Graph-based Distributed Cooperative Navigation

Vadim Indelman, Pini Gurfil, Ehud Rivlin and Hector Rotstein

Abstract— **This paper addresses the problem of distributed cooperative navigation. A new graph-based method is developed for on-demand calculation of the required correlation terms, considering a general multi-robot measurement model. These correlation terms are necessary for the consistent EKF-based data fusion when several statistically-dependent sources of information are used. The measurement model relates between the navigation information transmitted by any number of robots and the actual readings taken by the available onboard sensors. The transmitted information is not necessarily of the current time instant, but may actually belong to some time instant from the past. Experiment results and a theoretical example of the developed method are presented considering a three-view measurement, formulated upon receiving three images of the same scene, captured by different robots at different a priori unknown time instances.**

I. Introduction

The ability of a group of cooperative robots to carry out various tasks strongly depends on navigation capabilities of each individual in the group. Most of the common navigation systems rely on the GPS for correcting the dead-reckoning errors. However, not all the robots may be equipped with a GPS receiver, and moreover, the GPS signal is unavailable when operating indoors, underwater, or on other planets.

When a group of collaborative robots is considered, the robots may provide navigation aids one to each other during their respective missions. For example, a navigation update may be performed whenever one robot observes another robot and uses its on-board camera and range sensor to measure the relative pose between themselves [1]-[5]. Another type of cooperative navigation (CN) update is to identify a common scene observed by different robots, and to express the resulting constraints as a measurement to the navigation filter. Such an approach was recently suggested by [6], [7] considering measurements that combine pairs of robots. In [8], it is suggested to utilize a technique of vision-aided navigation based on three-view geometry [9] to distributed CN. A measurement is formulated whenever a scene is observed from three different views, which may be captured by different robots.

However, the involved navigation information from different robots may be statistically dependent, regardless of the applied measurement model. For instance, the navigation information of any two robots becomes correlated after the first update is carried out. Ignoring this correlation may result in inconsistent and over-confident estimations [11].

V. Indelman and P. Gurfil are with the Faculty of Aerospace Engineering, Technion - Israel Institute of Technology, Haifa 32000, Israel (e-mail: ivadim@tx.technion.ac.il, pgurfil@technion.ac.il).

E. Rivlin is with the Department of Computer Science, Technion - Israel Institute of Technology, Haifa 32000, Israel (e-mail: ehudr@cs.technion.ac.il).

H. Rotstein is with RAFAEL - Advanced Defense Systems Limited, Israel (e-mail: hector@rafael.co.il).

Several approaches have been proposed for coping with the correlations terms in multi-robot scenarios, assuming relative pose measurements. In [2], an augmented covariance matrix, comprised of covariance and cross-covariance matrices between all the robots in the group, is maintained in a distributed framework. Howard et. al. [10] suggested a method for avoiding handling correlated updates. In [11], the cross-covariance terms are also 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.

This paper presents a new graph-based approach for calculating the required correlation terms in the EKF update step assuming a *general* multi-robot (MR) measurement model. This model relates between the navigation information from any number of robots and the actual readings of the onboard sensors. These are *not* necessarily taken at the same time. Therefore, in addition to the a priori unknown identity of the robots participating in each MR measurement, the time instances are a priori unknown as well.

These additional unknown parameters render any approach that is based on maintaining the correlation terms impractical. As a consequence, one has either to assume statistical independence, thereby neglecting the actual correlation terms, or to compute these terms upon demand, as suggested in this paper. The graph-based approach developed herein, allows calculating the required correlation terms in general scenarios, possibly involving different MR measurement models and regardless of the MR measurements that were applied so far. It should be noted that a graph-based approach was already proposed in [7], where a nonlinear optimization is solved each time a new measurement arrives, considering relative pose and two-view measurements between pairs of robots. In contrast to this, we consider an EKF formulation, commonly used for CN, and a general MR measurement model that involve any number of robots.

II. Problem Formulation

A general nonlinear measurement $\mathbf{z}$, involving information taken from r robots of the N robots in the group, can be formulated as follows

$$\mathbf{z}(t) = \mathbf{h}(\{\chi_i(t_i)\}_{i=1}^{r}) \tag{1}$$

where $\chi_i(t_i)$ contains information taken from the ith robot at time t_i. It is typically composed of navigation data, and of parameters related to the applied measurement model, such as range or imagery data. Each robot may also contribute information generated at several different time instances.

Denote by $\mathbf{X}_i$ the errors in the navigation solution of the ith robot, and let Φ^i and $\boldsymbol{\omega}^i$ be the discrete system matrix and the discrete process noise, such that for any two time instances t_a and t_b, $t_a < t_b$:

$$\mathbf{X}_i(t_b) = \Phi^i_{t_a \to t_b} \mathbf{X}_i(t_a) + \boldsymbol{\omega}^i_{t_a \to t_b} \tag{2}$$

Denote also by $\tilde{\mathbf{X}}_i$ the estimation error of $\mathbf{X}_i$. Linearizing Eq. (1) about the parameters stored in the set $\{\chi_i\}_{i=1}^r$ gives

$$\mathbf{z} \approx \sum_{i=1}^{r} H_i(t_i)\mathbf{X}_i(t_i) + D_i(t_i)\mathbf{v}_i(t_i) \tag{3}$$

where H_i is the Jacobian of $\mathbf{z}$ with respect to the navigation solution of the ith robot, D_i is the Jacobian of $\mathbf{z}$ with respect to the other parameters in χ_i, and $\mathbf{v}_i$ is the error in these parameters, representing the measurement noise.

For example, a measurement that is generated based upon three images of the same scene, possibly captured by different robots, may be written as [9], [8] $\mathbf{z} \approx H_3\mathbf{X}_3 + H_2\mathbf{X}_2 + H_1\mathbf{X}_1 + D\mathbf{v}$, where $D\mathbf{v}$ represents the contribution of image noise from all the three images to the measurement.

Next, the method for on-demand calculation of the cross-covariance terms between the involved robots is described.

III. Concept

The cross-covariance terms may be calculated in the general case, given the history of the applied measurements models, i. e.: when and/or what measurements were applied, and what time instances and which robots were involved. These details may be represented by a directed graph $G = (V, E)$, locally maintained by every robot in the group, where V is the set of nodes and E is the set of directed weighted arcs. The weight of each arc reflects the information flow between the two connected nodes. The construction process of the graph and the communication protocol between the robots is discussed in [8].

The technique developed hereafter assumes that G is a directed *acyclic* graph (DAG). In a general MR scenario, this is guaranteed if each measurement is utilized for updating only the robots that contributed their current navigation information, i. e. $t_i = t$. For simplicity, in this paper we consider only *one* such robot. However, there are specific scenarios in which G remains a DAG even when all the robots involved in a given measurement are updated.

Two type of nodes exist in G. Nodes of the first type represent $\chi_i(t_i)$, i. e. the information taken from the ith robot at time instant t_i, that participates in one of the MR updates. The second type are nodes representing the update-events. Each node in G, can be connected to another node by a transition relation, and in addition, it may be involved in an MR measurement, in which case it would be also connected to an update-event node by an update relation.

The transition relation is given by Eq. (2), relating between the state vectors of the ith robot at two different time instances t_a and t_b. G will contain two nodes, representing these two time instances, only if each of them participates in some MR measurement. In this case, these two nodes will be connected by an arc, weighted by the transition matrix $\Phi^i_{t_a \to t_b}$. The noise process covariance matrix $Q^i_{t_a \to t_b} \doteq E[\boldsymbol{\omega}^i_{t_a \to t_b}(\boldsymbol{\omega}^i_{t_a \to t_b})^T]$ is associated to this arc as well.

In a general MR measurement model, given in Eq. (3), the a posteriori estimation error of the updated robot, denoted by q, is given by $\tilde{\mathbf{X}}_q^+ = (I - K_q H_q)\tilde{\mathbf{X}}_q^- - K_q \sum_{i=1\ ,\ i\neq q}^r H_i \tilde{\mathbf{X}}_i^- - K_q \sum_{i=1}^r D_i \mathbf{v}_i$, where each of the state vectors is given in an appropriate time instant. The graph G will contain $r+1$ nodes representing the above equation. Letting the nodes β_i represent the a priori estimation errors and the node α represents the a posteriori estimation error, the MR update relation is given by[1]

$$\begin{aligned} \tilde{\mathbf{X}}_\alpha &= \left(I - K_{\beta_q} H_{\beta_q}\right)\tilde{\mathbf{X}}_{\beta_q} - \\ &- K_{\beta_q} \sum_{i=1\ ,\ i\neq q}^{r} H_{\beta_i}\tilde{\mathbf{X}}_{\beta_i} - K_{\beta_q}\sum_{i=1}^{r} D_{\beta_i}\mathbf{v}_{\beta_i} \end{aligned} \tag{4}$$

The nodes β_i and β_q are the parents[2] of the node α. Thus, the weight of an arc connecting between a node β_i, representing $\tilde{\mathbf{X}}_i^-$, with the update-event node α, representing $\tilde{\mathbf{X}}_q^+$, is given by $-K_q H_i$ for $i \neq q$. This arc is also associated with a measurement noise covariance matrix $K_q D_i R_i (K_q D_i)^T$, where $R_i \doteq E[\mathbf{v}_i \mathbf{v}_i^T]$. The weight of an arc connecting β_q with α, is $I - K_q H_q$ and it is also associated with $K_q D_q R_q (K_q D_q)^T$.

It is assumed that the a priori and a posteriori covariance and cross-covariance terms between the nodes, which participated in the *same* MR update in the past, are known (e. g. this information can be stored in the nodes themselves).

Assume we need to calculate the cross-covariance between some two nodes c and d, representing $\tilde{\mathbf{X}}_c$ and $\tilde{\mathbf{X}}_d$, respectively, that may belong to different robots in the group. The first step is to construct two inverse-trees, containing all the possible routes in G to each of the nodes c and d.

This is performed as follows. The first tree, T_c, is initialized with the node c. Each next level is comprised of the parents of nodes that reside in the previous level, as determined from G, e. g., the second level of T_c contains all the nodes in G that are directly connected to c. The same process is executed for constructing a tree T_d for the node d. An example of such trees for 3-robot measurements is given in Section V-A. Note that every node in T_c and T_d has only one child but may have one or r parents. In the latter case, the node represents an MR update event.

The convention used here is that if some node a_i has several parents, the jth parent is denoted as a_{i+1}^j. Also, $a \equiv a_1$. See Fig. 1.

The concept of the proposed graph-based approach for calculating the cross-covariance terms is as follows. We start with the two nodes c and d, which are the first-level nodes in T_c and T_d. Since the term $E[\tilde{\mathbf{X}}_c \tilde{\mathbf{X}}_d^T]$ is unknown, we proceed

[1]The identity of the robots and time instances are represented by the node notations, and therefore these are not shown from now on explicitly.

[2]The node b is a parent of a node a only if there is an arc from b to a.

to the parents of these nodes. As noted above, two types of relations exist for a general graph topology. For example, assume that both of the nodes c and d are related to nodes in the next level by a transition relation (cf. Fig. 1). Thus, c and d each have only one parent, denoted as c_2 and d_2, respectively. The nodes c_2 and d_2 constitute the second level in the trees T_c and T_d, respectively. For instance, c_2 and c are connected via $\tilde{\mathbf{X}}_c = \Phi_{c_2\to c}\tilde{\mathbf{X}}_{c_2} + \boldsymbol{\omega}_{c_2\to c}$.

Now, $E[\tilde{\mathbf{X}}_c\tilde{\mathbf{X}}_d^T]$ may be written in several forms: $E\left[\tilde{\mathbf{X}}_c\left(\Phi_{d_2\to d}\tilde{\mathbf{X}}_{d_2} + \boldsymbol{\omega}_{d_2\to d}\right)^T\right]$, or $E\left[\left(\Phi_{c_2\to c}\tilde{\mathbf{X}}_{c_2} + \boldsymbol{\omega}_{c_2\to c}\right)\tilde{\mathbf{X}}_d^T\right]$, or $E\left[\left(\Phi_{c_2\to c}\tilde{\mathbf{X}}_{c_2} + \boldsymbol{\omega}_{c_2\to c}\right)\left(\Phi_{d_2\to d}\tilde{\mathbf{X}}_{d_2} + \boldsymbol{\omega}_{d_2\to d}\right)^T\right]$. Since the expression from the previous level, i. e. the first level, was already checked, it is now required to check whether any of the expressions involving nodes from the current level are known. In other words, the question is whether any of the pairs $E[\tilde{\mathbf{X}}_c\tilde{\mathbf{X}}_{d_2}^T]$, $E[\tilde{\mathbf{X}}_{c_2}\tilde{\mathbf{X}}_d^T]$ and $E[\tilde{\mathbf{X}}_{c_2}\tilde{\mathbf{X}}_{d_2}^T]$ are known. In addition, it is also required to know the correlation between the noise terms and the state vectors.

Assuming none of the pairs is known, we proceed to the next level, the third level. *Each* node in the second level may have either transition or MR update relation, with the nodes in the third level. Having reached a new level, the third level, new expressions for the required term $E[\tilde{\mathbf{X}}_c\tilde{\mathbf{X}}_d^T]$ may be written utilizing nodes from *this* level and *lower* levels.

Note that all the expressions from the previous level, the second level, were already analyzed. Putting aside for the moment terms that involve noise, it is obvious that less terms are required to be analyzed when nodes closer to c or d are considered. Therefore, it is preferred to start analyzing from the lower level and upward.

Consider that some term, for example, $E[\tilde{\mathbf{X}}_{c_3}\tilde{\mathbf{X}}_{d_3^1}^T]$, is known. This means that the nodes c_3 and d_3^1 in T_c and T_d, respectively, represent state vectors that either have participated in the same MR measurement in the past (and therefore $E[\tilde{\mathbf{X}}_{c_3}\tilde{\mathbf{X}}_{d_3^1}^T]$ was stored), or that these two nodes are identical ($c_3 \equiv d_3^1$). In any case, the known term $E[\tilde{\mathbf{X}}_{c_3}\tilde{\mathbf{X}}_{d_3^1}^T]$, accordingly weighted is part of $E[\tilde{\mathbf{X}}_c\tilde{\mathbf{X}}_d^T]$. Having a known term also means that there is no need to proceed to nodes of higher levels which are related to this term.

The procedure proceeds to higher levels until either all the terms required for calculating the cross-covariance $E[\tilde{\mathbf{X}}_c\tilde{\mathbf{X}}_d^T]$ are known, or reaching the top level in both trees. In the later case, the unknown terms of cross-covariance are zero.

The process noise terms are assumed to be statistically independent, $E[\boldsymbol{\omega}_{i_1\to i_2}\boldsymbol{\omega}_{j_1\to j_2}^T] = 0$, if $\boldsymbol{\omega}_{i_1\to i_2}$ and $\boldsymbol{\omega}_{j_1\to j_2}$ belong to different robots, or, otherwise $\{i_1 \to i_2\} \cap \{j_1 \to j_2\} = \phi$. The measurement noise is assumed to be statistically independent with the process noise. On the other hand, the process and measurement noise terms may be statistically *dependent* with the involved state vectors, as analyzed in the sequel.

IV. Algorithm

The algorithm gradually processes permutations between nodes in T_c and nodes in T_d at different levels, starting from the first level. The permutation set of the kth level is denoted by $\mathcal{M}_k$, with $\mathcal{M}_1 \doteq \{(c, d)\}$. Next, we describe an algorithm for calculating $E[\tilde{\mathbf{X}}_c\tilde{\mathbf{X}}_d^T]$ based on $\mathcal{M}_k$ from different levels.

Denote by $\breve{a}$ a descendant node of a. A pair of nodes (a, b) is said to be known, if $E[\tilde{\mathbf{X}}_a\tilde{\mathbf{X}}_b^T]$ is known, i. e.: it can be immediately retrieved, without any calculations, from the data stored in G. Denote also any route $a_k \to \cdots \to a_2 \to a$ in some tree T_b as $a_k \overset{T_b}{\Longrightarrow} a$.

A. Processing a single member of $\mathcal{M}_k$

In the general case, when processing the permutation set $\mathcal{M}_k$ from level k, all the nodes on the route to the leaf (which is c in T_c and d in T_d) should be considered, starting with the leaf and going up. For example, assume that for some member (c_k, d_k) in $\mathcal{M}_k$, the routes to the leaf nodes are $c_k \overset{T_c}{\Longrightarrow} c$ and $d_k \overset{T_d}{\Longrightarrow} d$. Fig. 1(a) schematically illustrates a general route $c_k \overset{T_c}{\Longrightarrow} c$. We start with checking whether (c_k, d) or (c, d_k) are known. If not, we proceed to check whether (c_k, d_2) or (c_2, d_k) are known, and so on. The procedure ends when a known pair of nodes is found, or when reaching and analyzing the couple (c_k, d_k). When a known couple of nodes is discovered, its contribution to the cross-covariance $E[\tilde{\mathbf{X}}_c\tilde{\mathbf{X}}_d^T]$ is calculated.

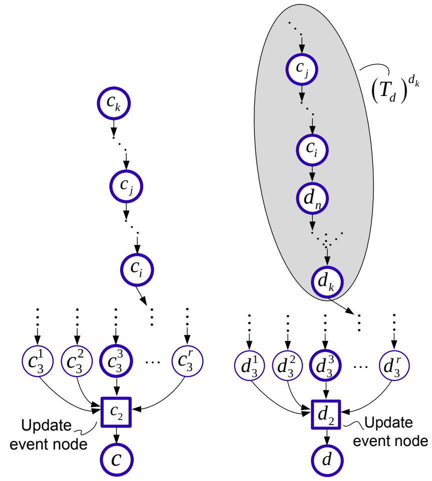
Denote the overall weight of the routes $c_k \overset{T_c}{\Longrightarrow} c$ and $d_k \overset{T_d}{\Longrightarrow} d$ by $W_c(c_k)$ and $W_d(d_k)$, respectively. If a known pair $(c_k, \breve{d}_k)$ was discovered, its contribution to $E[\tilde{\mathbf{X}}_c\tilde{\mathbf{X}}_d^T]$ is calculated according to: $W_c(c_k)E[\tilde{\mathbf{X}}_{c_k}\tilde{\mathbf{X}}_{\breve{d}_k}^T]W_d^T(\breve{d}_k) + Q_{c_k\breve{d}_k}$. Similarly, if $(\breve{c}_k, d_k)$ is the discovered known pair, its contribution is $W_c(\breve{c}_k)E[\tilde{\mathbf{X}}_{\breve{c}_k}\tilde{\mathbf{X}}_{d_k}^T]W_d^T(d_k) + Q_{\breve{c}_k d_k}$. The noise terms $Q_{c_k\breve{d}_k}$ and $Q_{\breve{c}_k d_k}$ are analyzed in Section IV-C.

Denoting by $w(a, b)$ the arc weight connecting the node a to the node b in G, then $W_c(c_k)$ and $W_d(d_k)$ are given by $\Pi_{i=2}^k w(c_i, c_{i-1})$ and $\Pi_{i=2}^k w(d_i, d_{i-1})$, respectively.

After finishing analyzing some member (c_k, d_k) in $\mathcal{M}_k$, the permutation set $\mathcal{M}_k$ is updated as follows. If (c_k, d_k) is known, it is removed from $\mathcal{M}_k$. However, if $(c_k, \breve{d}_k)$ or $(\breve{c}_k, d_k)$ were found known, we define $\mathcal{M}'_k$ as $\{(c_k, d') \mid d' \in \{\pi_d(\breve{d}_k)\}\ ,\ (c_k, d') \in \mathcal{M}_k\}$ in the former case, and as $\{(c', d_k) \mid c' \in \{\pi_c(\breve{c}_k)\}\ ,\ (c', d_k) \in \mathcal{M}_k\}$ in the later case. $\mathcal{M}_k$ is then updated according to $\mathcal{M}_k = \mathcal{M}_k \diagdown \mathcal{M}'_k$.

B. Calculation of $\mathcal{M}_{k+1}$

Having described how each level in the trees T_c and T_d is handled, we turn to explain the mechanism for advancing to the next level. After finishing processing all the members in $\mathcal{M}_k$, the only members left in $\mathcal{M}_k$ are those for whom the procedure did not find any known pair. If $\mathcal{M}_k$ is empty, the algorithm terminates. The set of permutations in the next level, $\mathcal{M}_{k+1}$, is constructed based on the parents of each of the nodes that appear in $\mathcal{M}_k$: $\mathcal{M}_{k+1} = \left\{(c_{k+1}^s, d_{k+1}^t) \mid c_{k+1}^s \in \{\pi_c(a)\}, d_{k+1}^t \in \{\pi_d(b)\}\right., \forall (a, b) \in \left.\mathcal{M}_k\right\}$, where s and t distinguish between several parents.



(a) The tree T_c (b) The tree T_d

Fig. 1. The node c_j in T_c has descendents that appear as ancestors of d_k in $(T_d)^{d_k}$, therefore contributing noise terms to the calculated $E[\tilde{\mathbf{X}}_c\tilde{\mathbf{X}}_d^T]$.

C. Contribution of Noise Terms

Consider a general case of processing some member $(c_k, d_k) \in \mathcal{M}_k$, as described in Section IV-A.

Let $\boldsymbol{\eta}_{c_i:c_j}$, $i > j$, denote the involved noise term in expressing $\tilde{\mathbf{X}}_{c_j}$ by $\tilde{\mathbf{X}}_{c_i}$, and possibly other nodes outside the route $c_i \to \cdots \to c$. $\boldsymbol{\eta}$ may involve noise terms of two types: process noise and measurement noise.

Define $(T_a)^b$ for some nodes b and a in G as follows. Given the location of node b in the tree T_a, $(T_a)^b$ is a sub-tree of T_a, comprised of all the ancestors of b in T_a and the node b itself (cf. Fig. 1(b)). Let c_m and d_p be some two nodes in the trees T_c and T_d, respectively.

Lemma 4.1: If $(T_d)^{d_p}$ does *not* contain any nodes from the route $c_m \xRightarrow{T_c} c$, then $\boldsymbol{\eta}_{c_m:c}$ and $\tilde{\mathbf{X}}_{d_p}$ are statistically-independent.
The proof of this and other Lemmas to follow, can be found in [12].

In particular, if T_d does *not* contain any nodes from the route $c_m \xRightarrow{T_c} c$, then $\boldsymbol{\eta}_{c_m:c}$ and $\tilde{\mathbf{X}}_d$ are statistically-independent. Similarly, the above is also valid when considering $(T_c)^{c_m}$ or T_c, with the proper adjustments.

At this point we assume, without loss of generality, that in the process of analyzing the member (c_k, d_k), described in Section IV-A, the pair (c_j, d_k) was discovered as known. Since nodes from lower levels are analyzed first, no other known pair (c_r, d_k) or (c_k, d_r) exists with $r < j$.

Lemma 4.2: The route $d_k \xRightarrow{T_d} d$ does not contain any node c_r from the route $c_j \to \cdots \to c_r \to \cdots \to c$ in T_c for any $1 \le r < j$. In case $r = j$, the node $c_r = c_j$, may only appear in the route $d_k \xRightarrow{T_d} d$ as d_k.

The conclusion from Lemmas 4.1 and 4.2 is that if $(T_d)^{d_k}$ does *not* contain any nodes from the route $c_j \xRightarrow{T_c} c$, then $\boldsymbol{\eta}_{c_j:c}$ is statistically-independent with all the states represented by the nodes $\{d_k \xRightarrow{T_d} d\} \cup (T_d)^{d_k}$. Here $\{d_k \xRightarrow{T_d} d\}$ denotes the group of nodes in the route $d_k \xRightarrow{T_d} d$.

Yet, $\boldsymbol{\eta}_{c_j:c}$ may still be statistically-*dependent* with states represented by the nodes in $T_d \setminus \{d_k \xRightarrow{T_d} d\} \setminus (T_d)^{d_k}$, if among these nodes there is at least one node from the route $c_j \xRightarrow{T_c} c$. This leads to the following statement.

Lemma 4.3: If during the whole execution of the algorithm described in Sections IV-A and IV-B, for all the found known pairs (a, b), $a \in T_c$ and $b \in T_d$, there are no descendants of a in T_c that are ancestors of b in T_d, and there are no descendants of b in T_d that are ancestors of a in T_c, then *all* the involved noise terms from T_c, participating in calculation of $E[\tilde{\mathbf{X}}_c\tilde{\mathbf{X}}_d^T]$, are statistically-independent with $\tilde{\mathbf{X}}_d$, and *all* the involved noise terms from T_d are statistically-independent with $\tilde{\mathbf{X}}_c$.

When the conditions of Lemma 4.3 are not satisfied, $E[\tilde{\mathbf{X}}_c\tilde{\mathbf{X}}_d^T]$ will contain expressions involving noise covariance from different time instances and robots. Returning to the discovered known pair (c_j, d_k), we now assume that there *are* descendants of c_j in T_c that appear as ancestors of d_k in T_d. Among these descendants, denote by c_i the descendant that is closest to c, as illustrated in Fig. 1. The child of c_i in T_d is denoted by d_n.

Lemma 4.4: The route $c_j \xRightarrow{T_c} c_i$ appears in $(T_d)^{d_k}$.

Observe that Lemma 4.4 is also valid for any sub-route $c_j \xRightarrow{T_c} c_{i'}$ of the route $c_j \xRightarrow{T_c} c_i$. Furthermore, $(T_d)^{d_k}$ might contain several instances of the sub-routes $c_j \xRightarrow{T_c} c_{i'}$.

Now we analyze the correlation between the noise term $\boldsymbol{\eta}_{c_l:c_{l-1}}$, involved between any two adjacent nodes c_l and c_{l-1} in the route $c_j \to \cdots \to c_l \to c_{l-1} \to \cdots \to c_i$, and $\tilde{\mathbf{X}}_{d_k}$. The term $E[\boldsymbol{\eta}_{c_l:c_{l-1}}\tilde{\mathbf{X}}_{d_k}^T]$, with $i+1 \le l \le j$, may be calculated as follows.

Assume for now that $(T_d)^{d_k}$ contains only a single appearance of $c_l \to c_{l-1}$. Then $\tilde{\mathbf{X}}_{d_k}$ is given by (cf. Fig. 1)

$$\begin{aligned}\tilde{\mathbf{X}}_{d_k} &= W_{d_k}(c_l)\tilde{\mathbf{X}}_{c_l} + \sum_{r=k}^{n-1} W_{d_k}(d_r)\boldsymbol{\eta}_{d_{r+1}:d_r} + \\ &+ W_{d_k}(d_n)\boldsymbol{\eta}_{c_i:d_n} + \sum_{r=i}^{l-1} W_{d_k}(c_r)\boldsymbol{\eta}_{c_{r+1}:c_r} + \boldsymbol{\nu}_d\end{aligned} \tag{5}$$

where $\boldsymbol{\nu}_d$ is composed of state vectors and noise terms represented by nodes in $(T_d)^{d_k} \setminus \{c_l \to c_{l-1} \to \cdots \to d_k\}$.

Since it was assumed that $c_l \to c_{l-1}$ appears only once in $(T_d)^{d_k}$, $(T_d)^{d_k} \setminus \{c_l \to c_{l-1} \to \cdots \to d_k\}$ does not contain $c_l \to c_{l-1}$. Therefore, according to Lemma 4.1, $\boldsymbol{\eta}_{c_l:c_{l-1}}$ and $\boldsymbol{\nu}_d$ are statistically-independent and thus, from Eq. (5), $E[\boldsymbol{\eta}_{c_l:c_{l-1}}\tilde{\mathbf{X}}_{d_k}^T]$ is given by $E[\boldsymbol{\eta}_{c_l:c_{l-1}}\boldsymbol{\eta}_{c_l:c_{l-1}}^T]W_{d_k}^T(c_{l-1})$.

The term $E[\boldsymbol{\eta}_{c_l:c_{l-1}}\boldsymbol{\eta}_{c_l:c_{l-1}}^T]$ is equal to a process noise covariance in case the arc connecting c_l and c_{l-1} represents a transition relation, and to the appropriate measurement noise covariance, in case of an update relation (cf. Section III).

However, in a general case, $(T_d)^{d_k}$ may contain several instances of $c_l \to c_{l-1}$, each instance with its own route $c_l \to c_{l-1} \to \cdots \to d_k$. Letting u distinguish between these different instances of $c_l \to c_{l-1}$ in $(T_d)^{d_k}$, and denoting by $W_b^u(a)$ the overall weight of the uth route from the node a to the node b in the tree T_b, the term $E[\boldsymbol{\eta}_{c_l:c_{l-1}}\tilde{\mathbf{X}}_{d_k}^T]$ becomes: $E[\boldsymbol{\eta}_{c_l:c_{l-1}}\boldsymbol{\eta}_{c_l:c_{l-1}}^T]\sum_u\left(W_{d_k}^u(c_{l-1})\right)^T$.

Furthermore, when considering the whole tree T_d, $c_l \to c_{l-1}$ may appear not only in $(T_d)^{d_k}$. According to Lemma 4.2, $c_l \to c_{l-1}$ cannot appear in $d_k \stackrel{T_d}{\Longrightarrow} d$. Thus, in addition to $(T_d)^{d_k}$, $c_l \to c_{l-1}$ may be also found only in $T_d \diagdown (T_d)^{d_k} \diagdown \{d_k \stackrel{T_d}{\Longrightarrow} d\}$. However, the contribution of the correlation between $\boldsymbol{\eta}_{c_l:c_{l-1}}$ and the state vectors represented by nodes in $T_d \diagdown (T_d)^{d_k} \diagdown \{d_k \stackrel{T_d}{\Longrightarrow} d\}$ would be calculated when processing other members in $\mathcal{M}_k$.

Therefore, the contribution of $E[\boldsymbol{\eta}_{c_l:c_{l-1}}\tilde{\mathbf{X}}_d^T]$ to $E[\tilde{\mathbf{X}}_c\tilde{\mathbf{X}}_d^T]$, due to the known pair (c_j, d_k), is $E[\boldsymbol{\eta}_{c_l:c_{l-1}}\boldsymbol{\eta}_{c_l:c_{l-1}}^T]\sum_u\left(W_d^u(c_{l-1})\right)^T$, with $i+1 \leq l \leq j$.

$\tilde{\mathbf{X}}_c$ can be expressed in a similar manner to Eq. (5) as (cf. Fig. 1) $\tilde{\mathbf{X}}_c = W_c(c_j)\tilde{\mathbf{X}}_{c_j} + \sum_{r=i}^{l-1} W_c(c_r)\boldsymbol{\eta}_{c_{r+1}:c_r} + \boldsymbol{\nu}_c$, where $\boldsymbol{\nu}_c$ is composed of state vectors and noise terms outside the route $c_j \stackrel{T_c}{\Longrightarrow} c_i \stackrel{T_c}{\Longrightarrow} c$. The contribution of noise terms to $E[\tilde{\mathbf{X}}_c\tilde{\mathbf{X}}_d^T]$ when processing (c_k, d_k), due to descendent nodes of c_j that are also ancestors of d_k, is therefore $\bar{Q}_1 \doteq \sum_{l=i+1}^{j} W_c(c_{l-1})E[\boldsymbol{\eta}_{c_l:c_{l-1}}\boldsymbol{\eta}_{c_l:c_{l-1}}^T]\sum_u\left(W_d^u(c_{l-1})\right)^T$.

Yet, in addition to the above, the node d_k in T_d may have descendents which also appear as ancestors of c_j in T_c. This situation may be handled in a similar manner. Denote by d_s the descendent of d_k in T_d that is closest to d and appears in $(T_c)^{c_j}$, with $1 < s < k$. Thus, the contribution of noise terms to $E[\tilde{\mathbf{X}}_c\tilde{\mathbf{X}}_d^T]$ when processing (c_k, d_k), due to descendent nodes of d_k that are also ancestors of c_j, is $\bar{Q}_2 \doteq \sum_{m=s+1}^{k}\sum_u W_c^u(d_{m-1})E[\boldsymbol{\eta}_{d_m:d_{m-1}}\boldsymbol{\eta}_{d_m:d_{m-1}}^T]W_d^T(d_{m-1})$.

In conclusion, the overall contribution of noise terms to $E[\tilde{\mathbf{X}}_c\tilde{\mathbf{X}}_d^T]$, when the discovered known pair is (c_j, d_k), is $Q_{c_jd_k} \doteq \bar{Q}_1 + \bar{Q}_2$. An explicit algorithm for calculating $Q_{c_jd_k}$ can be found in [12]. The same technique should be applied in case (c_k, d_j) is the discovered known pair.

D. Computational Complexity

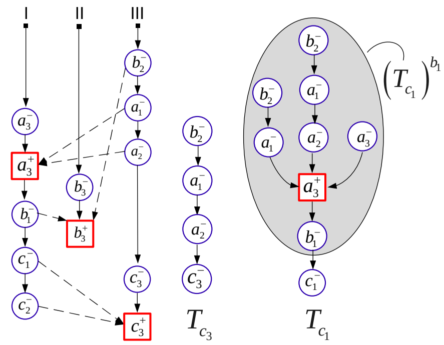
It can be shown [12] that the worst case is bounded by $O\left(n^2 log(rn)\right)$, where n is the number of MR updates represented in G.

V. Example and Experiment Results

In this section, the proposed algorithm is demonstrated for three-view MR updates [9], [8], i. e. $r = 3$.

A. Example

Consider calculation of the term $E[\tilde{\mathbf{X}}_{c_3}^-(\tilde{\mathbf{X}}_{c_1}^-)^T]$ given the graph in Fig. 2(a). The trees $T_{c_3^-}$ and $T_{c_1^-}$ are shown in Fig. 2(b). In this example, $E[\tilde{\mathbf{X}}_{c_3}^-(\tilde{\mathbf{X}}_{c_1}^-)^T]$ may be calculated based on the known term $E[\tilde{\mathbf{X}}_{b_2}^-(\tilde{\mathbf{X}}_{b_1}^-)^T]$, which is analyzed upon reaching the fourth level. As can be seen, among the descendants of the node b_2^- in $T_{c_3^-}$, are the nodes a_1^- and a_2^-. However, these nodes appear also as ancestors of the node b_1^- in $T_{c_1^-}$. Thus, according to Section IV-C, the noise terms associated with the route $b_2^- \to a_1^- \to a_2^-$ are not statistically independent with $\tilde{\mathbf{X}}_{b_1}^-$.



(a) Graph (b) The trees T_{c_3} and T_{c_1}

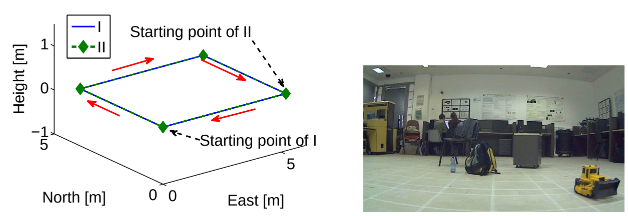
Fig. 2. An example assuming three-view measurements.

Applying the proposed algorithm, the term $E[\tilde{\mathbf{X}}_{c_3}^-(\tilde{\mathbf{X}}_{c_1}^-)^T]$ is calculated as $\Phi_{b_2\to c_3}^{III}E[\tilde{\mathbf{X}}_{b_2}^-(\tilde{\mathbf{X}}_{b_1}^-)^T](\Phi_{b_1\to c_1}^{I})^T + \Phi_{a_2\to c_3}^{III}Q_{a_1\to a_2}^{I}A_2^T(\Phi_{a_3\to c_1}^{I})^T + \Phi_{a_1\to c_3}^{III}Q_{b_2\to a_1}^{III}(A_1 + A_2\Phi_{a_1\to a_2}^{III})^T(\Phi_{a_3\to c_1}^{I})^T$, with $A_1 = -K_{a_3}H_{a_1}$ and $A_2 = -K_{a_3}H_{a_2}$.

B. Experiment Results

The experiment setup consists of a single ground vehicle, attached with a 207MW Axis network camera[3] and MTi-G Xsens IMU/INS[4]. The vehicle was manually commanded, while the camera captured images perpendicular to the motion heading. The IMU and imagery data was recorded for post-processing at 100 Hz and 15 Hz, respectively. These two sources of data were synchronized [9]. The vehicle performed two different trajectories that represent a pattern holding scenario. The hardware (IMU and camera) was turned off between these two trajectories, thereby legitimating to treat each trajectory as if it was performed by a different vehicle, equipped with a similar hardware.

The reference trajectories, manually measured in the experiment, and one of the captured images are given in Fig. 3.



(a) (b)

Fig. 3. Trajectories and one of the captured images in the experiment.

The obtained position and velocity errors for the two vehicles are shown in Fig. 4 (in North, East and Down axes). Three curves are shown: navigation error, square root covariance of the filter, and navigation error in an inertial scenario (given for reference).

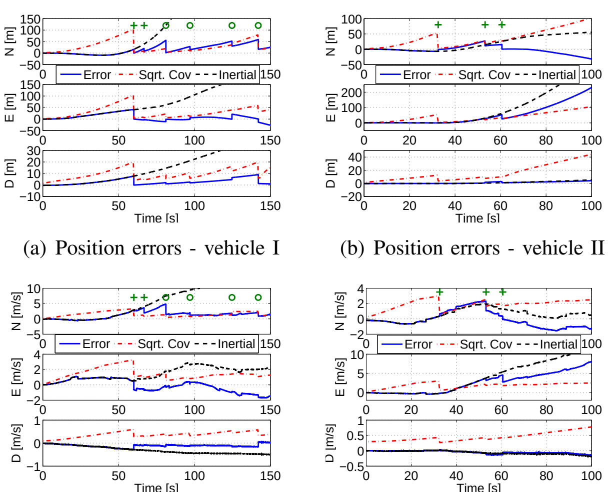
[3]http://www.axis.com/products/cam_207mw/index.htm.

[4]http://www.xsens.com/en/general/mti-g.

The three-view updates [9], [8] were applied for navigation-aiding in two modes: MR-updates, in which 2 of the 3 images (and navigation data) are taken from another vehicle, and Self-update, in which all the 3 images are captured by the same vehicle. Vehicle I is updated twice using data obtained from vehicle II, and 4 times based on its own images. Vehicle II is updated 3 times utilizing the information received from vehicle I. The vehicles performed inertial navigation elsewhere.

The position error was calculated by subtracting the navigation solution from the true trajectories. In a similar manner, the velocity error was computed by subtracting the navigation solution from the true velocity profiles. Since velocity was not measured in the experiment, it was approximately calculated assuming a constant velocity in each phase[5].

As seen, the position error of vehicle I was nearly nullified in all axes as the result of the first update, which was of MV type. The next update (also MV) caused to another reduction in the north position error. After completing a loop in the trajectory, it became possible to apply three-view updates in Self-update mode for vehicle I. In the overall, due to the applied 6 three-view updates, the position error of vehicle I has been confined to around 50 meters in north and east directions, and 10 meters in altitude. As a comparison, the position error of vehicle I in an inertial scenario reaches, after 150 seconds of operation, 900, 200 and 50 meters in north, east and down directions, respectively. The position error of vehicle II has been also dramatically reduced as the result of the MV updates. For example, after the third update, the position error was nearly nullified in north direction and reduced from 50 to 20 meters in east direction. The velocity errors are also considerably reduced in all axes.



(a) Position errors - vehicle I (b) Position errors - vehicle II

(c) Velocity errors - vehicle I (d) Velocity errors - vehicle II

Fig. 4. Position and velocity errors of vehicles I and II in the experiment. MR-updates are denoted by (+), Self-updates are denoted by (o).

[5]The phase duration and the translation that each vehicle has undergone in each phase are known from analyzing the IMU measurements and from the true trajectories.

VI. Conclusions

This paper described a graph-based method for consistent EKF data fusion for distributed cooperative navigation. A general measurement model was assumed, which involves navigation information, obtained from several robots, and the actual readings of the onboard sensors. These different sources of information may represent different time instances. For example, in a three-view vision-based measurement model, a measurement is formulated whenever three images of the same scene are captured by different robots at potentially different time instances, neither of which are a priori known. Since the transmitted information may be statistically dependent, the correlation terms participating in the EKF update step should be known. These terms are computed based upon demand on the proposed graph-based technique.

The method was demonstrated on a three-view measurement model in a theoretical example and in an experiment that involved real imagery and navigation data. Results of pattern holding scenario performed by two vehicles are shown, in which the position and velocity errors of both vehicles are drastically reduced in all axes each time the three-view measurements, along with the calculated correlation terms, are executed.

References

[1] Kurazume, R., Nagata, S. and Hirose, S.,"Cooperative Positioning with Multiple Robots," *Proceedings of the IEEE International Conference on Robotics and Automation*, San Diego, CA, May 1994, pp. 1250–1257.

[2] Roumeliotis, S. I. and Bekey, G. A.,"Distributed Multirobot Localization," *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 5, 2002, pp. 781–795.

[3] Smaili, C., Najjar, M. E. E. and Charpillet, F.,"Multi-sensor Fusion Method Using Bayesian Network for Precise Multi-vehicle Localization," *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, Beijing, China, 2008, pp. 906–911.

[4] Knuth, J. and Barooah, P.,"Distributed collaborative localization of multiple vehicles from relative pose measurements," *Forty-Seventh Annual Allerton Conference*, Illinois, USA, 2009, pp. 314–321.

[5] Sharma, R. and Taylor, C. N.,"Vision Based Distributed Cooperative Navigation for MAVs in GPS denied areas," *Proceedings of the AIAA Infotech@Aerospace Conference*, Washington, USA, April 2009.

[6] Merino, L., Wiklund, J., Caballero, F., Moe, A., Ramiro, J., Forssen, E., Nordberg, K. and Ollero, A.,"Vision-Based Multi-UAV Position Estimation," *IEEE Robotics and Automation Magazine*, September 2006, pp. 53–62.

[7] Kim, B., Kaess, M., Fletcher, L., Leonard, J., Bachrach, A., Roy, N. and Teller, S.,"Multiple Relative Pose Graphs for Robust Cooperative Mapping," *Proceedings of the IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, May 2010.

[8] Indelman, V., Gurfil, P., Rivlin, E. and Rotstein, H., "Distributed Vision-Aided Cooperative Localization and Navigation based on Three-View Geometry," *Proceedings of the IEEE Aerospace Conference*, Montata, USA, March 2011.

[9] Indelman, V., Gurfil, P., Rivlin, E. and Rotstein, H., "Mosaic Aided Navigation: Tools, Methods and Results," *IEEE/ION PLANS*, CA, USA, May 2010, pp. 1212–1225.

[10] Howard, A., Mataric, M. J. and Sukhatme, G. S.,"Putting the 'I' in 'Team' - an Ego-Centric Approach to Cooperative Localization," *Proceedings of the IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, 2003, pp. 868–874.

[11] Bahr, A., Walter, M. R. and Leonard, J. J.,"Consistent Cooperative Localization," *Proceedings of the IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 2009.

[12] Indelman, V., "Navigation Performance Enhancement Using Online Mosaicking," *PhD thesis*, in preparation, Technion, Israel.