2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Distributed Block Coordinate Moving Horizon Estimation for 2D Visual-Inertial-Odometry SLAM

Emilien Flayac[1] Iman Shames[2]

***Abstract*— This paper presents a Visual Inertial Odometry Landmark-based Simultaneous Localisation and Mapping algorithm based on a distributed block coordinate nonlinear Moving Horizon Estimation scheme. The main advantage of the proposed method is that the updates on the position of the landmarks are based on a Bundle Adjustment technique that can be parallelised over the landmarks. The performance of the method is demonstrated in simulations in different environments and with different types of robot trajectory. Circular and wiggling patterns in the trajectory lead to better estimation performance than straight ones, confirming what is expected from recent nonlinear observability theory.**

I. INTRODUCTION

Visual Inertial and Visual Inertial Odometry Simultaneous Localisation and Mapping (VI-SLAM and VIO-SLAM) is the problem of localising a robot in a unknown environment while building a map of it using only visual, inertial and wheel odometry measurements. VI-SLAM has gained a lot of attention in the recent decades due to the low cost and low energy consumption of cameras [7] and the generalisation of Inertial Measurement Unit (IMU). SLAM problems are usually tackled using either filtering techniques or optimisation-based techniques. Typical filtering techniques include Extended Kalman or Information Filters (EKF-EIF) and Particle Filters (PF) [25]. EKFs are cheap and simple to implement but suffer from consistency issues due to successive linearisations and from bad scalability with respect to the number of landmarks considered in the environment [23], [7], [13]. PFs are generally more precise and consistent than EKFs but substantially more computationally costly. Optimisation-based localisation and mapping techniques have recently proved to perform better than filtered-based methods for medium to large problems and at a reduced cost thanks to sparsification techniques [11], [10]. However, proven techniques like Pose SLAM or Graph SLAM are mostly operated offline, [12], [26], while Bundle Adjustment has mainly been applied to purely visual settings, [24], [2]. These methods are usually contained in the broader framework of Full Information Estimation (FIE) and Moving Horizon Estimation (MHE) framework, where the trajectory of a system is recovered by minimising the output error generated by the actual and predicted measurements, under a dynamical constraint. MHE is a simplified version of FIE where the optimisation is only performed on a sliding time window instead of starting from the initial time. Several VI-SLAM algorithms based on this idea have been designed, [21], [9], [22], [20], [15], [16], [8], [3]. However, the structure of the resulting optimisation problem does not seem to have been exploited yet. For example, in order to solve an MHE problem where the variables are the state of the system and fixed independent landmarks, one could iteratively fix the state variables and solve for the landmark variables and vice versa. This technique is called Block Coordinate Descent (BCD), see Chapter 2 of [1]. It has been applied to Visual SLAM for example in [23], PTAM SLAM [17] and ORB-SLAM [3] sometimes under the denomination of *motion-only* problem for state trajectory estimation or *structure only* problem for landmark estimation. In these setup, the structure-only problem is typically high dimensional and become computationally costly. In this paper, we propose a distributed BCD method for Moving Horizon Estimation applied to landmark-based VI-O SLAM that allows one to parallelise the computations of landmark estimates.

Many works in the field of robotics have showed that persistently excited path including circular ones are common sufficient conditions for good estimation using bearing measurements, [5], [4], [19], [14]. Thus, the performance and robustness to noise of the proposed method depend on the trajectory of the robot. It is then demonstrated through simulations in several scenarios with different levels of excitations: a circular path in a circular corridor, a straight path in a straight corridor, a 'snaking' path in a straight corridor. The rest of the paper is organised as follows: Section II describes the dynamical and measurements models considered, Section III presents a batch version of the MHE problem of interest, Section IV presents its block coordinated version, Section V summarises the estimation algorithm and Section VI gives simulation results.

II. DYNAMICAL AND MEASUREMENT MODELS

A. Differential drive model

We consider a mobile robot represented by a 2D position, $x = (x_1, x_2) \in \mathbb{R}^2$ and an orientation $\theta \in \mathbb{R}$. We assume it follows the differential drive dynamics such that:

$$\dot{x}_1 = \frac{\omega_r R_r + \omega_l R_l}{2}\cos(\theta),\ \dot{x}_2 = \frac{\omega_r R_r + \omega_l R_l}{2}\sin(\theta), \qquad (1)$$
$$\dot{\theta} = \frac{\omega_r R_r - \omega_l R_l}{D},$$

[1] Emilien Flayac (emilien.flayac@isae.fr) is with the Department of Complex System Engineering, ISAE-Supaero, Toulouse, France

[2]Iman Shames (iman.shames@anu.edu.au) is with the CIICADA Lab, School of Engineering, The Australian National University, Canberra, Australia.

This work received funding from the Australian Government, via grant AUSMURIB000001 associated with ONR MURI grant N00014-19-1-2571

where $R_r > 0$, $R_l > 0$ and $D > 0$ are respectively the radius of the right wheels, the radius of the left wheels and the distance between the two sets of wheels and $\omega_{ref} = (\omega_r, \omega_l) : \mathbb{R}^+ \to \mathbb{R}^2$ represent the angular velocities of the right and left wheels. By setting $z = (x, \theta)$, we can sum up (1) as follows:

$$\dot{z} = f(z, \omega_{ref}), \tag{2}$$

where $f : \mathbb{R}^3 \times \mathbb{R}^2 \to \mathbb{R}^3$. Let $z_0 \in \mathbb{R}^3$ be an initial condition and $t_0 \geq 0$ be the reference initial time. In the sequel, for $t \geq t_0$, $z(t) = (x(t), \theta(t))$ represents the solution of (2) at time t starting from z_0 with initial time t_0 and input ω_{ref}.

B. Discretization scheme

In the sequel, we consider measurements obtained at discrete points in time with two different sampling rates. With this in mind, let $\Delta_{odo} > 0$ and $\Delta_{vis} > 0$, be respectively the discretization step of odometry and visual measurements. We assume in the rests of the paper that $m = \frac{\Delta_{vis}}{\Delta_{odo}} \in \mathbb{N}^*$. Thus, for $k \geq 0$ and $0 \leq i \leq m$, we define $t_{k,i}$ such that:

$$t_{k,i} = t_0 + k\Delta_{vis} + i\Delta_{odo}. \tag{3}$$

where $t_{k,N} = t_{k+1,0}$ for any $k \geq 0$. When no ambiguity is possible, we will denote $t_{k,0}$ by t_k for any $k \geq 0$.

C. Odometry and Inertial measurements

We assume that one does not have access to ω_{ref} but only to a noisy discretised version through odometry measurements. It is denoted by ω_{odo} and reads:

$$\omega_{odo}(t) = \sum_{k=0}^{+\infty} \sum_{i=0}^{m} \omega_{k,i} \mathbf{1}_{\{t_{k,i} \leq t < t_{k,i+1}\}}, \tag{4}$$

where $\mathbf{1}$ denotes the indicator function, $\omega_{k,i} = \omega_{ref}(t_{k,i}) + d^{odo}_{k,i}$ and $(d^{odo}_{k,i})_{k \geq 0, 0 \leq i \leq m-1}$ is an i.i.d. sequence of centered Gsaussian perturbations with covariance Q^{odo}. Thus, for a sequence $\omega_{k,0:i}$,$0 \leq i \leq m$, the discretized dynamics between t_k and $t_{k,i}$ can be written $z(t_{k,i}) = f_{dis,i}(z(t_k), \omega_{k,0:i})$ for some function $f_{dis,i}$. We also assume that for any $k \geq 0$, inertial measurements are processed and give information on the displacement of z between $t_{k,0}$ and $t_{k+1,0}$ denoted by u_k and defined as follows:

$$u_k = z(t_{k+1,0}) - z(t_{k,0}) + d^{in}_k. \tag{5}$$

where $d^{in}_k \in \mathbb{R}^3$ is a Gaussian perturbation of covariance Q_{in} representing the error caused by the integration of inertial measurements. From (5) and for $t \geq t_0$, one can define $u(t)$ similarly to (4):

$$u(t) = \sum_{k=0}^{+\infty} u_k \mathbf{1}_{\{t_{k,0} \leq t < t_{k+1,0}\}}, \tag{6}$$

D. Landmark-based bearing measurement model

For $J \in \mathbb{N}^*$, let $\ell = \left(\ell^{(j)}\right)_{1 \leq j \leq J} \in (\mathbb{R}^2)^J$ be a collection of J landmarks represented by a 2D position. We assume, that for $1 \leq j \leq J$ and $k \geq 0$, if landmark j is seen by the robot at time t_k, a measurement of the direction between the robot and landmark j in the body frame of the robot is obtained from visual information. It is denoted by $y^{(j)}_k$ and defined formally as follows:

$$y^{(j)}_k = a^{(j)}_k \left(R(-\theta(t_k)) \frac{\ell^{(j)} - x(t_k)}{\|\ell^{(j)} - x(t_k)\|} + v^{(j)}_k \right), \tag{7}$$

where $\|\cdot\|$ denotes the Euclidean norm, $R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$, $v^{(j)}_k \in \mathbb{R}^2$ is a Gaussian perturbation representing the measurement noise, and $a^{(j)}_k$ is a data association parameter such that $a^{(j)}_k = 1$ when landmark j is seen at time t_k and $a^{(j)}_k = 0$ otherwise. We set $y_k = \left(y^{(j)}_k\right)_{1 \leq j \leq J}$ and $v_k = \left(v^{(j)}_k\right)_{1 \leq j \leq J}$ so that (7) can be written as follows:

$$y_k = A_k(h(z(t_k), \ell) + v_k), \tag{8}$$

where for any $k \geq 0$, $h = \begin{bmatrix} h^{(1)} \\ \vdots \\ h^{(J)} \end{bmatrix}$ with $h^{(j)}_k : \mathbb{R}^3 \times \mathbb{R}^2 \to \mathbb{R}^2$ for any $1 \leq j \leq J$, and A_k is the diagonal matrix of appropriate size with repetitions of binary numbers $a^{(j)}_k$ on its diagonal. The covariance matrix of v_k is denoted by R_{vis}. Beside, we consider the sensor-centric view where the initial state is assumed known and can be considered as the origin of the robot frame. The goal of the following is to estimate the state of (2) at time $t_{k,0}$ for any $k \geq 0$, $z(t_{k,0})$, and the position of the collection of landmarks, ℓ knowing the initial state z_0 and time t_0. In fact, estimating z at the all the times $t_{k,i}$ for $k \geq 0$ and $1 \leq i \leq N-1$ woud lead to an intractable number of variables in the optimisation problems.

III. Batch Moving horizon estimation for bearing-only SLAM

A. Discretized formulation of MHE

Fix $N \in \mathbb{N}^*$ be a time horizon. In the sequel, for any $n \geq 1$, we denote by $\mathbb{S}^{++}_n$ the set of positive definite $n \times n$ matrices. Besides, for any $S \in \mathbb{S}^{++}_n$, $\|\cdot\|_S$ denotes the norm weighted by S. For any integer $k \geq 0$, Moving Horizon Estimators are designed to forget about the input and output trajectory before time $k - N$. In this section, we are first interested in the discretized MHE Problem in a batch formulation. Thus, one is jointly looking for a state trajectory $(\zeta_l)_{k-N \leq l \leq k}$ and a vector of landmarks p that match the visual and inertial measurements. Integrating (1) between $t_{k,0}$ and $t_{k+1,0}$ inside an optimisation problem is not computationally tractable. Thus, the state vector $(\zeta_l)_{k-N \leq l \leq k}$ are linked using (5) leading to:

$$\zeta_{l+1} = \zeta_l + u_l + d_l, \quad l = k-N, \dots, k-1,$$

where $(d_l)_{k-N \leq l \leq k-1}$ are noise variable added to take into account the presence of disturbations. It is then important to keep track of the knowledge of the past and weigh it in the optimisation problem through an *arrival cost*. Thus, we assume that a state estimate and a landmark estimate at time $k-N$ respectively denoted by $\hat{z}_{k-N}$ and $\hat{\ell}_{k-N}$ are available.

We also assume that a weighting matrix denoted by Π_{k-N} is available. Its computation is detailed in section III-B.

$$\begin{aligned}&\min_{\zeta_l,d_l,p} \quad \left\| \begin{bmatrix} \zeta_{k-N} - \hat{z}_{k-N} \\ p - \hat{\ell}_{k-N} \end{bmatrix} \right\|^2_{\Pi^{-1}_{k-N}} + \|y_k - A_k h(\zeta_k, p)\|^2_{R^{-1}_{vis}} \\ &+ \textstyle\sum_{l=k-N}^{k-1} \|d_l\|^2_{Q^{-1}_{in}} + \|y_l - A_l h(\zeta_l, p)\|^2_{R^{-1}_{vis}} \\ &\text{s.t.} \quad \zeta_{l+1} = \zeta_l + u_l + d_l, \quad l = k-N, \ldots, k-1, \end{aligned} \tag{$P_{k,ba}$}$$

Note that in Problem $(P_{k,ba})$ the resulting dynamics is a discrete time single integrator whose input are the inertial measurements.

B. Arrival cost computation

The goal of this section is to detail the computation of Π_k for any $k \geq 0$. First, we fix a matrix $\Pi_0 \in \mathbb{S}^{++}_{3+2J}$. Then, for any integer $k \geq 0$, Π_k is computed using the equation of an Extended Kalman Filter by integrating forward the most recent MHE state and landmark estimate. More precisely, if we fix some joint estimate $\hat{\xi}_k = (\hat{z}_k, \hat{\ell}_k) \in \mathbb{R}^{3+2J}$ and a covariance matrix $\Pi_k \in \mathbb{S}^{++}_{3+2J}$ for some $k \geq 0$ then for any $0 \leq i \leq m$, we denote by $\xi^+_{k,i}$ the prediction at $t_{k,i}$ from system (2) with input ω_{odo}. It reads:

$$\xi^+_{k,i} = f_{tot}(\hat{\xi}_k, \omega_{k,0:i}), \tag{9}$$

where $\omega_{k,0:i} = (\omega_{k,0}, \ldots, \omega_{k,i})$ and $f_{tot}(z, \ell, \omega) = \begin{bmatrix} f_{dis,i}(z,\omega) \\ \ell \end{bmatrix}$. From this, one can compute the prediction of the covariance matrix up to time $t_{k,i}$ which is denoted by $\Pi^+_{k,i}$. It is defined recursively as follows for any $0 \leq i \leq m-1$:

$$\Pi^+_{k,0} = \Pi_k, \tag{10}$$

$$\Pi^+_{k,i+1} = F(\xi^+_{k,i})\Pi_{k,i}F^T(\xi^+_{k,i}) + G(\xi^+_{k,i})Q^{odo}G^T(\xi^+_{k,i}), \tag{11}$$

where $F(\xi^+_{k,i}) = \nabla_{(z,\ell)} f_{tot}(\hat{\xi}_k, \omega_{k,0:i})$ and $G(\xi^+_{k,i}) = \nabla_\omega f_{tot}(\hat{\xi}_k, \omega_{k,0:i})$ and ∇f_{tot} representing the differential of f_{tot} with respect to the indicated variables. Then, the Kalman gain K_{k+1} and the corrected covariance matrix Π_{k+1} at time $t_{k,N} = t_{k+1,0}$ are computed using a standard linearised correction step using the visual measurement model (8).

IV. Block coordinate Moving horizon estimation for Bearing-only SLAM

The idea of this section is to present the distributed block coordinate version of the problems $(P_{k,ba})$ where one looks alternatively for the collection of landmarks for a given state trajectory estimate and for a state trajectory for given landmark estimates. In order to decouple state and landmark variables the matrices Π_k are assumed to block diagonal matrices composed of $J+1$ blocks: one 3×3 blocks for state/state correlations only denoted by $\Pi_{zz,k}$, and J 2×2 blocks for one-by-one landmark/landmark correlations denoted by $(\Pi_{\ell^{(j)}\ell^{(j)},k})_{1 \leq j \leq J}$. This assumption implies that the cost in $(P_{k,ba})$ is separable with respect to landmarks for a fixed state trajectory estimate which makes distributed resolution possible.

A. Distributed landmark estimation for a given trajectory

More precisely, let $(\hat{z}_l)_{k-N \leq l \leq k}$ be some estimates of $(z(t_l))_{k-N \leq l \leq k-1}$. By removing constant terms with respect to p, the landmark estimation problem reads:

$$\min_p \quad \left\| p - \hat{\ell}_{k-N} \right\|^2_{\Pi^{-1}_{\ell\ell,k-N}} + \textstyle\sum_{l=k-N}^{k} \|y_l - A_l h(\hat{z}_l, p)\|^2_{R^{-1}_{vis}} \tag{$P_{k,\ell}$}$$

Note that $(P_{k,\ell})$ depends only on the trajectory estimates and not on any dynamics. Besides, if the visual measurement are supposed independent, then R_{vis} is block diagonal with respect to individual landmarks. Since we assumed that $\Pi_{\ell\ell,k-N}$ is block diagonal, $(P_{k,\ell})$ can be split and solved landmark by landmark. For any $1 \leq j \leq J$, the split problem reads:

$$\begin{aligned}&\min_{p^{(j)}} \quad \left\| p^{(j)} - \hat{\ell}^{(j)}_{k-N} \right\|^2_{\Pi^{-1}_{\ell^{(j)}\ell^{(j)},k-N}} \\ &+ \textstyle\sum_{l=k-N}^{k} \|y^{(j)}_l - a^{(j)}_l h^{(j)}(\hat{z}_l, p^{(j)})\|^2_{R^{-1}_{vis,j}}\end{aligned} \tag{$P_{k,\ell,j}$}$$

where $R_{vis,j}$ is the block of R_{vis} corresponding to landmark ℓ^j. Consequently if the landmark j is seen at time k (i.e. $a^{(j)}_k = 1$) then $(P_{k,\ell,j})$ is then solved by a nonlinear programming (NLP) solver using only a fixed number of iterations.

B. State estimation for given landmarks estimates

In this section, for an integer $k \geq 0$, we fix a landmark estimate $\hat{\ell}_k$. Then, the state trajectory estimation subproblem coming from $(P_{k,ba})$ reads:

$$\begin{aligned}&\min_{\zeta_l,d_l} \quad \|\zeta_{k-N} - \hat{z}_{k-N}\|^2_{\Pi^{-1}_{zz,k-N}} + \|y_k - A_k h(\zeta_k, \hat{\ell}_k)\|^2_{R^{-1}_{vis}} \\ &+ \textstyle\sum_{l=k-N}^{k-1} \|d_l\|^2_{Q^{-1}_{in}} + \|y_l - A_l h(\zeta_l, \hat{\ell}_k)\|^2_{R^{-1}_{vis}} \\ &\text{s.t.} \quad \zeta_{l+1} = \zeta_l + u_l + d_l, \quad l = k-N, \ldots, k-1, \end{aligned} \tag{$P_{k,z}$}$$

This problem can also solved approximately by a NLP solver. Similarly to the batch formulation one obtains an estimate $\hat{\xi}_k = (\hat{z}_k, \hat{\ell}_k) \in \mathbb{R}^{3+2J}$.

C. Arrival cost computation

The goal of this section is to detail the computation of a distributed family of the covariance matrices $\Pi_{zz,k}$ and $(\Pi_{\ell^{(j)}\ell^{(j)},k})_{j=1,\ldots,J}$ for any $k \geq 0$ and $\ell = 1, \ldots, J$. First, we fix a matrices $\Pi_{zz,0} \in \mathbb{S}^{++}_3$ and $\Pi_{\ell\ell,0} \in \mathbb{S}^{++}_2$ for $\ell = 1, \ldots, J$. Then, for any $k \geq 0$, the matrices $\Pi_{zz,k}$ and $(\Pi_{\ell^{(j)}\ell^{(j)},k})_{j=1,\ldots,J}$ are computed using the equations of an adhoc distributed Extended Kalman Filter. For conciseness, the matrices $\Pi_{\ell^{(j)}\ell^{(j)},k}$ are renamed $\Pi_{jj,k}$

1) Block Coordinate Prediction step: Similarly to the batch version of the EKF from section III-B, we fix some joint estimate $\hat{\xi}_k = (\hat{z}_k, \hat{\ell}_k) \in \mathbb{R}^{3+2J}$ and a covariance matrices $\Pi_{zz,k} \in \mathbb{S}^{++}_3$ and $(\Pi_{jj,k})_{j=1,\ldots,J}$ for some $k \geq 0$

then for any $0 \leq i \leq m$, we denote by $\xi^+_{k,i}$ the prediction at $t_{k,i}$ from system (2) with input ω_{odo}. It reads:

$$\xi^+_{k,i} = f_{tot}(\hat{\xi}_k, \omega_{k,0:i}), \tag{12}$$

where $\omega_{k,0:i} = (\omega_{k,0}, \dots, \omega_{k,i})$ and $f_{tot}(z, \ell, \omega) = \begin{bmatrix} f_{dis,i}(z,\omega) \\ \ell \end{bmatrix}$. From this, one can compute the prediction of the covariance matrices up to time $t_{k,i}$ which are denoted by $\Pi^+_{zz,k,i} \in \mathbb{S}^{++}_3$ and $(\Pi^+_{jj,k,i})_{j=1,\dots,J}$. It is defined recursively as follows for any $0 \leq i \leq m$, and any $j = 1, \dots, J$:

$$\Pi^+_{zz,k,0} = \Pi_{zz,k}, \qquad \Pi^+_{jj,k,0} = \Pi_{jj,k}, \tag{13}$$

$$\Pi^+_{zz,k,i+1} = F(\xi^+_{k,i})\Pi_{zz,k,i}F^T(\xi^+_{k,i}) + G(\xi^+_{k,i})Q^{odo}G^T(\xi^+_{k,i}), \tag{14}$$

$$\Pi^+_{jj,k,i+1} = \Pi^+_{jj,k,i} \tag{15}$$

where $F(\xi^+_{k,i}) = \nabla_{(z,\ell)} f_{dis,i}(\hat{\xi}_k, \omega_{k,0:i})$ and $F(\xi^+_{k,i}) = \nabla_\omega f_{dis,i}(\hat{\xi}_k, \omega_{k,0:i})$ and $\nabla f_{dis,i}$ representing the differential of f_{tot} with respect to the indicated variables.

2) Distributed Block Coordinate Correction step: First, a block coordinate correction step at time $t_{k,N} = t_{k+1,0}$ is applied using the visual measurements (8). We make an approximation of the visual observation equation:

$$y_{k+1} \approx A_{k+1}(h(\hat{z}^+_k + w^+_k, \ell) + v_{k+1})$$

where $w_k \sim \mathcal{N}(0, \Pi^+_{zz,k})$. Moreover, landmark j is updated at time k only if it is seen at that time i.e. when $a^{(j)}_k = 1$. We assume w^+_k and v_{k+1} are independent and we compute the Kalman gain for each landmark which denoted by $K_{j,k+1}$ and reads for any $j = 1, \dots, J$, if $a^{(j)}_k = 1$ then:

$$K_{j,k+1}(\xi^+_{k,m}) = \Pi^+_{j,k} H^T_{j,k+1}(\xi^+_{k,m}) \tag{16}$$

$$[H_{j,k+1}(\xi^+_{k,m})\Pi^+_{jk} H^T_{j,k+1}(\xi^+_{k,m}) \tag{17}$$

$$+ R_{vis} + H_{z,k+1}(\xi^+_{k,m})\Pi_{zz,k} H^T_{z,k+1}(\xi^+_{k,m})]^{-1}, \tag{18}$$

which leads to the following definition of $\Pi_{jj,k+1}$:

$$\Pi_{jj,k+1} = \Pi_{jj,k} - K_{j,k+1}(\xi^+_{k,m})H_{j,k+1}(\xi^+_{k,m})\Pi_{jj,k}, \tag{19}$$

where $H_{j,k+1}(\xi^+_{k,m}) = a^{(j)}_{k+1}\nabla_{\ell^{(j)}} h^{(j)}(\xi^+_{k,m})$. Otherwise if $a^{(j)}_k = 0$ then $\Pi_{jj,k+1} = \Pi_{jj,k}$.

One now assumes that some updated estimate of the collection of landmarks $\hat{\ell}_{k+1}$ has been computed. We make the following approximation of the visual observation equation

$$y_{k+1} \approx A_{k+1}(h(z_k, \hat{\ell}_{k+1} + w_{\ell,k+1}) + v_{k+1})$$

where $w_{\ell,k+1} \sim \mathcal{N}(0, \Pi^+_{\ell\ell,k})$. Then, we assume $w_{\ell,k+1}$ and v_{k+1} are independent and we compute the Kalman gain for the robot's state which denoted by $K_{z,k+1}$ and reads:

$$K_{z,k+1}(\xi^+_{k,m}) = \Pi^+_{zz,k,m} H^T_{z,k+1}(\xi^+_{k,m}) \tag{20}$$

$$(H_{z,k+1}(\xi^+_{k,m})\Pi^+_{zz,k,m} H^T_{j,k+1}(\xi^+_{k,m}) \tag{21}$$

$$+ R_{vis} + H_{\ell,k+1}\Pi_{\ell\ell,k+1} H^T_{\ell,k+1})^{-1}, \tag{22}$$

which leads to the following definition of $\Pi_{jj,k+1}$:

$$\Pi_{zz,k+1} = \Pi_{zz,k} - K_{z,k+1}(\xi^+_{k,m})H_{z,k+1}(\xi^+_{k,m})\Pi^+_{zz,k,m}, \tag{23}$$

where $H_{\ell,k+1}(\xi^+_{k,m}) = A_{k+1}\nabla_\ell h(\xi^+_{k,m})$.

The assumption that the matrices $\Pi_{k,\ell\ell}$ are block diagonal is strong because it means that one neglects the correlations between landmarks. However, the proposed distributed Kalman covariance update allows one to reintroduce correlation between the landmark and state estimates which seems to be enough to get a good confidence measure on the state and landmarks to be used as an arrival cost in the MHE problems.

V. Algorithm

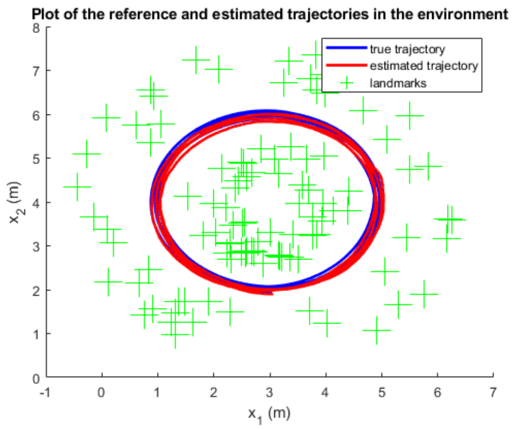
Algorithm 1 Block coordinate Descent in MHE for SLAM

Fix $\hat{z}_0 = z_0$ and $\xi_0 = z_0$
Choose $\hat{\ell}_0$ and Π_0
Get y_0
for $k = 1, 2, \dots$ **do**
  Get y_k.
  Compute Π_k.
  if $k < N$ **then**
    Get $\hat{z}_k$ by integrating (2) from t_{k-1} to t_k with input ω_{odo}.
    for $j = 1, \dots, J$ **do**
      if $a^{(j)}_k = 1$ **then**
        Solve $(P_{k,\ell,j})$ at $(\hat{z}_0, \dots, \hat{z}_k)$ by setting $N = k + 1$ and get an optimal solution $p^{(j)*}$.
        $\hat{\ell}^{(j)}_k \leftarrow p^{(j)*}$.
      end if
    end for
  else
    Compute $\hat{z}^+_k$ using (2) with input ω_{odo}.
    for $j = 1, \dots, J$ **do**
      if $a^{(j)}_k = 1$ **then**
        Solve $(P_{k,\ell,j})$at $(\hat{z}_{k-N}, \dots, \hat{z}_{k-1}, z^+_{k-1})$ and get an optimal solution $p^{(j)*}$.
        $\hat{\ell}^{(j)}_k \leftarrow p^{(j)*}$.
      end if
    end for
    Solve $(P_{k,z})$ at $\hat{\ell}_k$ and get $\zeta^*_{k-N}, \dots, \zeta^*_k$.
    $\hat{z}_k \leftarrow \zeta^*_k$.
  end if
end for

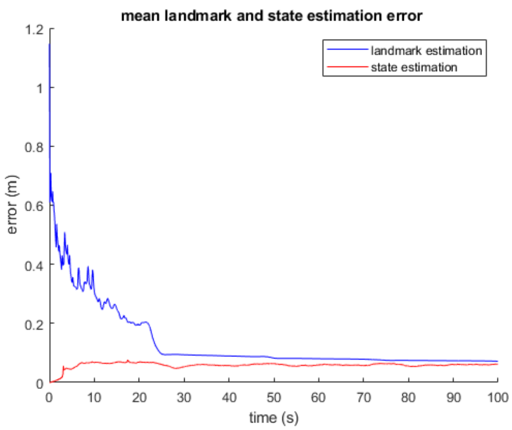
The resulting algorithm is summarised in Algorithm 1.

Remark 5.1: The main advantage of Algorithm 1 is that the landmark update can be parallelised since both the MHE problem $(P_{k,\ell,j})$ and the Kalman update for the arrival cost (16)-(19) are distributed over the landmark variables.

Since the landmarks that are not seen at time k are not updated, loop closure does not allow the filter to correct every landmark at the same time. However, as shown in Section VI, several loops ensures that all the map is properly updated.



(a)



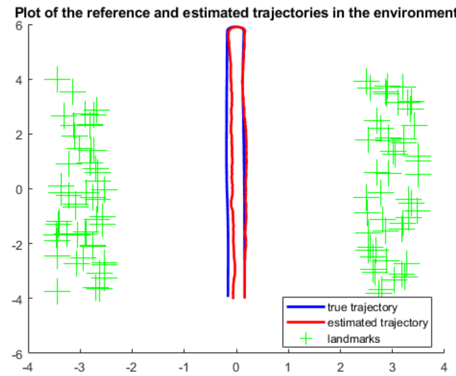
(b)

Fig. 1: Plots of a sample circular trajectory, its estimation and the corresponding average landmark and state estimation errors for 50 Monte-Carlo simulations.

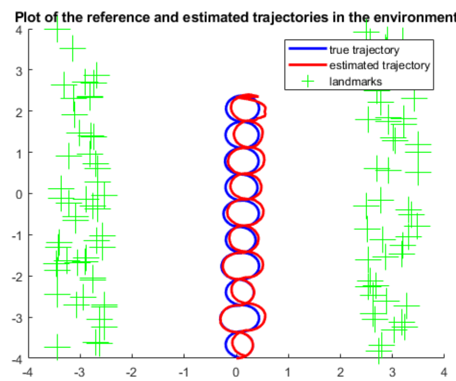
A key factor for good performance of such Moving Horizon Estimation schemes are observability conditions, see [6], [18]. Because of the nonlinearities in the measurement and dynamic equations, observability properties depend on the trajectory of the extended system state/landmark and so might the estimation error. Thus, the goal of Section VI is to illustrate the results of the proposed estimation algorithm for different robot trajectories and different environments.

VI. Simulations

In this section, we present simulations of a 2D environments with with 2 configurations of $J = 100$ landmarks and 3 types of trajectories, a circular one, a straight one and one with wiggles. Figure 1a shows a example of circular trajectory with several loops with landmarks dispatched in a double ring. The parameters of the robot from (1) have been chosen as characteristics of a standard Jackal robot knowing: $D = 0.043$, $R_r = R_l = 0.1$. The noise covariance have been set as follows: $R_{vis} = 0.001I$, $Q_{in} = 10^{-4}\Delta_{vis}I_3$, and $Q_{odo} = 0.009I_2$, where I denotes the identity matrix of appropriate dimension. The parameters $a_k^{(j)}$, representing data association are assumed to be given without error, for



(a)



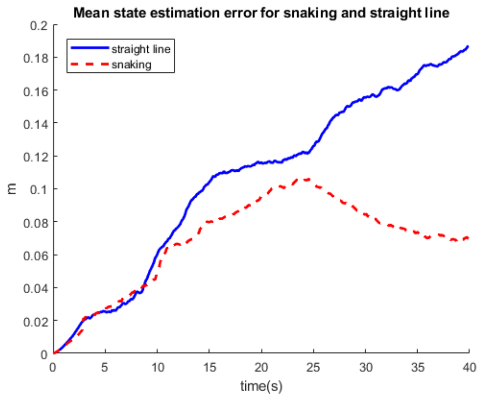
(b)

Fig. 2: Plots of a sample straight and a wiggling trajectory with their estimates in the same environment

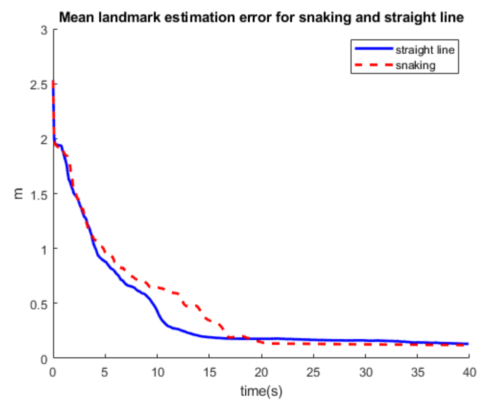
any $1 \leq j \leq J$ and any $k \geq 0$. Besides, a maximal range has been implemented on the bearing sensor through the variables $a_k^{(j)}$. It is of $2m$ for the circular scenarios and $3.6m$ for the two others. Running times have not been included since code optimisation is not the topic of this paper and the actual parallelisation process of the distributed scheme has not been implemented yet. The performance of the proposed method in this case is demonstrated in Figure 1b where both the state and mean landmark estimation error are converging to a small value. Note that the initial state estimation error is assumed to be zero since the initial position and orientation of the robot is assumed to be known. Observability theory coming from circumnavigation [5], [4], [19], [14], [6] suggest that circular patterns should improve estimation performance. Figure 2a and 2b show an example of a back-and-forth straight and snaking trajectory in a corridor-like environment with landmarks on the side only. Figure 3a and 3b show that, as expected, the wiggling patterns are helping the estimation process which result in a smaller state estimation error than in the case of a straight trajectory.

VII. Conclusion

In this paper, a block coordinated Moving Horizon Estimation algorithm for Visual Inertial Odometry SLAM is



(a)



(b)

Fig. 3: Plots of the average state and landmark errors in the case of a straight and wiggling trajectory in the same environment after 50 Monte-Carlo simulations.

proposed. It is leveraging ideas coming from Bundle Adjustement, nonlinear estimation and nonlinear programming in order to make the estimation process distributed over the landmarks. The performance of the proposed method is demonstrated through simulations in several environments, with several robot trajectories in the presence of noise.

References

[1] Dimitri P Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3):334–334, 1997.

[2] Alvaro Parra Bustos, Tat-Jun Chin, Anders Eriksson, and Ian Reid. Visual SLAM: Why Bundle Adjust? In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2385–2391, Montreal, QC, Canada, May 2019. IEEE.

[3] Carlos Campos, Richard Elvira, Juan J. Gomez Rodriguez, Jose M.M Montiel, and Juan D. Tardos. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, December 2021.

[4] S. de Marco, M-D. Hua, T. Hamel, and C. Samson. Position, Velocity, Attitude and Accelerometer-Bias Estimation from IMU and Bearing Measurements. In *2020 European Control Conference (ECC)*, pages 1003–1008, Saint Petersburg, Russia, May 2020. IEEE.

[5] Mohammad Deghat, Iman Shames, Brian D. O. Anderson, and Changbin Yu. Localization and Circumnavigation of a Slowly Moving Target Using Bearing Measurements. *IEEE Transactions on Automatic Control*, 59(8):2182–2188, August 2014.

[6] Emilien Flayac and Iman Shames. Non-uniform observability for moving horizon estimation and stability with respect to additive perturbation. *arXiv preprint arXiv:2104.12328*, 2021.

[7] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial Intelligence Review*, 43(1):55–81, January 2015.

[8] Sejong Heo, Jaehyuck Cha, and Chan Gook Park. EKF-Based Visual Inertial Navigation Using Sliding Window Nonlinear Optimization. *IEEE Transactions on Intelligent Transportation Systems*, 20(7):2470–2479, July 2019.

[9] Timo Hinzmann, Thomas Schneider, Marcin Dymczyk, Andreas Schaffner, Simon Lynen, Roland Siegwart, and Igor Gilitschenski. Monocular visual-inertial slam for fixed-wing uavs using sliding window based nonlinear optimization. In *Advances in Visual Computing: 12th International Symposium, ISVC 2016, Las Vegas, NV, USA, December 12-14, 2016, Proceedings, Part I 12*, pages 569–581. Springer, 2016.

[10] Shoudong Huang. A review of optimisation strategies used in simultaneous localisation and mapping. *Journal of Control and Decision*, 6(1):61–74, January 2019.

[11] Shoudong Huang and Gamini Dissanayake. A critique of current developments in simultaneous localization and mapping. *International Journal of Advanced Robotic Systems*, 13(5):172988141666948, September 2016.

[12] V. Ila, J.M. Porta, and J. Andrade-Cetto. Information-Based Compact Pose SLAM. *IEEE Transactions on Robotics*, 26(1):78–93, February 2010.

[13] Yujiao Jia, Xinying Yan, and Yihan Xu. A Survey of simultaneous localization and mapping for robot. In *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pages 857–861, Chengdu, China, December 2019. IEEE.

[14] Tor A Johansen and Edmund Brekke. Globally Exponentially Stable Kalman Filtering for SLAM with AHRS. page 8.

[15] Marco Karrer, Patrik Schmuck, and Margarita Chli. CVI-SLAM—Collaborative Visual-Inertial SLAM. *IEEE Robotics and Automation Letters*, 3(4):2762–2769, October 2018.

[16] Yosuke Kishimoto, Kiyotsugu Takaba, and Asuka Ohashi. Moving Horizon Multi-Robot SLAM Based on C/GMRES Method. In *2019 International Conference on Advanced Mechatronic Systems (ICAMechS)*, pages 22–27, Kusatsu, Shiga, Japan, August 2019. IEEE.

[17] Georg Klein and David Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 1–10, Nara, Japan, November 2007. IEEE.

[18] Sven Knufer and Matthias A. Muller. Time-Discounted Incremental Input/Output-to-State Stability. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 5394–5400, Jeju, Korea (South), December 2020. IEEE.

[19] Florent Le Bras, Tarek Hamel, Robert Mahony, and Claude Samson. Observers for position estimation using bearing and biased velocity information. In *Sensing and Control for Autonomous Vehicles: Applications to Land, Water and Air Vehicles*, pages 3–23. Springer, 2017.

[20] Haomin Liu, Mingyu Chen, Guofeng Zhang, Hujun Bao, and Yingze Bao. ICE-BA: Incremental, Consistent and Efficient Bundle Adjustment for Visual-Inertial SLAM. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1974–1982, Salt Lake City, UT, June 2018. IEEE.

[21] Todd Lupton and Salah Sukkarieh. Visual-Inertial-Aided Navigation for High-Dynamic Motion in Built Environments Without Initial Conditions. *IEEE Transactions on Robotics*, 28(1):61–76, February 2012.

[22] Raul Mur-Artal and Juan D. Tardos. Visual-Inertial Monocular SLAM with Map Reuse. *IEEE Robotics and Automation Letters*, 2(2):796–803, April 2017. arXiv: 1610.05949.

[23] Hauke Strasdat, J.M.M. Montiel, and Andrew J. Davison. Visual SLAM: Why filter? *Image and Vision Computing*, 30(2):65–77, February 2012.

[24] Takafumi Taketomi, Hideaki Uchiyama, and Sei Ikeda. Visual SLAM algorithms: a survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, 9(1), December 2017.

[25] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, March 2002.

[26] Sebastian Thrun and Michael Montemerlo. The Graph SLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures. *The International Journal of Robotics Research*, 25(5-6):403–429, May 2006.