Factor Graph Based Incremental Smoothing in Inertial Navigation Systems

Vadim Indelman*, Stephen Williams*, Michael Kaess† and Frank Dellaert*

*College of Computing, Georgia Institute of Technology, Atlanta, GA 30332, USA
†Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, MA 02139, USA

***Abstract*—This paper describes a new approach for information fusion in inertial navigation systems. In contrast to the commonly used filtering techniques, the proposed approach is based on a non-linear optimization for processing incoming measurements from the inertial measurement unit (IMU) and any other available sensors into a navigation solution. A factor graph formulation is introduced that allows multi-rate, asynchronous, and possibly delayed measurements to be incorporated in a natural way. This method, based on a recently developed incremental smoother, automatically determines the number of states to recompute at each step, effectively acting as an adaptive fixed-lag smoother. This yields an efficient and general framework for information fusion, providing nearly-optimal state estimates. In particular, incoming IMU measurements can be processed in real time regardless to the size of the graph. The proposed method is demonstrated in a simulated environment using IMU, GPS and stereo vision measurements and compared to the optimal solution obtained by a full non-linear batch optimization and to a conventional extended Kalman filter (EKF).**

***Index Terms*—Navigation, information fusion, factor graph, filtering**

I. Introduction

Information fusion in inertial navigation systems is essential for any practical application. In the past few decades, many research efforts have been devoted to developing navigation aiding methods to eliminate, or at least reduce, the inevitable drift in pure inertial navigation solution. These methods typically rely on various sensors, operating at different frequencies, to correct the inertial navigation solution and also to constrain future development of navigation errors by correcting the incoming inertial measurements. While different approaches for navigation-aiding can be found in the literature, arguably the most common approach is based on various variants of the well-known extended Kalman filter (EKF).

Despite the EKF's reputation as a workhorse, incorporating measurements from different asynchronous sensors operating at multiple frequencies is cumbersome, involving redundant computations or ad-hoc approximations. An augmented state is typically used to accommodate measurements from multi-rate sensors, and in order to better incorporate non-linear measurement models, EKF versions such as the iterated EKF and unscented EKF, can be applied. However, current state-of-the-art techniques require updating the whole augmented state vector each time a measurement arrives, which can be expensive if the state vector is large. In practice, this is not always required, since some of the state variables remain unchanged in certain conditions. Moreover, handling delayed measurements either involves a special and non-trivial treatment, or requires a higher resolution of the augmented state vector. Another alternative is to maintain a buffer of past navigation solutions. However, such an approach produces only an approximated solution.

We suggest a *factor graph* formulation for processing all available sensor measurements into a navigation solution. A factor graph [17] is a probabilistic graphical model which, unlike Bayes nets or Markov Random Fields, is represented by a bipartite graph comprising of variable and factor nodes. Variable nodes are associated with system states (or parts thereof), and factors are associated with measurements, and the factor graph encodes the posterior probability of the states over time, given all available measurements. Using factor graphs allows handling different sensors at varying frequencies in a simple and intuitive manner. In particular, since an IMU measurement describes the motion between two consecutive time instances, each IMU measurement introduces a new factor to the factor graph connecting to the navigation state nodes at those two time instances. This factor may also be connected to other nodes used for parameterizing errors in the IMU measurements (such as bias and scale factor terms). These nodes can be added at a lower frequency than the navigation nodes. The factor graph representation also provides plug and play capability, as new sensors are simply additional sources of factors that get added to the graph. Likewise, if a sensor becomes unavailable due to signal loss or sensor fault, the system simply refrains from adding the associated factors; no special procedure or coordination is required.

Calculating the full navigation solution over all states can be performed efficiently using recently-developed incremental smoothing techniques. While this seems as a computationally expensive operation, the incremental approach optimizes only a small part of the nodes in the graph, rendering the proposed approach suitable for high frequency applications. In particular, processing incoming IMU measurements generates a simple factor chain. The incremental approach can operate on this topology extremely efficiently, providing a navigation solution in real time.

In what follows, we next discuss related work on information fusion in inertial navigation systems. Section III introduces the factor graph formulation and presents factors for some of the common sensors in navigation applications. The incremental non-linear optimization is then discussed in Section IV. Simulation results demonstrating the proposed approach are provided in Section V, while Section VI suggests concluding remarks.

II. Related Work

Traditional inertial navigation systems are based on the strapdown mechanism [8], in which IMU measurements are integrated into a navigation solution. Typically, navigation-aiding methods apply filtering approaches for fusing measurements from other available sensors with the inertial solution. The most common are different variants of the EKF. For example, [3] and [31] consider EKF and unscented Kalman filter formulations for integrating GPS with inertial navigation, [25] uses EKF for estimating the pose and the velocity of a spacecraft based on previously mapped landmarks and [32] uses EKF for INS in-flight-alignment.

Another approach for information fusion is calculating the optimal solution based on a non-linear optimization involving all the unknown variables and using all the available measurements. This approach, also known as bundle adjustment (BA), is commonly used in the robotics community for solving the full simultaneous localization and mapping (SLAM) problem [5, 7, 9, 18, 30]. Recently it has been applied for information fusion in inertial navigation systems [20, 24]. Mourikis and Roumeliotis [23, 24] suggested incorporating vision and IMU measurements using a flexible augmented state vector EKF, applying a batch BA for handling loop closures and limiting the effect of linearization errors. In [20], a batch non-linear optimization formulation is suggested for fusing incoming IMU, GPS and visual measurements, recovering the robot's pose, observed landmarks, as well as IMU biases, camera calibration and the camera-to-IMU transformation.

Unlike the batch algorithms suggested in [20, 24], the factor graph formulation allows calculating *incremental* updates to the non-linear optimization solution. Following a recently developed method for incremental smoothing [15, 16], the actual optimization performed with each new incoming measurement, involves solving only a small portion of the variables, leaving the rest unchanged. Only variables that are expected to benefit from the new measurement are updated.

The applied incremental optimization approach is equivalent to an adaptive fixed-lag smoother (as opposed to the commonly used fixed-lag smoother [22]), in which the size of the lag is adjusted according to the actual topology of the graph and the nature of the incoming measurements. In particular, processing IMU measurements can be done extremely efficiently, thereby allowing operation at a high-frequency. Incorporating measurements from different, possibly asynchronous, sensors becomes a matter of connecting the factors defined by these measurements to the appropriate nodes in the factor graph. While delayed measurements require special care in filtering approaches [2, 8, 28, 33], such measurements can be incorporated into factor graph as easily as any other measurement.

A similar approach has been proposed in [27] for pose estimation using a square root information fixed-lag smoother [5], reporting a performance at 20Hz. The current paper formulates the aided inertial navigation problem in terms of a factor graph and applies an improved incremental smoothing technique [16]. The actual optimization uses Lie-algebraic techniques to account for the involved non-linearities, similarly to [1, 10, 27]. It is the author's belief that recent advances in navigation aiding [14, 19] can be formulated within the proposed factor graph framework as well.

Although not at the focus of this paper, loop closures can be handled in the same framework [16] as well.

III. Factor Graph Formulation in Inertial Navigation Systems

Instead of understanding the navigational smoothing problem in conventional terms of matrix operations, we represent it with a graphical model known as a factor graph [17]. A factor graph is a bipartite graph $G = (\mathcal{F}, \mathcal{X}, \mathcal{E})$ with two types of nodes: *factor nodes* $f_i \in \mathcal{F}$ and *variable nodes* $x_j \in \mathcal{X}$. Edges $e_{ij} \in \mathcal{E}$ can exist only between factor nodes and variable nodes, and are present if and only if the factor f_i involves a variable x_j. The factor graph G defines one factorization of the function $f(\mathcal{X})$ as:

$$f(\mathcal{X}) = \prod_i f_i(\mathcal{X}_i),$$

where $\mathcal{X}_i$ is the set of all variables x_j connected by an edge to factor f_i.

In terms of non-linear least squares optimization, each factor encodes an error function that should be minimized by adjusting the estimates of the variables $\mathcal{X}$. The optimal estimate $\hat{\mathcal{X}}$ is the one that minimizes the error of the entire graph $f(\mathcal{X})$.

$$\hat{\mathcal{X}} = \arg\min_{\mathcal{X}} \left(\prod_i f_i(\mathcal{X}_i) \right).$$

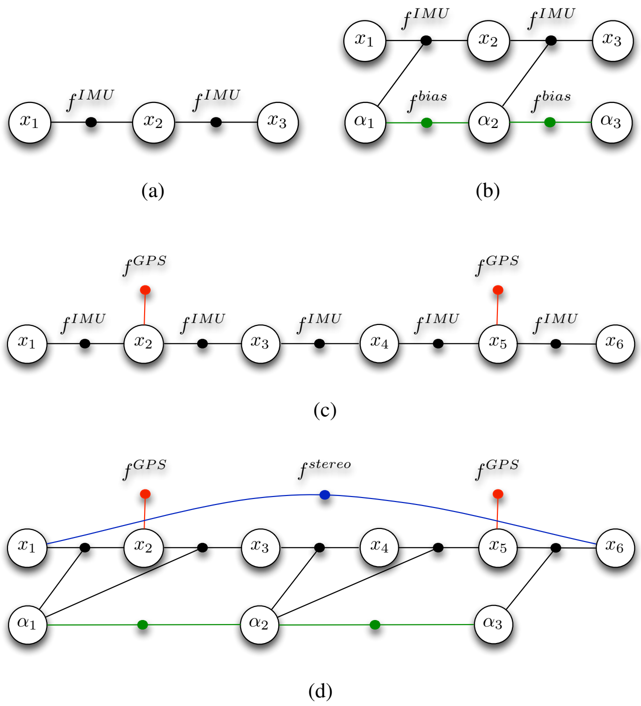
While a factor represents the general concept of an error function that should be minimized, it is common in the navigational literature to design a measurement model $h(\cdot)$ that predicts a sensor measurement given a state estimate. The factor then captures the error between the predicted measurement and actual measurement. This is equivalent to measurement models within the standard EKF context. Assuming a Gaussian noise model, a measurement factor may be written as:

$$f_i(\mathcal{X}_i) = d[h_i(\mathcal{X}_i) - z_i],$$

where $h_i(\mathcal{X}_i)$ is the measurement model as a function of the state variables $\mathcal{X}_i$, z_i is the actual measurement value and the operator $d(.)$ denotes a certain cost function. For a Gaussian noise distribution this is the squared Mahalanobis distance, defined as $d(e) \triangleq e^T \Sigma^{-1} e$, with Σ being the estimated measurement covariance. Process models can be represented using factors in a similar manner [5].

Next, we present factor formulations for different measurement models, covering the most common sensors in typical navigation applications. The considered sensors are IMU, GPS, and monocular and stereo vision.

A. IMU Measurements



(a)

(b)

(c)

(d)

Figure 1. (a) Factor graph representation for two IMU measurements relating between the navigation nodes x_1, x_2 and x_3. (b) Factor graph formulation with navigation and bias nodes. (c) A factor graph with IMU and GPS measurements operating at different rates. (d) Factor graph representation with multi-frequency measurements: high-frequency IMU measurements, lower frequency GPS and stereo vision measurements. Bias nodes are introduced at a lower frequency than navigation nodes.

The time evolution of the navigation state x, representing the robot's position, velocity and orientation, can be conceptually described by the following continuous non-linear differential equations (cf. Appendix) :

$$\dot{x} = h_c\left(x, \alpha, a_m, \omega_m\right), \tag{1}$$

where a_m and ω_m are the acceleration and angular velocity of the robot as measured by the on-board inertial sensors. Also, α is the calculated model of errors in IMU measurements that is used for correcting the incoming IMU measurements. This model of IMU errors is usually estimated in conjunction with the estimation of the navigation state. Linearization of Eq. (1) will produce the well known state space representation with the appropriate Jacobian matrices and a process noise [8], which is assumed to be zero-mean Gaussian noise.

In general, the time propagation of α can be described according to some non-linear model of its own: $\dot{\alpha} = g_c\left(\alpha, x, a_m, \omega_m\right)$. However, a more practical model for α, such as random walk, can be described as

$$\dot{\alpha} = g_c\left(\alpha\right). \tag{2}$$

Throughout this paper, the term bias vector (or bias node) is used when referring to α, although in practice it can represent any model of IMU errors.

A given IMU measurement, $z_k \doteq \left[\begin{array}{cc} a_m^T & \omega_m^T \end{array}\right]^T$, connects the states at two consecuative time instances, denoted by t_k and t_{k+1}. Different numerical schemes, ranging from a simple Euler integration to high-order Runge-Kutta integration, can be applied for solving these equations. However, the factor graph framework allows us to adopt a simple Euler integration prediction function with an associated integration uncertainty. The underlying non-linear optimization will adjust individual state estimates appropriately. The discrete representation of the continuous formulation (1)-(2) is

$$\begin{aligned} x_{k+1} &= h\left(x_k, \alpha_k, z_k\right) \\ \alpha_{k+1} &= g\left(\alpha_k\right). \end{aligned} \tag{3}$$

If desired, the factor graph framework can accommodate more sophisticated schemes as well.

Each of the equations (3), defines a factor connecting related nodes in the factor graph: an IMU factor f^{IMU} connecting the navigation nodes x_k, x_{k+1} and the bias node α_k, and a bias factor f^{bias} connecting the bias nodes α_k and α_{k+1}.

The IMU factor for a given IMU measurement z_k is defined as follows. The IMU measurement z_k and the current estimate of x_k, α_k are used to predict the values for x_{k+1}. The difference between this prediction and the current estimate of x_{k+1} is the error function represented by the factor:

$$f^{IMU}\left(x_{k+1}, x_k, \alpha_k\right) \triangleq d\left(x_{k+1} - h\left(x_k, \alpha_k, z_k\right)\right), \tag{4}$$

Each of the involved unknown vectors in Eq. (4) are represented by variable nodes in the factor graph, while the IMU factor is a factor node connecting these variables. When adding a new node x_{k+1} to the graph, a reasonable initial value for x_{k+1} is required. This can be taken, for example, from the prediction $h\left(x_k, \alpha_k, z_k\right)$. Figure 1a illustrates a factor graph with three navigation nodes and two IMU factors.

In a similar manner, the bias factor associated with the calculated model of IMU errors is given by

$$f^{bias}\left(\alpha_{k+1}, \alpha_k\right) \triangleq d\left(\alpha_{k+1} - g\left(\alpha_k\right)\right) \tag{5}$$

with α_{k+1} and α_k represented in the factor graph as variable nodes. The equivalent factor graph for IMU and bias factors is given in Figure 1b.

Since α does not change significantly over short periods of time, it makes sense to introduce the factor (5) at a slower rate than the IMU factor (4), which is added to the factor graph at IMU frequency. Denoting the most recent bias node by α_l, Eq. (5) changes to $f^{bias}\left(x_{k+1}, x_k, \alpha_l\right) \triangleq d\left(x_{k+1} - h\left(x_k, \alpha_l, z_k\right)\right)$, while the IMU factor formulation (4) is $f^{IMU}\left(x_{k+1}, x_k, \alpha_l\right) \triangleq d\left(x_{k+1} - h\left(x_k, \alpha_l, z_k\right)\right)$. Such a scenario is illustrated in Figure 1d.

B. GPS Measurements

GPS measurements are an example for demonstrating how time-delayed measurements can be easily incorporated into the factor graph. The GPS measurement equation is given by

$$z_k^{GPS} = h^{GPS}(x_l) + n_{GPS},$$

where n_{GPS} is the measurement noise and h^{GPS} is the measurement function, relating between the measurement z_k^{GPS} to the robot's position. In the presence of lever-arm, rotation will be part of the measurement equation as well [8]. GPS measurements are time-delayed since usually $t_k > t_l$. Consequently, the above equation defines a unary factor

$$f^{GPS}(x_l) \triangleq d\left(z_k^{GPS} - h^{GPS}(x_l)\right),$$

which is only connected to the node x_l. GPS pseudo-range measurements can be accommodated in a similar manner as well. Factor graphs with GPS measurements and measurements from other sensors, operating at different rates, are shown in Figures 1c and 1d.

C. Monocular and Stereo Vision Measurements

Incorporating vision sensors can be done on several levels, depending on the actual measurement equations and the assumed setup.

Assuming a monocular pinhole camera, the measurement equation is given by the projection equation [11]

$$p = K\left[\begin{array}{cc} R & t \end{array}\right] X \tag{6}$$

with p and X being the measured pixels and the coordinates of the observed 3D landmark, both given in homogeneous coordinates. The rotation matrix R and the translation vector t represent the transformation between the camera and the global frame (i.e. the global pose), and therefore can be calculated from the current estimate of the navigation node x at the appropriate time instant. When observing known landmarks with a calibrated camera, this equation defines a unary factor on the node x. The much more challenging problem of observing unknown landmarks, also known as full SLAM or BA, requires adding the unknown landmarks into the optimization by including them as nodes in the factor graph and representing the measured pixels by a binary factor connecting between the appropriate navigation and landmark nodes. Alternatively, to avoid including the unknown landmarks into the optimization, one can use multi-view constraints [11, 21], such as two-view [6] and three-view constraints [13], instead of the projection equation (6).

A stereo camera rig is an another commonly used setup. Assuming a known baseline, it is possible to estimate the relative transformation between two stereo frames. This can be formulated as a binary factor connected to navigation nodes at the time instances of these frames. Denoting this relative transformation by T_Δ and the global poses of the two stereo cameras by $T_{k_1}$and T_{k_2}, calculated based on the current values of the navigation nodes x_{k_1} and x_{k_2}, the binary factor becomes

$$f^{stereo}(x_{k_1}, x_{k_2}) \triangleq d\left(T_\Delta - (T_{k_1} - T_{k_2})\right).$$

Since visual measurements are usually obtained at a lower frequency, the size of the fixed-lag is adaptively increased. An illustration of the interaction between the stereo-vision binary factor and other factors is shown in Figure 1d.

IV. Incremental Smoothing

Before presenting our approach for incremental smoothing, which is essential for real-time applications, it is helpful to first discuss a batch optimization.

A. Batch Optimization

We solve the non-linear optimization problem encoded by the factor graph by repeated linearization within a standard Gauss-Newton style non-linear optimizer. Starting from an initial estimate x_0, Gauss-Newton finds an update Δ from the linearized system

$$\arg\min_{\Delta} J(x_0)\Delta - b(x_0), \tag{7}$$

where $J(x_0)$ is the sparse Jacobian matrix at the current linearization point x_0 and $b(x_0) = f(x_0) - z$ is the residual given the measurement z. The Jacobian matrix is equivalent to a linearized version of the factor graph, and its block structure reflects the structure of the factor graph. After solving equation (7), the linearization point is updated to the new estimate $x_0 + \Delta$.

In practice, the error functions for the factors defined in the previous section, such as (4)-(5), as well as all the involved Jacobians, are calculated using the underlying Lie algebra structure of the full 6 degree-of-freedom Euclidean motion in a similar manner to [1, 10, 27, 29].

In the context of an IMU measurement, the Jacobian matrices, calculated for the non-linear optimization process about the current linearization points, are

$$\frac{\partial \epsilon_x}{\partial x_{k+1}}, \frac{\partial \epsilon_x}{\partial x_k}, \frac{\partial \epsilon_x}{\partial \alpha_k}$$

for the IMU factor, and

$$\frac{\partial \epsilon_\alpha}{\partial \alpha_k}, \frac{\partial \epsilon_\alpha}{\partial \alpha_{k+1}}$$

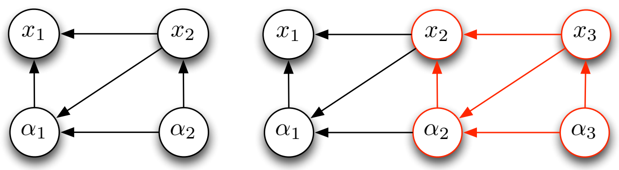
for the bias factor, using Eqs. (4)-(5). In the above equations, $\epsilon_x \triangleq x_{k+1} - h(x_k, \alpha_k, z_k)$ and $\epsilon_\alpha \triangleq \alpha_{k+1} - g(\alpha_k)$.

Each linearized factor graph is solved by variable elimination, equivalent to matrix factorization. Solving for the update Δ requires factoring the Jacobian matrix into an equivalent upper triangular form using techniques such as QR or Cholesky. Within the factor graph framework, these same calculations are performed using variable elimination [12]. A variable ordering is selected and each node is sequentially eliminated from the graph, forming a node in a chordal Bayes net [26]. A Bayes net is a directed, acyclic graph that encodes the conditional densities of each variable. Chordal means that any undirected loop longer than three nodes has a chord or a short cut. This chordal Bayes net is equivalent to the upper triangular matrix that results from matrix factorization, and is used to obtain the update Δ by backsubstitution.

While the elimination order is arbitrary and any order will form an equivalent Bayes net, they may differ significantly in complexity as measured by their number of edges. The selection of the elimination order does affect the *structure* of the Bayes net and the corresponding amount of computation.

A good elimination order will make use of the natural sparsity of the system to produce a small number of edges, while a bad order can yield a fully connected Bayes net. Heuristics do exist, such as approximate minimum degree [4], that approach the optimal ordering for generic problems.

B. Incremental Optimization



(a) (b)

Figure 2. (a) Bayes net for a factor graph containing navigation nodes x_1,x_2 and bias nodes α_1, α_2 . Assumed elimination order: $x_1, \alpha_1,, x_2, \alpha_2$. (b) Adding new IMU and bias factors connecting between the nodes x_2, α_2 and (the new) nodes x_3, α_3, produces a factor graph shown in Figure 1b. Incremental inference allows processing only a small part of the Bayes net, as indicated in red color, regardless to the size of the factor graph.

In the context of the navigation smoothing problem, each new sensor measurement will generate a new factor in the graph. This is equivalent to adding a new row (or block-row in the case of multi-dimensional states) to the measurement Jacobian of the linearized least-squares problem.

The key insight is that optimization can proceed incrementally because most of the calculations are the same as in the previous step and can be reused. When using the algorithm of Section IV-A to recalculate the Bayes net, one can observe that only a part of the Bayes net is modified by the new factor. So instead of recalculating everything, we focus computation on the affected parts of the Bayes net. But what are the affected parts? That question was answered in the incremental smoothing algorithm iSAM2 [16] that we apply here. Informally, a part of the Bayes net is converted back into a factor graph, the new factor is added, and the resulting (small) factor graph is re-eliminated. The eliminated Bayes net is merged with the unchanged parts of the original Bayes net, creating the same result as a batch solution would obtain. To deal with non-linear problems efficiently, iSAM2 combines the incremental updates with selective relinearization.

As long as only sequential IMU measurements are processed, the resulting graph (and Bayes net) will have a chain-like structure, as illustrated in Figures 1a and 1b. The underlying adaptive fixed-lag smoothing in the incremental smoothing approach [16], allows processing each new IMU measurement by optimizing *only 4 nodes* in the factor graph, regardless to the actual size of the graph. A sketch of the different steps in this process is given in Figure 2.

The smoothing lag is automatically adjusted whenever measurements from additional sensors become available. Thus, all the nodes affected by these measurements are optimized, yielding a solution that approaches the optimal solution that would be obtained by a full smoother.

V. Results

The proposed method was examined in a simulated environment using measurements from several sensors operating at different rates. A ground truth trajectory was created, simulating a flight of an aerial vehicle at a 40 m/s velocity and a constant height of 200 meter above mean ground level. The trajectory consists of several segments of straight and level flight and maneuvers, as shown in Figure 3a.

Based on the ground truth trajectory, ideal IMU measurements were generated at 100 Hz, while taking into account Earth's rotation and changes in the gravity vector (cf. Appendix). These measurements were corrupted with a constant bias and a zero-mean Gaussian noise in each axis. Bias terms were drawn from a zero-mean Gaussian distribution with a standard deviation of $\sigma = 10$ mg for the accelerometers and $\sigma = 10$ deg/hr for the gyroscopes. The noise terms were drawn from a zero-mean Gaussian distribution with $\sigma = 100\,\mu g/\sqrt{Hz}$ and $\sigma = 0.001\,deg/\sqrt{hr}$ for the accelerometers and gyroscopes.

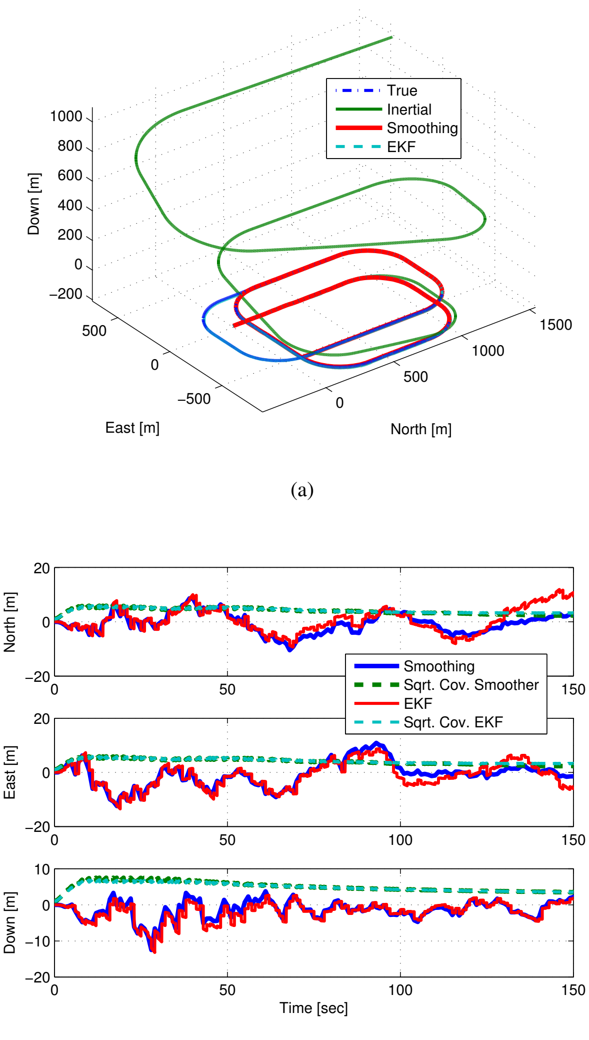
In addition to IMU measurements, GPS measurements were generated at 1 Hz and corrupted with a Gaussian noise with a standard deviation of 10 meters. A stereo camera rig was also assumed, producing relative pose measurements T_Δ (cf. Section III-C) at a 0.5 Hz frequency. These measurements were calculated from observations of unknown landmarks, scattered on the ground with ± 50 meters elevation. Finally, a bank of known landmarks was used to produce short-track visual observations. Each landmark is observed in only a few camera frames ($3-4$ frames). These landmarks were assumed to be known within a 10 meters precision (1σ). A zero-mean Gaussian noise, with $\sigma = 0.5$ pixels, was added to all visual measurements. To avoid double counting, visual observations of known landmarks were *not* used for calculating relative transformations.

The implemented IMU factor is based on a strapdown mechanism [8]. IMU nodes were added to the factor graph at the frequency of IMU measurements, while a random walk process was used for the bias factor, with new bias nodes added to the factor graph every 0.5 seconds.

The next sections compare between the proposed incremental smoother and a conventional EKF in a basic scenario, and demonstrate the performance of the smoother in a multi-rate information fusion scenario. The results are shown in terms of estimation/optimization errors relative to the beginning of trajectory.

A. Incremental Smoothing vs. EKF

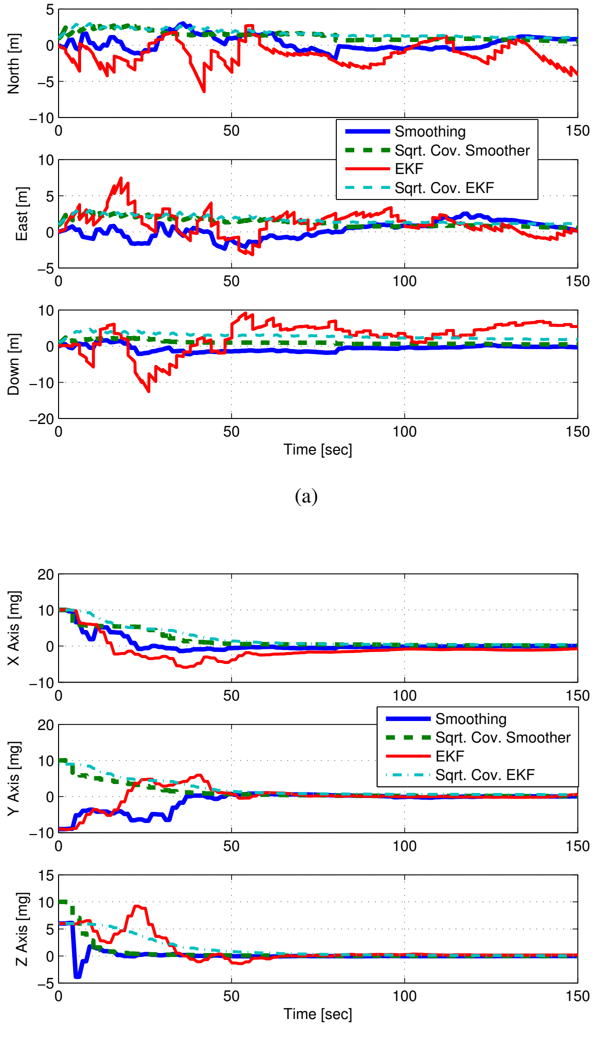
A comparison between the proposed incremental smoothing approach to a conventional EKF is shown in Figures 3-4. As seen in Figure 3b, a similar performance was obtained when using IMU and GPS measurements, because of the implemented simplified GPS measurement equation that is linear with the position terms. In terms of timing, the incremental smoother produces results every 4 msec on average, with a standard deviation of 2.7 msec, thereby allowing operation at IMU frequency. All runs were performed on a single core of an



(a)

(b)

Figure 3. (a) Ground truth and estimated trajectory. (b) Position errors using IMU and GPS measurements. A similar performance is obtained in smoothing and filtering approaches.



(a)

(b)

Figure 4. Comparison between incremental smoothing and an EKF using IMU and visual observations of short-track landmarks, operating at 100 and 0.5 Hz, respectively. Incremental smoothing produces significantly better results: (a) Position errors. (b) Accelerometer bias estimation errors.

Intel i7-2600 processor with a 3.40GHz clock rate and 16GB of RAM memory.

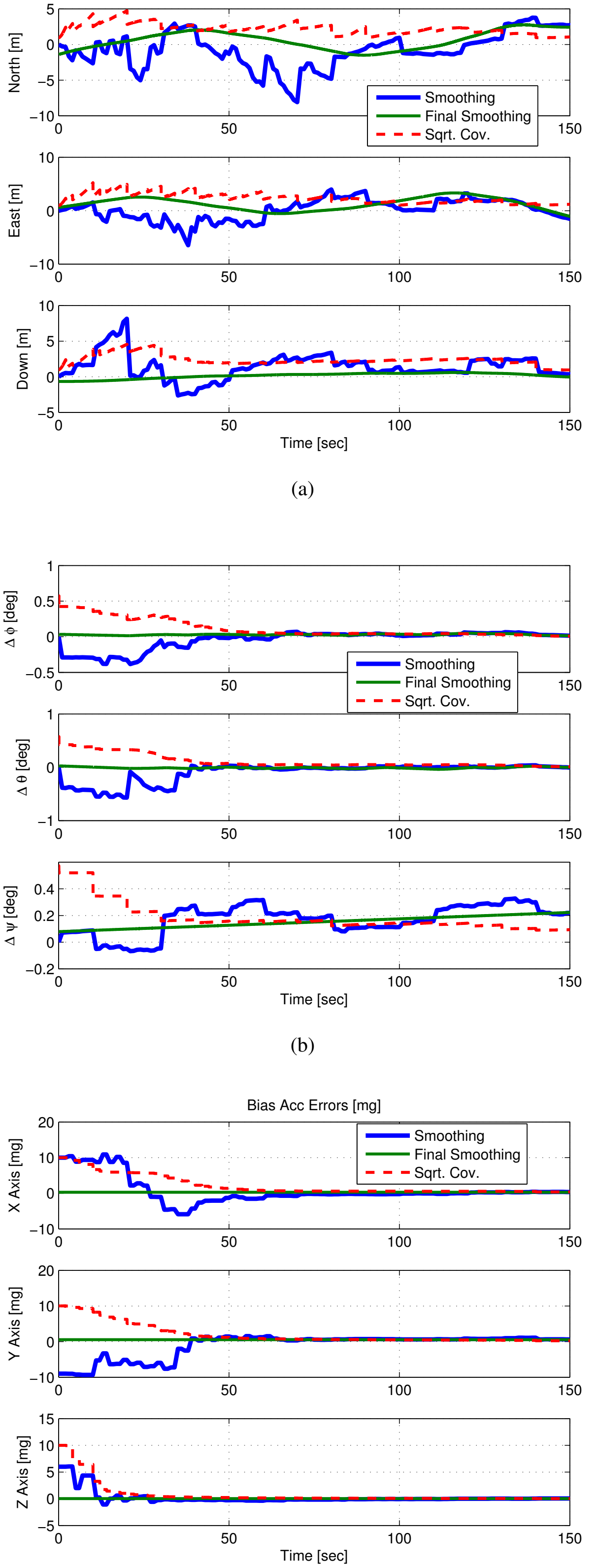
In contrast to GPS measurements, incorporating IMU and visual observations of known landmarks (performed at 0.5 Hz), with the non-linear measurement equation (6), produced much better results in favor of the smoother, as shown in Figure 4. While an improved performance of the filter is expected when applying an iterated EKF, the whole state vector would be estimated each time, in contrast to the proposed approach. As mentioned in Section I, this is an expensive operation when using a large augmented state vector so that measurements from sensors, operating at different frequencies, could be accommodated.

B. *Incremental Smoothing in a Multi-Sensor Scenario*

We now consider a scenario with several sensors operating at different frequencies. Specifically, in addition to the high-frequency IMU measurements, relative pose measurements were incorporated at a 0.5 Hz frequency and visual observations of known landmarks were introduced every 10 seconds.

Estimation errors of position, attitude and accelerometer's bias are given in Figure 5, which shows the incremental smoothing solution obtained at each time step, the estimated square root covariance, and the final smoothing solution of the whole trajectory. The final smoothing solution, equivalent to incremental smoothing at the final time, is as expected, significantly better from the actual (concurrent) smoothing solution and may be useful for various applications such as mapping.

A comparison between the incremental smoothing solution, obtained by the proposed approach, to a batch optimization (cf. Section IV-A), yielded nearly identical results. Actual plots of this comparison are not shown due to space limitation.



(a)

(b)

(c)

Figure 5. Incremental smoothing results in a multi-sensor scenario, using IMU, relative pose measurements and visual observations of short-track known landmarks, operating at $100, 0.5$ and 0.1 Hz, respectively. Position (a), attitude (b) and accelerometer bias (c) estimation errors of the smoother are shown and compared to the estimated square root covariance and to errors in the final smoothing.

In terms of performance, position errors are confined within 8 meters and are much smaller most of the time. Orientation and accelerometer bias is gradually estimated, especially during the vehicle's maneuvers (first maneuver is during $t = 20$ to 50 seconds) that increase the system's observability. Gyroscopes bias was only partially estimated, both in the smoothing and filtering approaches (not shown), due to insufficient precision of the measurements. Overall, the estimated covariance is consistent with the estimation error.

VI. Conclusion

This paper described a factor graph formulation for information fusion problems in inertial navigation systems. The new formulation provides for the incorporation of asynchronous and/or out-of-sequence measurements from multiple sensors, supporting the development of a plug and play capability for registering and un-registering sensors based upon their availability. Incoming measurements were processed into a navigation solution based on a non-linear optimization. Applying a recently developed method for incremental inference, which is equivalent to an adaptive fixed-lag smoother, provided close to an optimal performance with a minimum time delay. The validity of the proposed approach was demonstrated in a simulated environment that included IMU, GPS and stereo-vision measurements, and compared to the full batch non-linear optimization, and to a conventional filter.

Acknowledgments

This work was supported by the all source positioning and navigation (ASPN) program of the Air Force Research Laboratory (AFRL) under contract FA8650-11-C-7137. The views expressed in this work have not been endorsed by the sponsors.

Appendix: INS Kinematic Equations

This appendix provides the inertial navigation equations leading to Eq. (1). Assuming some general frame a and denoting by b and i the body and inertial frames, the time derivative of the velocity, expressed in frame a, is given by [8]:

$$\dot{v}^a = R_b^a f^b + g^a - 2\Omega_{ia}^a v^a - \left(\Omega_{ia}^a \Omega_{ia}^a + \dot{\Omega}_{ia}^a\right) p^a, \quad (8)$$

where R_b^a is a rotation matrix transforming from body frame to frame a, f^b is the specific force measured by the accelerometers, p^a is the position vector, evolving according to

$$\dot{p}^a = v^a \quad (9)$$

and the matrix Ω_{ia}^a is defined as

$$\Omega_{ia}^a = [\omega_{ia}^a]_\times$$

with ω_{ia}^a being the rotational rate of frame a with respect to the inertial frame i, expressed in frame a, and $[.]_x$ is the skew-symmetric operator, defined for any two vectors q_1 and q_2 as $[q_1]_\times q_2 = q_1 \times q_2$. The vector g^a is the position-dependent gravity acceleration.

The time-evolution for the rotation between frame b and a is given by

$$\dot{R}_b^a = R_b^a \Omega_{ab}^b \tag{10}$$

with $\Omega_{ab}^b = \left[\omega_{ab}^b\right]_\times$ and the rotation rate ω_{ab}^b measured by the gyroscopes. The IMU measurement notations f^b and ω_{ab}^b correspond to the notations a_m and ω_m used in Section III-A.

Eqs. (8)-(10) can always be written in the form of Eq. (1), although different expressions are obtained for each choice of the frame a (such as inertial frame, tangent frame, etc.).

References

[1] M. Agrawal. A Lie algebraic approach for consistent pose registration for motion estimation. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2006.

[2] Y. Bar-Shalom. Update with out-of-sequence measurements in tracking: Exact solution. *Signal and Data Processing of Small Targets 2000, Proceedings of SPIE*, 4080:541–556, 2000.

[3] J.L. Crassidis. Sigma-point Kalman filtering for integrated GPS and inertial navigation. *IEEE Trans. Aerosp. Electron. Syst.*, 42 (2):750–756, 2006.

[4] T.A. Davis, J.R. Gilbert, S.I. Larimore, and E.G. Ng. A column approximate minimum degree ordering algorithm. *ACM Trans. Math. Softw.*, 30(3):353–376, 2004. ISSN 0098-3500.

[5] F. Dellaert and M. Kaess. Square Root SAM: Simultaneous localization and mapping via square root information smoothing. *Intl. J. of Robotics Research*, 25(12):1181–1203, Dec 2006.

[6] D. Diel, P. DeBitetto, and S. Teller. Epipolar constraints for vision-aided inertial navigation. In *Proceedings of the IEEE Workshop on Motion and Video Computing (WACV/MOTION'05)*, Washington, DC, USA, 2005.

[7] R.M. Eustice, H. Singh, and J.J. Leonard. Exactly sparse delayed-state filters for view-based SLAM. *IEEE Trans. Robotics*, 22(6):1100–1114, Dec 2006.

[8] J.A. Farrell. *Aided Navigation: GPS with High Rate Sensors*. McGraw-Hill, 2008.

[9] J. Folkesson, P. Jensfelt, and H.I. Christensen. Graphical SLAM using vision and the measurement subspace. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Aug 2005.

[10] Strasdat H., Montiel J. M. M., and Davison A. J. Scale drift-aware large scale monocular SLAM. In *Robotics: Science and Systems (RSS)*, Zaragoza, Spain, June 2010.

[11] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[12] P. Heggernes and P. Matstoms. Finding good column orderings for sparse QR factorization. In *Second SIAM Conference on Sparse Matrices*, 1996.

[13] V. Indelman, P. Gurfil, E. Rivlin, and H. Rotstein. Real-time vision-aided localization and navigation based on three-view geometry. *IEEE Trans. Aerosp. Electron. Syst.*, 48(2), 2012.

[14] E.S. Jones and S. Soatto. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *Intl. J. of Robotics Research*, 30(4), Apr 2011.

[15] M. Kaess, V. Ila, R. Roberts, and F. Dellaert. The Bayes tree: An algorithmic foundation for probabilistic robot mapping. In *Intl. Workshop on the Algorithmic Foundations of Robotics*, Dec 2010.

[16] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *Intl. J. of Robotics Research*, 31:217–236, Feb 2012.

[17] F.R. Kschischang, B.J. Frey, and H-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Inform. Theory*, 47(2), February 2001.

[18] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, pages 333–349, Apr 1997.

[19] T. Lupton and S. Sukkarieh. Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. *IEEE Trans. Robotics*, 28(1):61–76, Feb 2012.

[20] Bryson M., Johnson-Roberson M., and Sukkarieh S. Airborne smoothing and mapping using vision and inertial sensors. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3143–3148, 2009.

[21] Y. Ma, S. Soatto, J. Kosecka, and S.S. Sastry. *An Invitation to 3-D Vision*. Springer, 2004.

[22] P. Maybeck. *Stochastic Models, Estimation and Control*, volume 1. Academic Press, New York, 1979.

[23] A.I. Mourikis and S.I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3565–3572, April 2007.

[24] A.I. Mourikis and S.I. Roumeliotis. A dual-layer estimator architecture for long-term localization. In *Proc. of the Workshop on Visual Localization for Mobile Platforms at CVPR*, Anchorage, Alaska, June 2008.

[25] Trawny N., Mourikis A. I., Roumeliotis S. I., Johnson A. E., and Montgomery J. F. Vision-aided inertial navigation for pin-point landing using observations of mapped landmarks: Research articles. *J. of Field Robotics*, 24(5):357–378, May 2007.

[26] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[27] A. Ranganathan, M. Kaess, and F. Dellaert. Fast 3D pose estimation with out-of-sequence measurements. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 2486–2493, San Diego, CA, Oct 2007.

[28] X. Shen, E. Son, Y. Zhu, and Y. Luo. Globally optimal distributed Kalman fusion with local out-of-sequence-measurement updates. *IEEE Transactions on Automatic Control*, 54(8):1928–1934, Aug 2009.

[29] C.J. Taylor and D.J. Kriegman. Minimization on the Lie group SO(3) and related manifolds. Technical Report 9405, Yale University, New Haven, CT, April 1994.

[30] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT press, Cambridge, MA, 2005.

[31] R Van Der Merwe, E Wan, and Sj Julier. Sigma-point Kalman filters for nonlinear estimation and sensor fusion: Applications to integrated navigation. *AIAA Guidance Navigation and Control Conference and Exhibit*, pages 1735–1764, 2006.

[32] Kong X., Nebot E. M., and Durrant-Whyte H. Development of a nonlinear psi-angle model for large misalignment errors and its application in INS alignment and calibration. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1430–1435, 1999.

[33] S. Zahng and Y. Bar-Shalom. Optimal update with multiple out-of-sequence measurements. In *Proc. of the SPIE, Signal Processing, Sensor Fusion, and Target Recognition XX*, May 2011.