Graph-Based Cooperative Navigation Using Three-View Constraints: Method Validation

Vadim Indelman*, Pini Gurfil†, Ehud Rivlin‡, Hector Rotstein§

*College of Computing, Georgia Institute of Technology, Atlanta, GA 30332, USA
†Faculty of Aerospace Engineering, Technion - Israel Institute of Technology, Haifa 32000, Israel
‡Department of Computer Science, Technion - Israel Institute of Technology, Haifa 32000, Israel
§RAFAEL - Advanced Defense Systems Limited, Israel

***Abstract*—One of the hard issues that arises in distributed navigation is keeping an up-to-date and consistent estimation of the dependency between the solutions computed by each one of the involved agents. This issue is critical for the consistent information fusion in distributed cooperative navigation and was recently tackled using a graph-based approach for the on-demand calculation of cross-covariance terms. In particular, the approach was applied to a method for visual aided, distributed cooperative navigation based on three-view geometry constraints, in which a measurement is formulated whenever the same scene is observed by several robots, not necessarily at the same time.**

The purpose of this paper is twofold. First, the claim that on-demand calculation of cross-covariance terms in three-view-based cooperative navigation is further substantiated, and the difficulties with other existing techniques are emphasized. Second, the efficiency of using the on-demand calculations is validated by comparing the results to those obtained by assuming the three-view multi-robot measurements schedule is known a priori. In this latter method, the required cross-covariance terms are calculated using a fixed-lag centralized smoother. The comparison clearly shows the advantages of using the on-demand scheme.

I. Introduction

The widespread availability of inexpensive inertial and other navigation sensors, cameras and computational power, and communication channels has awaken the interest in investigating cooperative navigation during the last decade. Indeed, suppose that a collection of robots must perform a coordinate task or simply share the same working area. Suppose further that these robots can compute an estimate of their pose with respect to the environment, can acquire images of their working area and can share information with other robots. It is then of interest to investigate ways of computing solutions for the *cooperative navigation* (CN) problem, namely, ways of improving the navigation solution of each one of the agents involved based on sharing the information acquired by all of them. Clearly, information sharing requires keeping track of the correlation between the navigation solutions of each robots, and doing this efficiently is critical for solving the CN problem.

Regardless of the method applied for CN [18], [11], [14], [19], [2], [13], [15], [17], [10], the navigation data involved in the measurement is obtained from different robots. In the general case, these sources of information may be statistically dependent. Ignoring this correlation may result in inconsistent and over-confident estimations [1].

Several approaches have been proposed for coping with the correlation terms in multi-robot (MR) systems, assuming relative pose measurements. In [18], an augmented covariance matrix, composed of covariance and cross-covariance matrices relating all the robots in the group, was maintained in a distributed manner. In [3], this approach was applied to cooperative mapping and localization. In this case, the augmented covariance matrix also contains parameters that represent the landmarks observed by each robot in the group. Howard et al. [5] suggested a method that allows to avoid correlated updates in certain situations. Similarly, in [1], the cross-covariance terms were not explicitly estimated. Instead, the authors proposed to maintain a bank of filters, tracking the origins of measurements and preventing a measurement to be used more than once. References [16] and [12] studied the filter inconsistency when correlated measurement sequences are used.

The current paper considers a recently-proposed approach [6], [8] for an explicit, on-demand, computation of correlation terms, that are required for a consistent information fusion. A *general* MR measurement model for CN is assumed. This model relates between the

navigation data from any number of robots and the actual readings of the onboard sensors of these robots, which are *not* necessarily taken at the same time. In the general case, all the involved sources of information may be correlated. Assuming this MR measurement model, in addition to the a priori unknown identities of the robots that generate an MR measurement, the time instances that appear in the measurement equation are unknown as well. Hence, any method that is based on maintaining the correlation terms impractical. Instead, using graph-based technique, the method developed in [6], [8] explicitly calculates the required correlation terms based on the history of all MR measurements performed thus far.

This paper aims to validate the above method for a specific MR measurement model that is based on the recently-developed three-view geometry constraints [6]. These constraints were applied for vision-aided navigation of a single vehicle [9], and were extended for cooperative navigation [10].

Consequently, the purpose of this paper is twofold. First, the claim that on-demand calculation of cross-covariance terms in three-view-based cooperative navigation is further substantiated, and the difficulties with other existing techniques are emphasized. Second, the efficiency of using the on-demand calculations is validated by comparing the results to those obtained by assuming the three-view multi-robot measurements schedule is known a priori. In this latter method, the required cross-covariance terms are calculated using a fixed-lag centralized smoother. The comparison clearly shows the advantages of using the on-demand scheme.

II. Problem Setup

Consider a group of N robots that perform tasks on a partially or totally shared region. It is assumed that each robot is equipped with navigation sensors and hence is capable of calculating its own navigation solution, and that this information can be shared via a communication channel with some of the other robots. Specifically, assume that the robots have an *inertial measurement unit* (IMU) so that, after appropriate initialization, a full navigation solution comprising position, velocity and angular orientation can be estimated. Denote by $\mathbf{x}_i$ and $\mathbf{x}_i^t$ the calculated and the (unknown) true navigation solutions of the ith robot, respectively, and let $\mathbf{u}_i$ represent the measurements of the robot's inertial navigation sensors. The measurements of the inertial sensors are invariably contaminated by errors; some of these errors - typically accelerometer bias and gyro drifts - can be modeled by an unknown vector of parameters $\boldsymbol{\alpha}_i^t$ while other errors are unmodeled and are taken into account when constructing and tunning the estimation problem. An estimated vecotr $\boldsymbol{\alpha}$ can be calculated and used for correcting the vector of measurements $\mathbf{u}$ in a typical compensated inertial navigation mechanization.

Let

$$\boldsymbol{\zeta}_i(t_k) \doteq \begin{bmatrix} \mathbf{x}_i(t_k) \\ \boldsymbol{\alpha}_i(t_k) \end{bmatrix} \quad , \quad \boldsymbol{\zeta}_i^t(t_k) \doteq \begin{bmatrix} \mathbf{x}_i^t(t_k) \\ \boldsymbol{\alpha}_i^t(t_k) \end{bmatrix}$$

and $\mathcal{N} \doteq \{1, \ldots, N\}$. Then

$$\boldsymbol{\zeta}_i(t_{k+1}) = \mathbf{f}(\boldsymbol{\zeta}_i(t_k), \mathbf{u}_i(t_k)) \quad , \quad i \in \mathcal{N} \tag{1}$$

Define the navigation error state vector:

$$\mathbf{X}_i(t) \doteq \begin{bmatrix} \mathbf{x}_i(t) - \mathbf{x}_i^t(t) \\ \boldsymbol{\alpha}_i(t) - \boldsymbol{\alpha}_i^t(t) \end{bmatrix} \equiv \boldsymbol{\zeta}_i(t) - \boldsymbol{\zeta}_i^t(t) \tag{2}$$

The linear approximation of Eq. (1) yields:

$$\dot{\mathbf{X}}_i(t) = \Phi^i(t)\mathbf{X}_i(t) + \boldsymbol{\omega}^i(t) \quad , \quad i \in \mathcal{N} \tag{3}$$

where Φ^i is a *process dynamics* matrix and $\boldsymbol{\omega}^i$ is a vector process noise, assumed to be white and zero-mean Gaussian. In practice, $\boldsymbol{\omega}^i$ must be tuned to capture the effects of under-modeling and linearization. For implementation, the differential equation above is replaced by the discrete-time model:

$$\mathbf{X}_i(t_b) = \Phi^i_{t_a \to t_b}\mathbf{X}_i(t_a) + \boldsymbol{\omega}^i_{t_a \to t_b} \quad , \quad i \in \mathcal{N} \tag{4}$$

where $\Phi^i_{t_a \to t_b}$ denotes the state transition matrix between any two time instances t_a and t_b, $t_b > t_a$, and $\boldsymbol{\omega}^i_{t_a \to t_b}$ is an equivalent discrete-time process noise. In addition to the inertial sensors, each robot is equipped with additional onboard exogenous navigation sensors. Of particular interest to this paper, the exogenous sensor will be a camera that is mounted on the robot and acquires images of the environment. Generically, the measurements of the exogenous sensors of the jth robot at some time instant t_a are denoted by $\mathbf{y}_j(t_a)$. These measurements are again corrupted by errors modeled as a vector of Gaussian white noise $\mathbf{v}_j(t_a)$. Let $\mathbf{y}_j^t(t_a) \doteq \mathbf{y}_j(t_a) - \mathbf{v}_j(t_a)$.

Collect the navigation solution $\mathbf{x}_j(t_i)$ and a parametrization of the inertial sensors errors $\boldsymbol{\alpha}_j(t_i)$ onto an intrinsic vector of estimated data $\boldsymbol{\zeta}_j(t_i)$. Then this vector together with the onboard sensor readings $\mathbf{y}_j(t_i)$ of the jth robot at time t_i, can be implicitly related by:

$$\mathbf{z} = \mathbf{h}(\{\boldsymbol{\zeta}_j(t_i), \mathbf{y}_j(t_i)\}_{i=1}^r) \quad , \quad j \in \mathcal{N} \tag{5}$$

where $\mathbf{z}$, represents the *residual measurement*. The parameter r in Eq. (5) denotes the total number of implicit relationships $(\boldsymbol{\zeta}_j(t_i), \mathbf{y}_j(t_i))$ included in $\mathbf{z}$. See [6] for further details.

For notation simplicity, it is assumed below that the robots originating $\mathbf{z}$ can be numbered as $1, \ldots, r$; cases in which a relationships for more than one time instant are included in the set are treated as if this information was provided by different robots. Thus, the residual measurement $\mathbf{z}$ can be written as:

$$\mathbf{z}(t) = \mathbf{h}(\{\boldsymbol{\zeta}_i(t_i), \mathbf{y}_i(t_i)\}_{i=1}^{r}) \tag{6}$$

Linearizing Eq. (6) about $\boldsymbol{\zeta}_i^t(t_k)$ and $\mathbf{y}_i^t(t_i)$ gives

$$\mathbf{z}(t) \approx \sum_{i=1}^{r} H_i(t_i)\mathbf{X}_i(t_i) + D_i(t_i)\mathbf{v}_i(t_i) \tag{7}$$

where

$$H_i(t_i) = \nabla_{\boldsymbol{\zeta}_i^t(t_i)}\mathbf{h} \quad , \quad D_i(t_i) = \nabla_{\mathbf{y}_i^t(t_i)}\mathbf{h} \tag{8}$$

since $\boldsymbol{\zeta}_i^t(t_k)$ and $\mathbf{y}_i^t(t_i)$ are unknown, the Jacobian matrices are approximated by

$$H_i(t_i) = \nabla_{\boldsymbol{\zeta}_i(t_i)}\mathbf{h} \quad , \quad D_i(t_i) = \nabla_{\mathbf{y}_i(t_i)}\mathbf{h} \tag{9}$$

The linearized model can now in principle be used in a Kalman Filter implementation. However this requires tracking the cross-covariance terms relating the different state vectors that appear in the measurement model (7). Denoting by $\tilde{\mathbf{X}}$ the estimation error of $\mathbf{X}$, the required cross-covariance terms are $E[\tilde{\mathbf{X}}_i(t_i)\tilde{\mathbf{X}}_j^T(t_j)]$ with $i, j = 1 \ldots r, i \neq j$. If these terms are known, a consistent measurement update can be employed.

Since tracking cross-covariances is a complex task, it is tempting to follow the approach in [18], wherein an augmented covariance matrix is maintained, consisting of the covariance matrices of all the robots in the group and of all cross-covariance matrices relating any pair of robots. However, this approach can only be applied when the measurement model involves concurrent information from different robots, as indeed is the case with relative pose measurements.

While the above formulation is valid for a general multi-robot (MR) measurement function $\mathbf{h}$, this paper focuses on a specific MR measurement model dictated by applying the recently-developed three-view geometry constraints [6], [9]. A method for cross-covariance calculation for a general MR measurement model can be found in [6], [8].

III. Three-View Geometry Constraints

Consider a set of three views observing a common overlapping area. These views can be captured by different robots, not necessarily at the same time. Assuming the matching features among these views are given (i.e., the correspondence problem is solved), the three-view geometry constraints can be formulated as follows [6]:

$$\bar{\mathbf{q}}_1^T(\bar{\mathbf{t}}_{1\to2} \times \bar{\mathbf{q}}_2) = 0 \tag{10}$$

$$\bar{\mathbf{q}}_2^T(\bar{\mathbf{t}}_{2\to3} \times \bar{\mathbf{q}}_3) = 0 \tag{11}$$

$$(\bar{\mathbf{q}}_2 \times \bar{\mathbf{q}}_1) \cdot (\bar{\mathbf{q}}_3 \times \bar{\mathbf{t}}_{2\to3}) = (\bar{\mathbf{q}}_1 \times \bar{\mathbf{t}}_{1\to2}) \cdot (\bar{\mathbf{q}}_3 \times \bar{\mathbf{q}}_2) \tag{12}$$

where $\mathbf{q}_1, \mathbf{q}_2$ and $\mathbf{q}_3$ are the line-of-sight vectors of the corresponding pixels in the three views, and $\mathbf{t}_{1\to2}$ and $\mathbf{t}_{2\to3}$ are the translation vectors between these views. The notation $\bar{\mathbf{a}}$ denotes the ideal value of some vector $\mathbf{a}$. The line-of-sight vector $\mathbf{q}$ for a given pixel $\mathbf{p}$ (in a homogeneous representation) can be calculated in the camera system as

$$\mathbf{q} = K^{-1}\mathbf{p}.$$

with K being the camera calibration matrix. Appropriate rotation matrices should be used for expressing all the vectors in Eqs. (10)-(12) in the same coordinate system, which can be chosen arbitrary.

The three-view constraints consist of the well-known epipolar geometry constraints (10)-(11) [4], and of an additional constraint (12) that allows to maintain a consistent scale between the three given views. As proven in [6], [9], these constraints are necessary and sufficient conditions for a general scene observed by the given three views.

In practice, the three-view constraints (10)-(12) will not be satisfied due to image noise and navigation errors. Therefore, there will be some residual error $\mathbf{z} \doteq \begin{bmatrix} z_1 & z_2 & z_3 \end{bmatrix}^T$, defined as

$$z_1 \doteq \mathbf{q}_1^T(\mathbf{t}_{1\to2} \times \mathbf{q}_2)$$

$$z_2 \doteq \mathbf{q}_2^T(\mathbf{t}_{2\to3} \times \mathbf{q}_3)$$

$$z_3 \doteq (\mathbf{q}_2 \times \mathbf{q}_1) \cdot (\mathbf{q}_3 \times \mathbf{t}_{2\to3}) - (\mathbf{q}_1 \times \mathbf{t}_{1\to2}) \cdot (\mathbf{q}_3 \times \mathbf{q}_2)$$

The above three equations form a non-linear function $\mathbf{h}$ of the navigation state of the different robots at appropriate time instances, and of the image measurements. Therefore, these equations can be written in the form of Eq. (6).

IV. Concept of Graph-Based CN Using Three-View Constraints

This section presents the concept of the method for graph-based cross-covariance calculation [6]. To better explain the intuition, we focus on a simple example of multi-robot CN using three-view measurements. The considered scenario is shown in Figure 1: A formation of two robots traveling along a straight line in north

Table I
TWO-ROBOT SCENARIO: INITIAL NAVIGATION ERRORS AND IMU ERRORS

Parameter Description	Robot I	Robot II	Units
Initial position error (1σ)	$(10,10,10)^T$	$(100,100,100)^T$	m
Initial velocity error (1σ)	$(0.1,0.1,0.1)^T$	$(0.3,0.3,0.3)^T$	m/s
Initial attitude error (1σ)	$(0.1,0.1,0.1)^T$	$(0.1,0.1,0.1)^T$	deg
IMU drift (1σ)	$(1,1,1)^T$	$(10,10,10)^T$	deg/hr
IMU bias (1σ)	$(1,1,1)^T$	$(10,10,10)^T$	mg

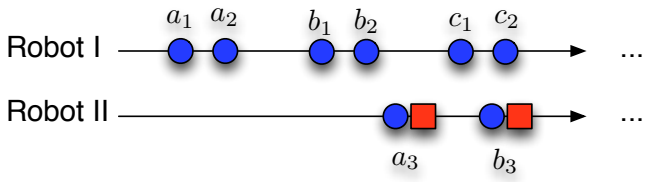

Figure 1. Three-view measurements schedule in the considered two-robot scenario.

direction is assumed. Each robot is calculating its own navigation solution based on the on-board inertial sensors. The first robot has better inertial sensors and its initial navigation solution is assumed to be of a better accuracy, as described in Table I.

While the robots perform the same trajectory, the first robot is 2 seconds ahead of the second robot. Since the robots perform the same trajectory, they observe similar areas on the ground, and therefore it is possible to use three-view constraints, as discussed in previous sections, for reducing for navigation aiding.

Thus, the first robot transmits images to the second robot. Assume that at some point of time, robot II identifies that two of the images, transmitted by robot I, overlap with its currently-captured image. In order to update its navigation system with the three-view constraints formulated based on these images, the appropriate correlation terms should be calculated.

The cross-covariance terms are computed in the following recursive way. Referring to Figure 1, assume the first measurement, comprised of a_1, a_2 and a_3 was carried out, and that the a priori and a posteriori covariance and cross-covariance terms are known. Now, it is required to calculate the cross-covariance terms $E[(\tilde{\mathbf{X}}^-_{b_3})(\tilde{\mathbf{X}}^-_{b_2})^T], E[(\tilde{\mathbf{X}}^-_{b_3})(\tilde{\mathbf{X}}^-_{b_1})^T]$ and $E[(\tilde{\mathbf{X}}^-_{b_2})(\tilde{\mathbf{X}}^-_{b_1})^T]$ for performing the second three-view update.

At this point, we assume the following basic navigation error state vector:

$$\mathbf{X} = \begin{bmatrix}\Delta\mathbf{P}^T & \Delta\mathbf{V}^T & \Delta\mathbf{\Psi}^T & \mathbf{d}^T & \mathbf{b}^T\end{bmatrix}^T \tag{13}$$

where $\Delta\mathbf{P} \in \mathbb{R}^3, \Delta\mathbf{V} \in \mathbb{R}^3, \Delta\mathbf{\Psi} = (\Delta\phi, \Delta\theta, \Delta\psi)^T \in [0, 2\pi) \times [0, \pi) \times [0, 2\pi)$ are the position, velocity and attitude errors, respectively, and $\mathbf{d}$ and $\mathbf{b}$ are the residual gyro drift and accelerometer bias, respectively:

$$\mathbf{d} \doteq \mathbf{d}_{IMU} - \mathbf{d}^t_{IMU} \quad , \quad \mathbf{b} \doteq \mathbf{b}_{IMU} - \mathbf{b}^t_{IMU} \tag{14}$$

with $\mathbf{d}^t_{IMU}, \mathbf{b}^t_{IMU}$ being the unknown true values of $\mathbf{d}_{IMU}, \mathbf{b}_{IMU}$. The position and velocity errors are expressed in the NED system, while $\mathbf{d}$ and $\mathbf{b}$ are given in the body-fixed reference frame.

The following equations can be written for the state propagation:

$$\begin{aligned}\tilde{\mathbf{X}}^-_{b_3} &= \Phi_{a_3\to b_3}\tilde{\mathbf{X}}^+_{a_3} + \boldsymbol{\omega}_{a_3:b_3} \\ \tilde{\mathbf{X}}^-_{b_i} &= \Phi_{a_2\to b_i}\tilde{\mathbf{X}}^-_{a_2} + \boldsymbol{\omega}_{a_2:b_i} \quad , \quad i = 1,2\end{aligned}$$

where $\boldsymbol{\omega}_{i:j}$ is the equivalent process noise between the time instances t_i and t_j of the appropriate vehicle. The cross-covariance terms $E[(\tilde{\mathbf{X}}^-_{b_3})(\tilde{\mathbf{X}}^-_{b_i})^T]$ with $i = 1, 2$, may be calculated as:

$$E\left[\left(\Phi_{a_3\to b_3}\tilde{\mathbf{X}}^+_{a_3} + \boldsymbol{\omega}_{a_3:b_3}\right)\left(\Phi_{a_2\to b_i}\tilde{\mathbf{X}}^-_{a_2} + \boldsymbol{\omega}_{a_2:b_i}\right)^T\right] \tag{15}$$

The a posteriori estimation error $\tilde{\mathbf{X}}^+_{a_3}$ is given by [10]:

$$\begin{aligned}\tilde{\mathbf{X}}^+_{a_3} &= (I - K_{a_3}H_{a_3})\,\tilde{\mathbf{X}}^-_{a_3} - K_{a_3}H_{a_2}\tilde{\mathbf{X}}^-_{a_2} - \\ &- K_{a_3}H_{a_1}\tilde{\mathbf{X}}^-_{a_1} - K_{a_3}D_a\mathbf{v}_a\end{aligned} \tag{16}$$

Since $\boldsymbol{\omega}_{a_2:b_2}$ is statistically independent of $\tilde{\mathbf{X}}^-_{a_3}, \tilde{\mathbf{X}}^-_{a_2}, \tilde{\mathbf{X}}^-_{a_1}$, and since $\boldsymbol{\omega}_{a_3:b_3}$ is statistically independent of $\tilde{\mathbf{X}}^-_{a_2}$ and $\boldsymbol{\omega}_{a_2:b_2}$ (cf. Fig. 1):

$$\begin{aligned}E\left[\tilde{\mathbf{X}}^+_{a_3}\boldsymbol{\omega}^T_{a_2:b_2}\right] &= 0 \\ E\left[\boldsymbol{\omega}_{a_3:b_3}\left(\Phi_{a_2\to b_2}\tilde{\mathbf{X}}^-_{a_2} + \boldsymbol{\omega}_{a_2:b_2}\right)^T\right] &= 0\end{aligned}$$

Denoting $P_{ab} \doteq E[(\tilde{\mathbf{X}}_a)(\tilde{\mathbf{X}}_b)^T]$ and incorporating the above into Eq. (15) yields

$$P^-_{b_3b_2} = \Phi_{a_3\to b_3}\left\{(I-K_{a_3}H_{a_3})\,P^-_{a_3a_2} - \right. \\ \left. -K_{a_3}H_{a_2}P^-_{a_2a_2} - K_{a_3}H_{a_1}P^-_{a_1a_2}\right\}\Phi^T_{a_2\to b_2} \quad (17)$$

where the expectation terms involving a_3, a_2, a_1 are known (from previous update). In a similar manner we get

$$P^-_{b_3b_1} = \Phi_{a_3\to b_3}\left\{(I-K_{a_3}H_{a_3})\,P^-_{a_3a_2} - \right. \\ \left. -K_{a_3}H_{a_2}P^-_{a_2a_2} - K_{a_3}H_{a_1}P^-_{a_1a_2}\right\}\Phi^T_{a_2\to b_1} \quad (18)$$

while $P^-_{b_2b_1} \doteq E[(\tilde{\mathbf{X}}^-_{b_2})(\tilde{\mathbf{X}}^-_{b_1})^T]$ is given by

$$P^-_{b_2b_1} = \Phi_{b_1\to b_2}P^-_{b_1b_1} \quad (19)$$

The above example demonstrated calculation of cross-covariance terms in a simple scenario. The graph-based approach developed in [6] follows a similar concept, while considering a general MR measurement model and a general schedule of measurements.

V. Fixed-Lag Centralized Smoother Approach

In this section the proposed method for cross-covariance calculation is compared to a centralized approach, in which a single state vector comprising information from all the robots in the group is maintained. The considered scenario consists of two robots that share visual and navigation information and perform a straight and level trajectory as in Section IV.

Since the three-view MR measurement model involves different time instances, a fixed-lag centralized smoothing was applied with a state vector and a covariance matrix defined as:

$$\breve{\mathbf{X}}(t_k) \doteq \left[\begin{array}{cc} \breve{\mathbf{X}}^T_{\mathrm{I}}(t_k) & \breve{\mathbf{X}}^T_{\mathrm{II}}(t_k) \end{array}\right]^T \quad (20)$$

$$\breve{P}(t_k) \doteq E\left[\breve{\mathbf{X}}(t_k)\breve{\mathbf{X}}^T(t_k)\right]$$

with

$$\breve{\mathbf{X}}_i(t_k) \doteq \left[\begin{array}{cccc} \mathbf{X}^T_i(t_k) & \mathbf{X}^T_i(t_{k-1}) & \ldots & \mathbf{X}^T_i(t_{k-w+1}) \end{array}\right]^T \quad (21)$$

where $i \in \{\mathrm{I}, \mathrm{II}\}$ and $\mathbf{X}_i(t_k)$ is the state vector of the i-th robot at time t_k, as defined in Eq. (13), and w is the size of the smoothing lag. The smoothing lag should be long enough to contain the information from all the time instances involved in each MR measurement, which therefore should be known *a priori* (as opposed to the considered graph-based approach [6]). Thus, since in this section $\mathbf{X}_i \in \mathbb{R}^{15\times 1}$ (cf. Eq. (13)) and recalling that N is the number of robots in the group, the overall dimensions of the centralized smoothing state vector $\breve{\mathbf{X}}(t_k)$ and covariance matrix $\breve{P}(t_k)$ are:

$$\breve{\mathbf{X}}(t_k) \in \mathbb{R}^{15wN\times 1} \quad , \quad \breve{P}(t_k) \in \mathbb{R}^{15wN\times 15wN} \quad (22)$$

In the specific scenario considered in this section, only 2 robots were involved ($N = 2$) and the minimum smoothing window lag to accommodate the measurement schedule (schematically) shown in Figure 1 is $w = 30$, corresponding to storing information of about 30 seconds for each robot in the state vector $\breve{\mathbf{X}}$.

In practice, to simplify the involved computations and because of the assumed known measurement schedule, it is possible to use a reduced version of the augmented state vector: Since in the considered scenario, Robot II contributes only its current information, while Robot I contributes information from two time instances from the past (t_1 and t_2), there is no need to store the past state information for Robot II. Therefore, referring to Eqs. (20) and (21), the actual augmented state vector $\breve{\mathbf{X}}$ for a given three-view measurement involving the time instances t_1, t_2 and t_3, with $t_k \equiv t_3$ being the current time, is of the following form:

$$\breve{\mathbf{X}}(t_3) \doteq \left[\begin{array}{cc} \breve{\mathbf{X}}^T_{\mathrm{I}}(t_3) & \mathbf{X}^T_{\mathrm{II}}(t_3) \end{array}\right]^T \quad (23)$$

where

$$\breve{\mathbf{X}}_{\mathrm{I}}(t_3) \doteq \left[\begin{array}{ccccc} \mathbf{X}^T_{\mathrm{I}}(t_k) & \ldots & \mathbf{X}^T_{\mathrm{I}}(t_2) & \ldots & \mathbf{X}^T_{\mathrm{I}}(t_1) \end{array}\right]^T,$$

while the augmented covariance matrix $\breve{P}$ is accordingly defined as

$$\breve{P}(t_k \equiv t_3) \doteq E[\breve{\mathbf{X}}(t_3)\breve{\mathbf{X}}^T(t_3)] \quad (24)$$

Since $w = 30$, the dimensions of $\breve{\mathbf{X}}$ and $\breve{P}$ are 465×1 and 465×465, respectively.

Updating all the involved robots in the centralized smoothing filter formulation using the specific measurement model of the three-view constraints is the topic of an ongoing research, since not all of the navigation parameters in these robots are observable. Therefore, while the formulation of the centralized smoothing approach allows updating all the involved robots, only Robot II is actually updated in the current implementation, as is the case in the results presented in the previous section.

To avoid updating Robot I in the centralized smoothing approach, while still maintaining a *consistent* augmented covariance matrix $\breve{P}(t_3)$, a set of constraints was introduced: for each state vector $\mathbf{X}_{\mathrm{I}}(t_i)$ of Robot I, which is part of $\breve{\mathbf{X}}(t_3)$, the constraints are

$$H_*\mathbf{X}_{\mathrm{I}}(t_i) = \mathbf{0} \quad (25)$$

where H_* is the identity matrix of the appropriate dimensions. These constraints lead the filter to believe that there is no need to update the state $\mathbf{X}_{\mathrm{I}}(t_i)$ by the three-view MR measurements. Thus, if $\breve{H}$ is the measurement matrix calculated for the augmented state $\breve{\mathbf{X}}(t_3)$:

$$\breve{H} \doteq \begin{bmatrix} 0 & \dots & 0 & H_I(t_2) & 0 & \dots & 0 & H_I(t_1) & H_{II}(t_3) \end{bmatrix}$$

the overall measurement matrix H is

$$\mathcal{H} \doteq \begin{bmatrix} \breve{H}_I & H_{II}(t_3) \\ \breve{I} & 0 \end{bmatrix},$$

where $\breve{H}_I, H_{II}(t_3)$ are defined as

$$\breve{H} \doteq \begin{bmatrix} \breve{H}_I & H_{II}(t_3) \end{bmatrix},$$

and $\breve{I}$ is an identity matrix of the appropriate dimensions.

Once the Kalman gain matrix is calculated using the measurement matrix $\mathcal{H}$, the additional columns in the gain matrix belonging to the above constraints are removed, followed by application of the standard filter equations for updating the augmented state vector $\breve{\mathbf{X}}$ and the covariance matrix $\breve{P}$. Applying constraints in this manner can be also found in [7], where the constraints were used to avoid updating navigation parameters along the motion heading, which are unobservable when considering two-view measurements.

It is important to emphasize that the calculated augmented covariance matrix $\breve{P}$ is consistent, i. e. it contains the actual covariance and cross-covariance terms of the involved robots at different time instances. Therefore, the centralized smoothing solution can be used to validate the solution obtained using the proposed graph-based approach for calculating the involved cross-covariance terms.

VI. Results

In this section the method for graph-based CN is compared to the fixed-lag centralized smoother, while considering three-view measurements. A formation scenario of two robots, described in Section IV is used. To this end, a statistical performance study was performed for both methods in a simulated environment. Further details regarding the simulated environment are described in [6].

The initial navigation errors, as well as the accelerometers bias and gyroscopes drift, were assumed to be of a zero-mean Gaussian distribution, with standard deviation values specified in Table I. In each simulation run, the drawn values of initial navigation errors were used for calculating the actual initial navigation solution, while the drawn drift and bias were applied to corrupt the ground-truth IMU measurements. The IMU measurements were also contaminated by a zero-mean Gaussian noise with a standard deviation of $100\ \mu g/\sqrt{Hz}$ and $0.001\ deg/\sqrt{hr}$ for the accelerometers and gyroscopes, respectively.

Figures 2-3 present the comparison results obtained from the statistical performance study. The results are shown in terms of the standard deviation error (σ), while the mean error is zero in both cases. Only the performance of Robot II is shown, since Robot I develops inertially as it is not updated by the three-view MR measurements.

As can be seen, the overall performance obtained using the graph-based approach is quite similar to the performance of the centralized smoothing approach. The errors in case of a centralized smoothing approach are slightly lower in the first few updates, in particular in the velocity and bias states (cf. Figures 2(b) and 3(b)). However, the graph-based approach manages to close most of this gap in all the shown navigation states (position, velocity, Euler angles and bias) after incorporating additional three-view MR measurements. The drift state is poorly estimated by both methods (not shown).

The fact that the graph-based approach is somewhat inferior to the centralized smoothing approach is not surprising, since the centralized smoothing approach essentially represents the optimal solution (given the linearization assumptions), while the graph-based approach makes an additional approximation, in the process of calculating the Kalman gain matrix. As discussed in [6], [9], [10], this approximation is made whenever only part of the involved robots are updated, as indeed is the case in this section.

However, while the centralized smoothing approach provides up-to-linearization optimal solution, it has two significant deficiencies compared to the proposed graph-based approach.

First, the fixed-lag centralized smoothing approach supports only MR measurements that involve time instances within the fixed lag window. Using a large (or unlimited) smoothing lag is not practical, since the dimensions of the augmented state vector and the augmented covariance matrix grow rapidly as a function of the smoothing lag size (cf. Eq. (22)). Therefore, choosing the actual size of the lag window requires *a priori* knowledge of the MR measurements to be performed. For example, in the scenario considered herein, it was possible to set the smoothing lag to 30 only because the three-view MR measurements schedule was assumed to

be known. In contrast to this, the proposed graph-based approach does not require the MR measurement schedule to be known a priori: even if the graph structure has a limited capacity, it can be used to process any MR measurement schedule, provided there is enough space to accommodate the involved information [6].

Second, the augmented state vector and covariance matrix should be constantly propagated and updated by the filter, resulting in a continuous high computational load, whereas in the proposed graph-based approach, a basic local state vector is maintained by each robot and the appropriate cross-covariance terms are calculated only when required.

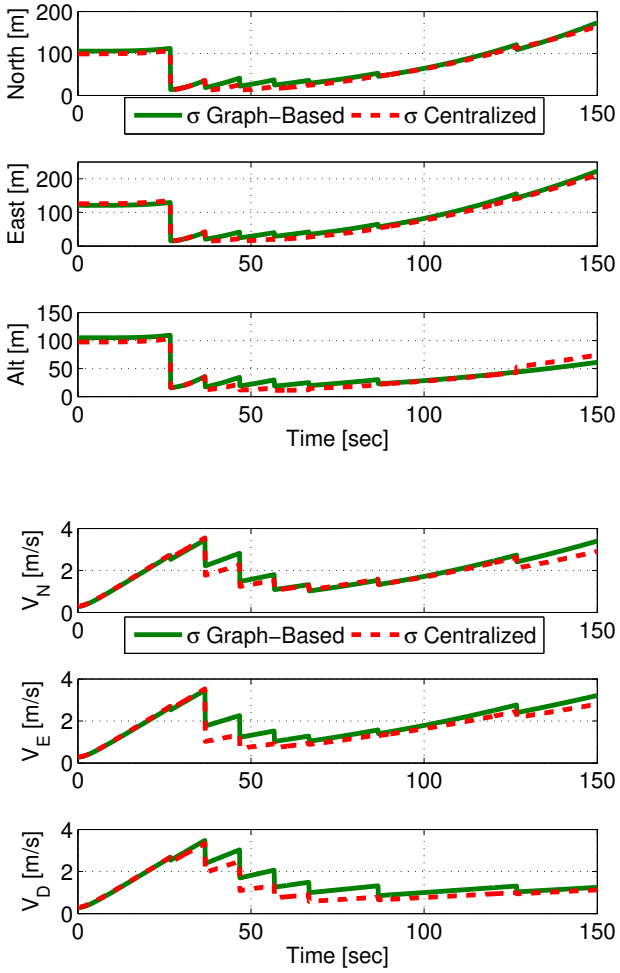

Figure 2. Monte-Carlo results comparing between the graph-based approach and the fixed-lag centralized smoother approach. (top) Position errors. (bottom) Velocity errors.

Consequently, this comparison validates the graph-based approach for cross-covariance calculation (at least) for similar scenarios and MR measurement models.

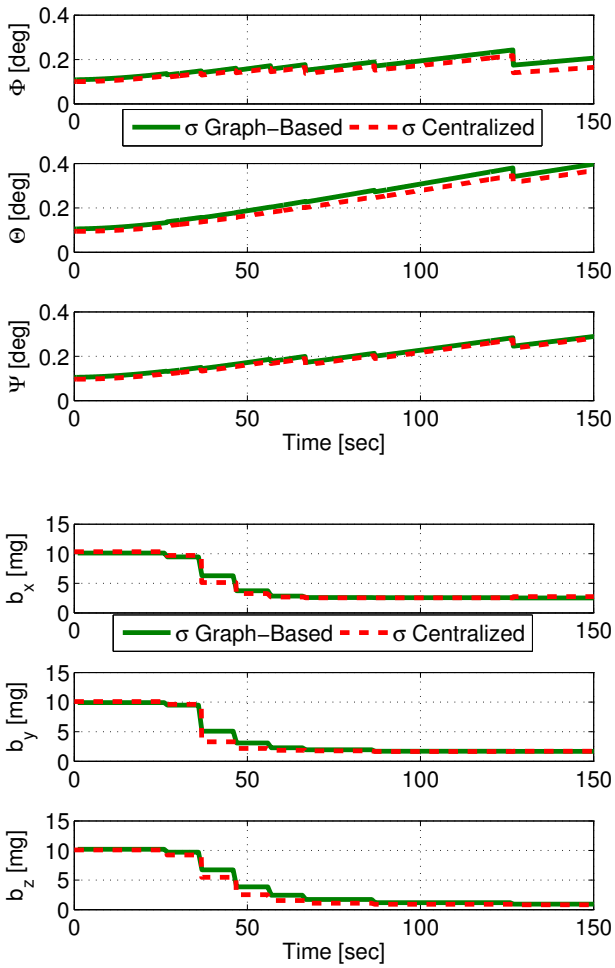

Figure 3. Monte-Carlo results comparing between the graph-based approach and the fixed-lag centralized smoother approach. (top) Attitude errors. (bottom) Bias estimation errors.

VII. Conclusions

This paper considered a recently-developed graph-based approach for consistent cooperative navigation, in which the required cross-covariance terms were explicitly calculated on-demand. While the original approach was formulated for a general multi-robot measurement model, it was recently applied to vision-aided cooperative navigation based on three-view constraints. This paper discussed the difficulties in other existing methods for maintaining a consistent information fusion, and validated the graph-based approach by a comparison to a fixed-lag centralized smoother. As opposed to the considered approach, applying a fixed-lag centralized smoother required assuming a known schedule of the three-view multi-robot measurements, thereby determining the length of the fixed-lag. The comparison clearly

showed the advantage of using the on-demand approach.

REFERENCES

[1] A. Bahr, M. R. Walter, and J. J. Leonard. Consistent cooperative localization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3415–3422, Kobe, Japan, May 2009.

[2] V. Caglioti, A. Citterio, and A. Fossati. Cooperative, distributed localization in multi-robot systems: a minimum-entropy approach. In *Proceedings of the IEEE Workshop on Distributed Intelligent Systems*, pages 25–30, 2006.

[3] J. W. Fenwick, P. M. Newman, and J. J. Leonard. Cooperative concurrent mapping and localization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington, USA, May 2002.

[4] R. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge University Press, 2000.

[5] A. Howard, M. J. Mataric, and G. S. Sukhatme. Putting the 'i' in 'team' - an ego-centric approach to cooperative localization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 868–874, Taipei, Taiwan, 2003.

[6] V. Indelman. Navigation performance enhancement using online mosaicking. Technion, Israel, 2011.

[7] V. Indelman, P. Gurfil, E. Rivlin, and H. Rotstein. Navigation aiding based on coupled online mosaicking and camera scanning. *Journal of Guidance, Control and Dynamics*, 33(6):1866–1882, 2010.

[8] V. Indelman, P. Gurfil, E. Rivlin, and H. Rotstein. Graph-based distributed cooperative navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Shanghai, China, May 2011.

[9] 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.

[10] V. Indelman, P. Gurfil, E. Rivlin, and H. Rotstein. Distributed vision-aided cooperative localization and navigation based on three-view geometry. *Robotics and Autonomous Systems*, 2012, to appear.

[11] R. Kurazume, S. Nagata, and S. Hirose. Cooperative positioning with multiple robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1250–1257, San Diego, CA, May 1994.

[12] M. T. Lazaro and J. A. Castellanos. Localization of probabilistic robot formations in slam. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, May 2010.

[13] G. L. Mariottini, G. Pappas, D. Prattichizzo, and K. Daniilidis. Vision-based localization of leader-follower formations. In *Proceedings of the IEEE International Conference on Decision and Control*, pages 635–640, Seville, Spain, 2005.

[14] A. Martinelli, F. Pont, and R. Siegwart. Multi-robot localization using relative observations. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pages 2797–2802, Barcelona, Spain, 2005.

[15] L. Montesano, J. Gaspar, J. Santos-Victor, and L. Montano. Fusing vision-based bearing measurements and motion to localize pairs of robots. In *Proceedings of the ICRA Workshop on Cooperative Robotics*, Barcelona, Spain, 2005.

[16] A. I. Mourikis, S. I. Roumeliotis, and J. W. Burdick. Sc-kf mobile robot localization: A stochastic cloning kalman filter for processing relative-state measurements. *IEEE Transactions on Robotics*, 23(4):717–730, 2007.

[17] E. D. Nerurkar and S. I. Roumeliotis. Multi-centralized cooperative localization under asynchronous communication. *Department of Computer Science and Engineering, University of Minnesota, Technical Report*, March 2010.

[18] S. I. Roumeliotis and G. A. Bekey. Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18(5):781–795, 2002.

[19] C. Smaili, M. E. E. Najjar, and F. Charpillet. Multi-sensor fusion method using bayesian network for precise multi-vehicle localization. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, pages 906–911, Beijing, China, 2008.