data. This pre-processed data, after gyroscope's bias estimation, is used for navigational computation.

where the data sequence is divided in n bins of length τ and a_{τ_i} is the average of each bin.

The results from this method are related to five basic noise terms appropriate for inertial sensor data. These are quantization noise, angle random walk, bias instability, rate random walk, and rate ramp [16]. For MEMS based accelerometers and gyroscopes, the velocity random walk/angle random walk and in-run bias stability are important.

The objective of this study is to see how the white noise and bias instability respond to change in number of IMUs. The increase in number of IMUs will increase both cost as well as size of the device. One has to optimize the number of IMUs based on the noise performance. The Allan variance analysis is done by selecting 1, 2, 4, 8, 16 and 32 IMUs at a time. The IMUs are chosen according to Fig. 2 and the corresponding TABLE I. The normalized fused data, as shown in Fig. 4, are collected for more than 30,000 seconds with data sampling rate of 1 kHz, at room temperature. The orientation of the device during collection of data is such that the resultant acceleration due to gravity is acting along the negative z-axis.

Multiple sets of normalized data are collected but the most stable among them are considered for AD computation. Overlapping AD is computed from the normalized data set. The AD of the normalized acceleration data along the x-axis is compared among the 10 different selections of IMUs as mentioned below. Similarly the same comparisons are done for the normalized acceleration data in the y and z axes as well as for normalized gyroscope data in x, y and z axes. The comparisons are shown in Fig. 5 and Fig. 6.

The slope of the Allan varinace curves for small values of averaging time, is almost $\frac{1}{2}$ for all the combinations of the IMUs, which clearly indicates the presence of random walk. Minima, which is used to determine bias instability values, can be identified in all the curves. The values of the random walk and the in-run bias stability for the accelerometers and the gyroscopes along the x, y and the z axes are measured. It can be observed from TABLE II that both the values are low in comparision to some of the commercial off-the shelf IMUs. The value of acceleration due to gravity g at the place

TABLE I
POSITION OF IMUS IN THE DEVICE

No. of IMUs	Case	IMU#
1	-	0
2	Same side	0,2
	Either side	0,1
4	Same side	0,2,4,6
	Either side	0,1,2,3
8	Same side	0,2,4,6,8,10,12,14
	Either side	0,1,2,3,4,5,6,7
16	Same side	0,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30
	Either side	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
32	-	0 to 31

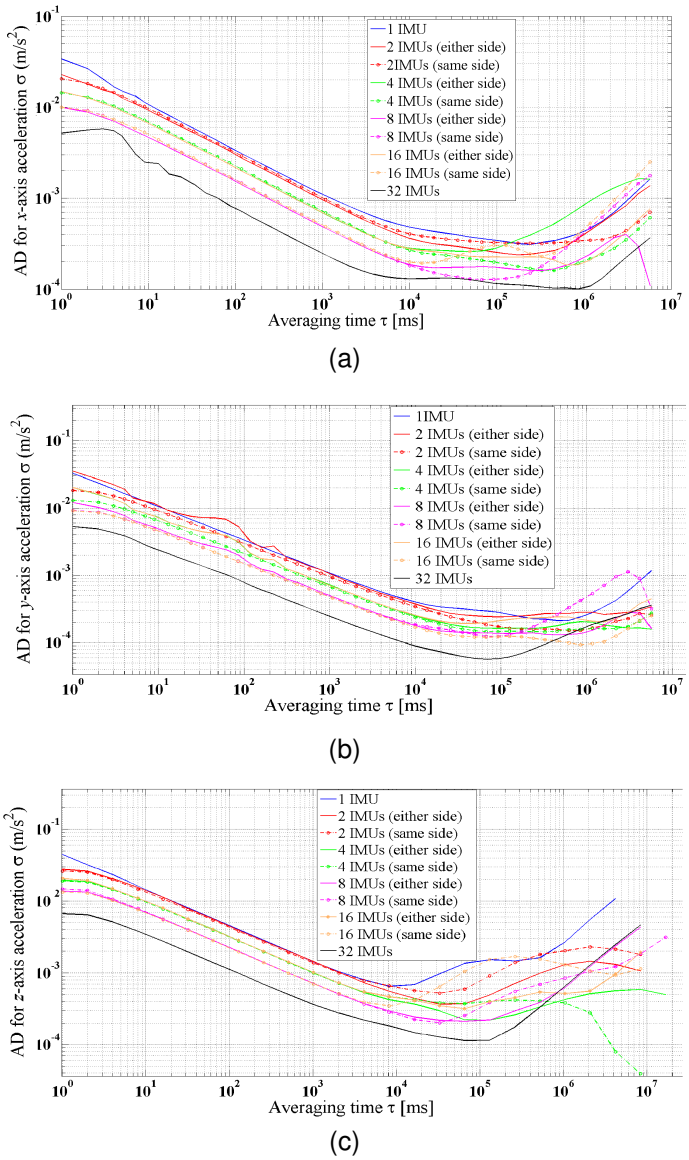
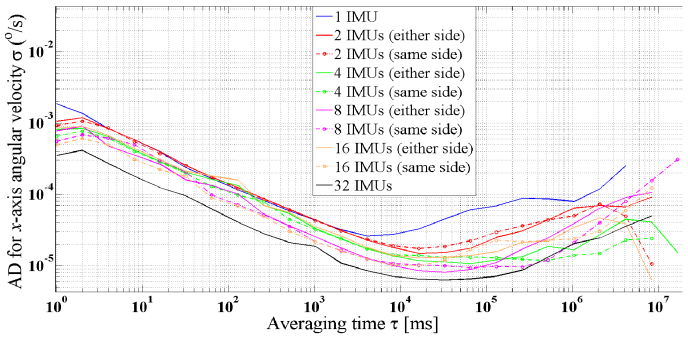
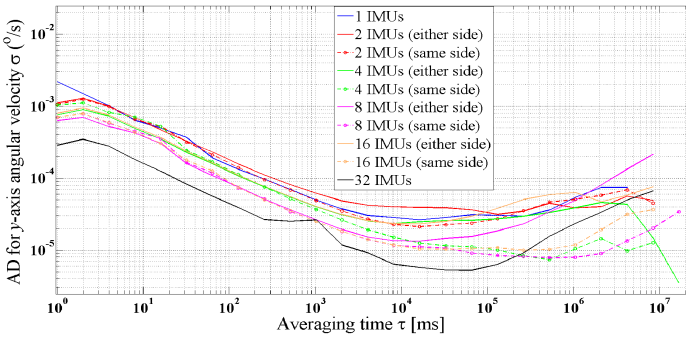


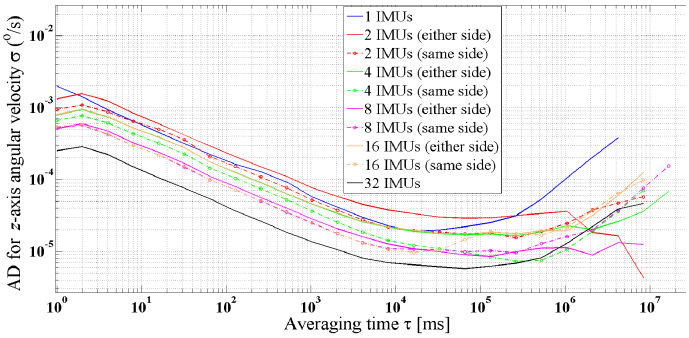
Fig. 5. Allan variance analysis of various combinations of accelerometers in (a) x-axis (b) y-axis (c) z-axis. Noise performance improves with increase in number of IMUs of a multi-IMU system. The combination of IMUs on the same side of the board, exhibits better noise performance.



(a)



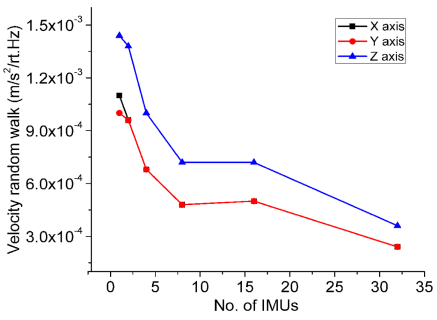
(b)



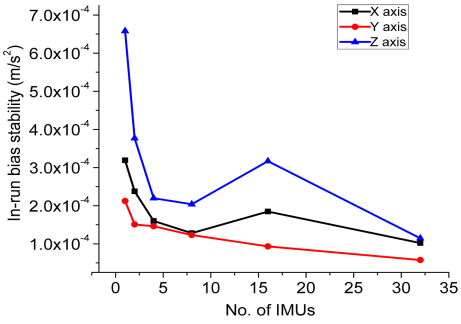
(c)

Fig. 6. Allan variance analysis of various combinations of gyroscopes in (a) *x*-axis (b) *y*-axis (c) *z*-axis. Noise performance improves with increase in number of IMUs of a multi-IMU system. Noise performance along z-axis is better than that of corresponding combinations of accelerometers.

of experiment, is 9.79019 m/s^2. With number of IMUs *N*, the drop in noise level of a multi-IMU system is $\sqrt{N}$, as expected [10]. Based on the measurements, the values of the velocity random walk and the in-run bias stability of the acceleration for different selections of IMUs are plotted in Fig. 7 for all the three orthogonal axes. Similarly the angle random walk and the in-run bias stability is also plotted for the gyroscopes, and shown in Fig. 8. For the cases where the number of IMUs under consideration is 2 or 4 or 8 or 16, the selection of IMUs can be on the same side or on either side. So in these kinds of scenarios the selection with better noise performance considered for the analysis presented in Fig. 7 and Fig. 8. From Fig. 7(a) and Fig. 8(a) it can be observed that *z*-axis shows the worst noise performance while it is comparable for

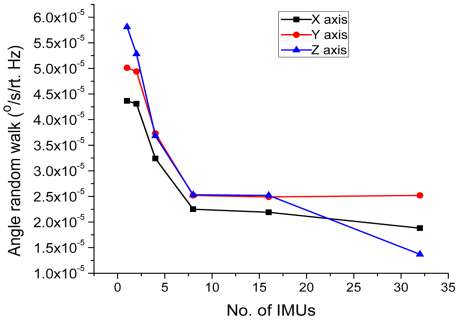


(a)

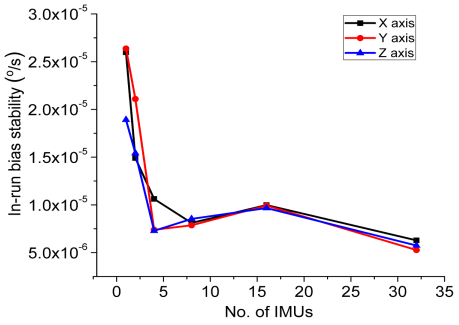


(b)

Fig. 7. The variation of the (a) Velocity random walk (b) In-run bias stability with the number of accelerometers in a multi-IMU system. Noise performance of the resultant accelerometer improves with increase in accelerometers in a multi-IMU system.



(a)



(b)

Fig. 8. The variation of the (a) Angle random walk (b) In-run bias stability with the number of gyroscopes in a multi-IMU system. Noise performance of the resultant gyroscope improves with increase in gyroscopes in a multi-IMU system.

TABLE II
COMPARISION OF MIMU4444 AND DIFFERENT COMMERCIAL OFF-THE-SHELF IMUS.

Device Name	Axis	Acc RW ($m/s/\sqrt{hr}$)	Acc IRBS (mg)	Gyro RW ($^\circ/\sqrt{hr}$)	Gyro IRBS ($^\circ$/hr)
KVH 1775[17]	-	0.070	0.05(1σ)	0.012	0.05(1σ)
MotionPak II [18]	x	0.0166	0.123	0.512	13.6
	y	0.0159	0.116	0.486	11.7
	z	0.0161	0.123	0.489	16.8
STIM300 [19]	x	0.050	0.015	0.114	0.32
	y	0.050	0.015	0.137	0.25
	z	0.052	0.020	0.120	0.46
MIMU4444 ***1-IMU***	x	0.066	0.033	0.395	0.094
	y	0.060	0.022	0.424	0.095
	z	0.086	0.066	0.458	0.068
MIMU4444 ***4-IMUs***	x	0.042	0.021	0.346	0.040
	y	0.042	0.015	0.371	0.055
	z	0.060	0.030	0.387	0.044
MIMU4444 ***32-IMUs***	x	0.014	0.010	0.257	0.023
	y	0.014	0.006	0.300	0.019
	z	0.022	0.012	0.219	0.021

x and y axes. Another important observation is that the noise performance for same number of IMUs is better for the IMUs on the same side. The noise performance for 16 IMUs is an exception, as it does not follow the parabolic nature of the graph.

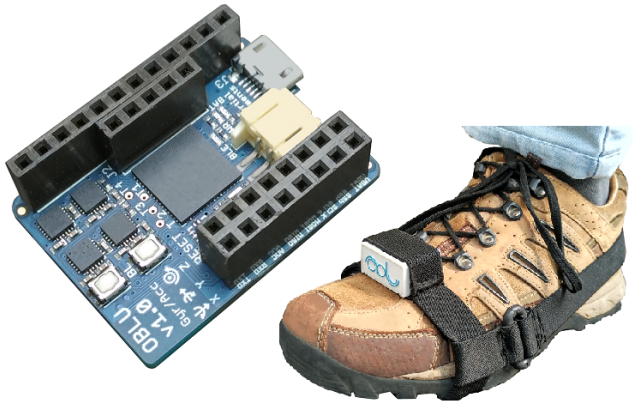
Now comparing the noise performance between different permutations of IMUs, it can be observed that MIMU4444 shows the best noise performance when all the thirty two IMUs are selected. But choosing thirty two IMUs is not a feasible idea, considering that such devices are used as a wearable positioning device. The improvement in noise performance is almost parabolic with the number of IMUs. For a four-IMU system, the noise is almost half compared to a single IMU system, without significant increase in cost and size. The results of the study led to an improved design of a shoe-mounted PDR device as depicted in Fig. 9. For easy reference it will be referred as Oblu (www.oblu.io). This device has an inbuilt Bluetooth for wireless transmission of PDR data.

IV. OPERATING POINTS FOR POWER PERFORMANCE

Shoe-mounted indoor navigation system has various applications where the time duration of usage is quite large. For such applications saving power without any degradation of the performance becomes very critical. Though there are various ways of saving power, the scope of this study is limited to saving power by finding the optimized operating frequencies of system clock and the I2C clock of the controller and IMUs interfacing bus respectively.

A. Varying I2C clock and system clock frequencies

The IMUs send data over I2C bus to the controller. The controller does not offer four inbuilt I2C ports. Therefore, I2C bit banging method is implemented to use the GPIOs



(a) (b)

Fig. 9. Oblu: A four-IMU shoe-mounted ZUPT-aided PDR device. (a) Fully populated Oblu board (b) Battery-operated and encased Oblu mounted on a shoe.

of the controller as I2C ports. This way, four sets of I2C ports are created using controller GPIOs to access all the four IMUs simultaneously. The controller operates on a system clock frequency of 64 MHz. We varied its operating frequency and observed change in power consumption and positioning performance. Three operating frequencies – 64 MHz, 48 MHz and 32 MHz – of the system clock are selected. No other frequency of the system clock was valid because of the limitations in data communication with an interfacing IC. The I2C clock frequency was varied between the maximum allowed 400 kHz and the lowest acceptable frequency limit such that the total time to read and process sensors data does not exceed the sampling time of 1 ms. The IMUs are sampled at fixed frequency of 1 kHz which is also the maximum allowed sampling rate [20]. Therefore, a new set of IMUs data is available for processing at every 1 ms. The controller reads data from the IMUs, performs pre-processing like data formatting, calibration compensation, data fusion etc. followed by navigation computation. This has to be done within 1 ms i.e. before the next set of IMU data is available. The data flow diagram is shown in Fig. 10.

1) Varying I2C clock frequency: First, the system clock frequency is fixed at 64 MHz and then the I2C frequency is slowly reduced from 400 kHz. At each I2C clock frequency, the power consumption is noted. The change in I2C frequency affects the I2C communication speed. Any reduction in the I2C clock frequency increases the time required to read the IMU data by the controller. The I2C frequency is reduced until the time required by the controller to perform data reading, pre-processing and navigational computation just exceeds 1 ms. Study of the I2C SCL (clock) line at a frequency of 400 kHz and 244 kHz is shown in Fig. 11. It is observed from Fig. 11 that the time taken by the controller to read the IMU data are 400 μs and 650 μs at I2C frequency of 400 kHz and 244 kHz respectively. It should be noted that when I2C operates at 244 kHz, the time needed to complete the whole process of positioning for each set of data just exceeds 1 ms. Therefore

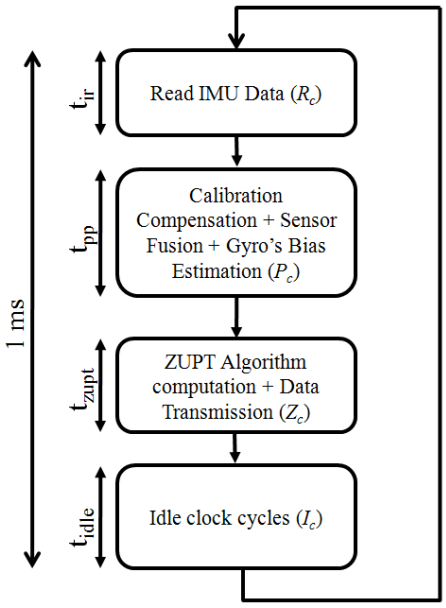


Fig. 10. The dataflow diagram: Time taken by the controller at each step of the entire process is indicated. Here t_{ir} is the time required to read the IMUs data, t_{pp} is the time required for preprocessing and t_{zupt} is the time required to run the ZUPT algorithm by the controller and t_{idle} is the idle system clock cycles that are available after all the computations are done. R_c, P_c, Z_c and I_c are the clock cycles required for reading IMU data, pre-processing the data, executing the navigational algorithm and for being idle respectively.

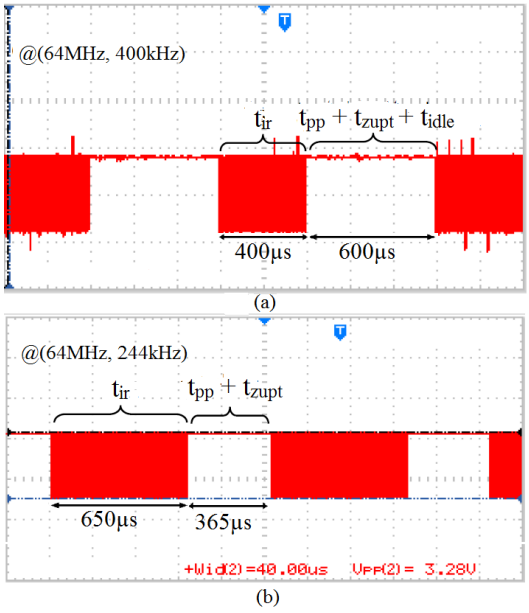


Fig. 11. The IMU data read: The I2C SCL (clock) line indicates the time spent in reading IMU data (t_{ir}), pre-processing (t_{pp}), navigational computation (t_{zupt}) and unused clock cycles (t_{idle}) at I2C clock frequency of (a) 400 kHz (b) 244 kHz

this is the corner case with no unused clock cycle remaining. To find the system clock cycle requirement for preprocessing t_{pp} and for executing the ZUPT algorithm t_{zupt}, the time required to compute normalized (fused) data is observed. The time interval between two successive data outputs is observed at I2C frequencies of 400 kHz and at 200 kHz as shown in Fig. 12(a). At I2C frequency of 200 kHz, The time gap between two successive outputs exceeds the input sampling time of 1 ms.

2) Varying system clock frequency: The above experiments are repeated for the system clock frequency of 48 MHz. It should be noted from TABLE III that at system clock frequency of 48 MHz, the controller takes 405 μs and 520 μs to read the IMUs data at I2C frequency of 400 kHz and 308 kHz respectively. At I2C frequency of 308 kHz, time required for successfully completing all the processes of navigational algorithm just exceeds 1 ms. Similarly, for system clock frequency of 32 MHz the total time required for completing entire process exceeds 1 ms even at 380 kHz, which is the maximum possible I2C frequency by I2C bit banging method at 32 MHz.

B. Clock cycle estimation

From Fig. 11(a) the number of clock cycles to read IMU data R_c at 64 MHz are

$$R_c = 400 \times 10^{-6} \times 64 \times 10^{6} = 25,600$$

It is important to note that IMU data read, at the same I2C frequency, would take different amount of time at different system clock frequencies because of the use of bit banging method for generating I2C over controller's GPIOs.

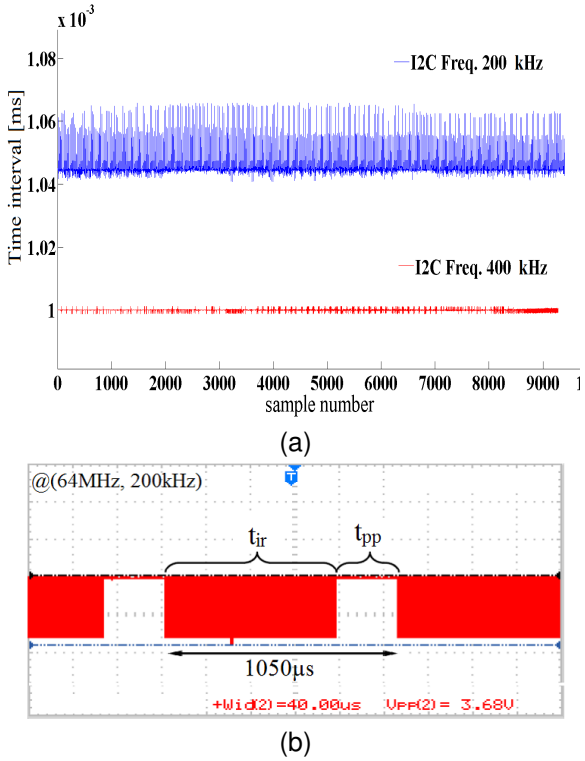


Fig. 12. Estimating pre-processing time: (a) The time interval between two successive pre-processed datasets at I2C speed (a) 400 kHz and 200 kHz (b) The I2C SCL (clock) line at 200 kHz. The time gap between two successive outputs exceeds the input sampling time of 1 ms.

TABLE III
COMPARISION OF POWER CONSUMPTION AT DIFFERENT MODES.

Sys Clk Freq (MHz)	I2C Freq (kHz)	Power consumption (mW)	t_{ir} (μs)	t_{pp}+t_{zupt} (μs)	t_{total} (μs)
64	400	381.85	400	365	765
	370	389.71	430	365	795
	315	391.42	510	365	875
	290	397.84	550	365	915
	244	404.52	650	365	1015
48	400	329.76	405	490	895
	381	335.62	425	490	915
	338	340.71	480	490	970
	308	343.55	520	490	1010
	240	354.35	680	490	1170
32	380	288.90	410	725	1135
	358	291.80	435	725	1160
	324	291.21	480	725	1205
	314	293.09	500	725	1225
	290	295.63	540	725	1265

From Fig. 11(b) it is observed that for system clock frequency of 64 MHz, the total time required for pre-processing (t_{pp}) and navigational computation (t_{zupt}) is ~365 μs. Therefore, the number of system clock cycles (N_c) required are

$$N_c = 365 \times 10^{-6} \times 64 \times 10^6 = 23,360$$

From Fig. 12(b) it can be noted that the time taken for preprocessing the data (t_{pp}) is 260 μs. Therefore the number of clock cycles (P_c) required for pre-processing are

$$P_c = 260 \times 10^{-6} \times 64 \times 10^6 = 16,640$$

It was further studied that among 16,640 clock cycles used for preprocessing, the number of clock cycles used only for calibration compensation are

$$P_{c_{cc}} = 35 \times 10^{-6} \times 64 \times 10^6 = 2,240$$

The number of system clock cycles (Z_c) required for computing ZUPT algorithm are

$$Z_c = (23,360 - 16,640) = 6,720$$

This also implies that the time required to calculate the ZUPT algorithm at 64 MHz system clock frequency, is 105 μs. At I2C clock frequency of 400 kHz, the idle clock cycles (I_c) are

$$I_c = (1000 - 400 - 260 - 105) \times 10^{-6} \times 64 \times 10^6 = 15,040$$

For the best power performance, the number of idle system clock cycles must be minimized. It should be noted that a small number of clock cycles are also utilized for data transmission.

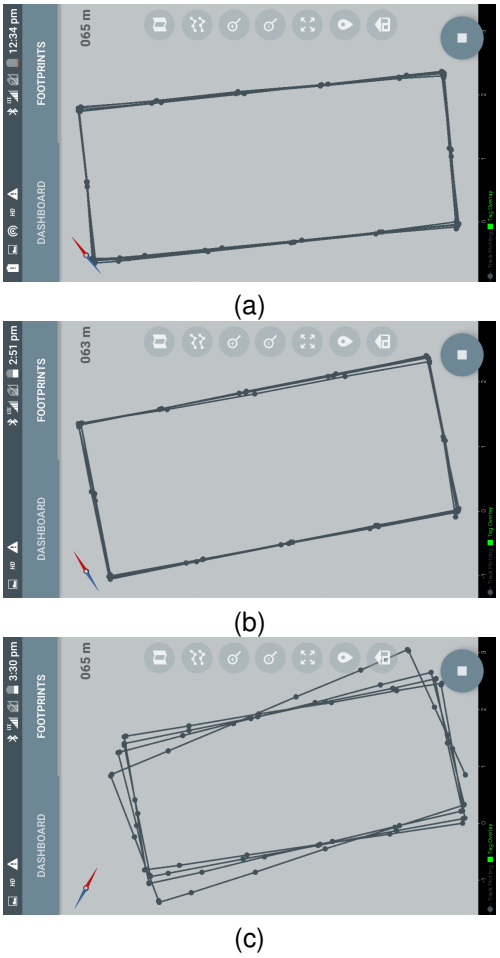


Fig. 13. Tracking results: The screenshot of the footprint as displayed on Android Application - Xoblu when the tracking experiments were performed with system clock frequency of (a) 64 MHz (b) 48 MHz (c) 32 MHz. The results obtained with 64 MHz and 48 MHz system clock frequencies are accurate but performance degradation is observed at system clock frequency of 32 MHz

C. Results

The power numbers are observed at different system clock and I2C clock frequencies, and are presented in TABLE III.

From the TABLE III it can be seen that the power consumption decreases with decrease in system clock frequency. At operating point (48 MHz, 400 kHz), the power consumption is 329.76 mW compared to 381.85 mW at (64 MHz, 400 kHz). It can be seen from TABLE III that almost

$$I_c @\ 48\ Mhz = 105 \times 10^{-6} \times 48 \times 10^6 \approx 5,000$$

clock cycles remain unused even after navigational computation at (48 MHz, 400 kHz). The power consumption at (48 Mhz, 400 kHz) is 13.6% less as compared to (64 MHz, 400 kHz) without any performance degradation.

Positioning experiments using 64 MHz and 48 MHz clock frequencies are performed and shown in Fig. 13(a) and Fig. 13(b). There is hardly any noticeable difference between the two. Multiple tracking experiments were conducted using same device at 64 MHz and 48 MHz system clock frequencies

without any observable difference. The indoor positioning data sample is collected in a rectangular space of size 5.5×2.4 m. The path is traversed four times. The data is collected on the companion Android application, Xoblu [21]. After multiple experiments, it is observed that the performance at 48 MHz consistently matches with that at 64 MHz system clock frequency. But at clock speed of 32 MHz performance degradation is observed as shown in Fig. 13(c).

Interestingly the power consumption increases slightly with reduction in I2C clock frequency. The I2C bus in general is held high, using pull up resistor, except when the data transmission takes place. The current flow through pull-up resistors takes place when the bus is held low. Therefore when the I2C frequency is lowered, the bus remains low for longer duration for the given same set of data transfer, which results in more current flow through the pull- up resistors. As a result the power consumption increases as the I2C clock frequency decreases.

V. Conclusion

This paper presents a design optimization study of multi-IMU array based indoor positioning devices, with respect to noise and power. Superior noise performance of a multi-IMU system is a key advantage as compared to a single IMU based positioning system. Presented Allan variance study on varying number of IMUs of the massive IMU array system, highlights that the noise in general goes on reducing with increase in number of IMUs. However one also has to optimize the cost, size and power efficiency of such systems to make them suitable for wearable applications. The variation in the noise with number of IMUs is parabolic in nature. As expected, for a four-IMU system the noise reduces to almost half as compared to single IMU system, without much increase in size, cost and power consumption of the system. This justifies use of four IMUs in the shoe-mounted inertial navigation system.

We performed the next set of experiments for power optimization, on four-IMU shoe-mounted ZUPT-aided PDR sensor - Oblu. We selected two predominant clocks of the system – the controllers main clock which is responsible for almost all the data processing, and the I2C clock which is used for reading sensors data from four IMUs. Study is performed on three possible controller's clock frequencies – 64 MHz, 48 MHz and 32 MHz. It is noted that ∼14% of the total power can be saved without any compromise in the positioning accuracy, by reducing controller's clock frequency from 64 MHz to 48 MHz. This is due to reduction in idle clock cycles which are present in case of 64 MHz system clock frequency. Clock frequency of 32 MHz is ruled out because of the time required to process input data samples exceeds input sampling time. Though one may study the possibility of operating at 32 MHz with reduced IMUs sampling rate. A slight increase in power consumption is observed with increase in I2C frequency because it takes more time to perform same amount of reading from the IMUs. Therefore, the highest possible I2C clock frequency of 400 kHz becomes the obvious choice.

Evolution of a multi-IMU shoe-mounted inertial navigation system is presented. The superior positioning performance and enhanced power efficiency enable many critical applications requiring infra-free indoor positioning. Innovative products and services around such low-cost PDR sensor would fuel further big innovations, and unleash its mass market applications.

References

[1] J. Rantakokko, J. Rydell, P. Stromback, P. Händel, J. Callmer, D. Tornqvist, F. Gustafsson, M. Jobs, and M. Gruden, *"Accurate and Reliable Soldier and First Responder Indoor Positioning: MultisensorSystems and Cooperative Localization"*, IEEE Wireless Communications, pp.10-18, April2011

[2] J-O. Nilsson, D. Zachariah, I. Skog, and P. Händel, *"Cooperative Localization by Dual Foot-mounted Inertial Sensors and Inter-agent Ranging"*, EURASIP Journal on Advances in Signal Processing, Special Issue on Signal Processing Techniques for Anywhere, Anytime Positioning, 2013, 2013:164.

[3] E. Foxlin, *"Pedestrian Tracking with SHOE-mounted Inertial Sensors"*, IEEE Comput. Graph. Appl., Vol. 25, No. 6, pp. 3846, Nov./Dec. 2005.

[4] S. Godha and G. Lachapelle, *"Foot Mounted Inertial System for Pedestrian Navigation"*, Meas. Sci. Technol., Vol. 19, pp. 19, Jul. 2008.

[5] R. Feliz, E. Zalama, and J. G. Garcia-Bermejo, *"Pedestrian Tracking Using Inertial Sensors"*, J. Phys. Agents, Vol. 3, pp. 3543, Jan. 2009.

[6] I. Skog, P. Händel, J-O. Nilsson and J. Rantakokko, *"Zero-velocity Detection - an Algorithm Evaluation"*, IEEE Transactions on Biomedical Engineering, Vol. 57, No. 11, pp. 2657-2666, Nov. 2010.

[7] J-O. Nilsson, A.K. Gupta, and P. Händel, , *"Foot-mounted Inertial Navigation Made Easy"*, Fifth International Conference on Indoor Positioning and Indoor Navigation (IPIN), Busan, Korea, October 27-30, 2014.

[8] J-O. Nilsson, I. Skog, P. Händel, and K.V.S. Hari, *"Foot-mounted INS for Everybody An Open-source Embedded Implementation"*, Proc. 2012 IEEE/ION Position Location and Navigation Symposium (PLANS), pp. 140-145, Myrtle Beach, SC, USA, April 2326, 2012.

[9] J-O. Nilsson and I. Skog, *"Inertial Sensor Arrays - A Literature Review"*, 2016 European Navigation Conference (ENC), Helsinki, Finland, May 30 June 2, 2016.

[10] J-O. Nilsson, I. Skog, P. Händel, and A. Nehorai, *"Inertial Sensor Arrays, Maximum Likelihood, and Cramr-Rao Bound"*, IEEE Transactions on Signal Processing, Vol. 64, No. 16, Aug. 2016.

[11] I. Skog, J-O. Nilsson and P. Händel *"An Open-Source Multi Inertial Measurement Units (MIMU) Platform,"*, The 1st IEEE Symposium on Inertial Sensors & Systems, Laguna Beach, CA, February 25-26, 2014.

[12] H. M. Schepers, H. F. J. M. Koopman, and P. H. Veltink, *"Ambulatory Assessment of Ankle and Foot Dynamics"*, IEEE Trans. Biomed. Eng., Vol. 54,May,2007.

[13] *MIMU4444 product brief*, http://www.inertialelements.com/documents /mimu4444/MIMU4444_product-brief.pdf

[14] A. Gupta, I. Skog, and P. Händel, *"Long-term Performance Evaluation of a Foot-mounted Pedestrian Navigation Device"*, 12th IEEE India International Conference on Electronics, Energy, Environment, Communication, Computer, Control (INDICON 2015), New Delhi, India, Dec. 17-20, 2015.

[15] M. Marinov and Z. Petrov, *"Allan Variance Analysis on Error Characters of Low-cost MEMS Accelerometer MMA8451Q"*, International Conference of Scientific Paper AFASES 2014, Brasov, Slovak Republic, May 22-24, 2014.

[16] *Allan Variance: Noise Analysis for Gyroscopes*, Freescale Semiconductor Document Number: AN5087 Application Note Rev. 0, 2/2015.

[17] *KVH 1775 specification*, www.kvh.com

[18] N .El-Sheimy, H. Hou, and X. Niu, *"Analysis and Modeling of Inertial Sensors Using Allan Variance"*, IEEE Transactions on Instrumentation and Measurement, Vol. 57, No. 1, Jan. 2008.

[19] *STIM300 Inertia Measurement Unit Datasheet*, www.sensonor.com

[20] *MPU-6500 Product-Specification*, Invensense Document Number: PS-MPU-6500A-01, Revision 1.1.

[21] https://play.google.com/store/apps/details?id=com.inertialelements.xoblu &hl=en