Exploitation of the Conditionally Linear Structure in Visual-Inertial Estimation

Zoran Sjanic*,† and Martin A. Skoglund**,†
*Saab AB, Saab Aeronautics, Linköping, Sweden
**Eriksholm Research Centre, part of Oticon A/S, Rørtangvej 20, DK-3070 Snekkersten, Denmark
†Division of Automatic Control, Department of Electrical Engineering, Linköping University, Linköping, Sweden
Email: zoran.sjanic@liu.se, martin.skoglund@liu.se

***Abstract*—In this work, estimators for platform pose and landmark maps for visual-inertial data are analysed. It is shown that the full, non-linear, visual-inertial problem has a conditionally linear substructure in the 2D case which can be exploited for efficient solutions, e.g., Block Coordinate Descent (BCD). It is also shown that the measurement noise from the non-linear model becomes parameter dependent resulting in biased estimates if that fact is ignored. However, the bias can be accounted for using the Iteratively Reweighted Least Squares (IRLS) method. In the 3D case the conditionally linear substructure is not separable. However, it can be shown that the Jacobian of the non-linear substructure can be calculated recursively resulting in an efficient solution. A simulated 2D visual-inertial example is used to illustrate the theoretical results.**

***Index Terms*—Visual-Inertial Estimation, Iteratively Reweighted Least Squares, SLAM**

I. Introduction

Mobile platforms with visual-inertial sensing are common in robotics and autonomous vehicles, and are used in applications and research. They enable odometry, Structure from Motion (SfM), tracking, and Simultaneous Localization and Mapping (SLAM) with applications in e.g., decision support and control. The inertial measurement unit (IMU) can forecast the platform's orientation at high sampling rate yielding better predictions for visual feature tracking, while vision-based orientation estimation can mitigate bias and drift errors inherent in inertial navigation.

For monocular vision systems the scale of the scene cannot be recovered without additional information. However, when combined with an IMU it can be used to estimate the pose and to recover the metric scale, a topic which have been studied by [1]–[3] and others. It is shown in [1] that given noise-free measurements of a landmark from five distinct vantage points, with known platform rotation, it is possible to recover the landmark position, the platform's position and velocity, and accelerometer bias in closed form. A similar idea was explored in [4] parametrizing the estimator with relative motion constraint between poses in a sparse representation of the trajectory and it is referred to as pre-integration.

Similarly, camera observations from several vantage points can be used to derive keyframes [5], [6] effectively downsampling the pose trajectory [7]. In the keyframes approach, camera measurements are used to express relative pose constraints, or feature-based constraints, rather than direct parametrized by landmarks. Such structure-free solutions [8] still produce internal estimates of landmark position, using non-linear least-squares (NLS), which are needed to introduce the feature-based constraints. The combination of structure-free SLAM utilizing pre-integration was explored in [9]–[11].

Estimation and initialization of variables such as landmarks has been considered extensively in computer vision. Polynomial methods can have closed-form solutions and are popular when computational speed is key, for instance using RANSAC outlier rejection. Another class are linear (or affine) formulations of e.g., essential and fundamental matrices, points, lines, and SFM see e.g., [12], [13] which are solved using least-squares or eigen-value methods. In [14] a comparison between polynomial, linear, and iterative linear estimators is done on camera-based triangulation problems. The linear methods may fail to converge [14] if, for example, the baseline is very short, which in turn results in a poorly conditioned system. These methods may even not converge with standard projection error minimization. Linear methods are often algebraic in the sense that they do not consider minimisation of the measurement errors since the projective model is not directly used to form the error resulting in parameter dependent noise. However, linear solutions can also minimise the re-projection error if the correct weighting is used [13], [15]. This can be treated using Iteratively Reweighted Least Squares (IRLS), see e.g., [16] by alternating parameter and noise covariance updating in an iterative fashion.

In this contribution we present two conditionally linear models coupling motion and landmarks in visual/inertial estimation. As an extension to previous work [1], which only consider noise-free models, the linear formulations implicate parameter-dependent noise which can be efficiently treated using IRLS. We suggest a chain of linear solutions combining motion, landmark, and noise estimation using IRLS and Block Coordinate Descent (BCD) and this could be used as either an initialization method for a non-linear solver or as an approximate solution on its own. The approach is numerically analysed in a 2D examples showing reduced bias and improved consistency. Furthermore, an efficient recursive Jacobian calculation is suggested for the corresponding 3D case.

II. System Definition

The proposed model is, more or less, standard for visual-inertial estimation of the platform and the observed envi-

This work has been supported by the Industry Excellence Center LINK-SIC founded by The Swedish Governmental Agency for Innovation Systems (VINNOVA) and Saab AB.

ronment. Its formulation is in a batch form, i.e., all the measurements are collected beforehand and available during estimation. For various reasons, an initial restriction is to consider a 2D formulation. First of all, 2D results are easier to visualise and understand. Second, as we will show below, the iterative solution for the 2D case has some attractive properties that we will exploit. An expansion to the 3D case will not be done here, but some insights into its properties will be given in Section VI.

It is assumed that the platform is moving in an environment parametrised by point landmarks. The data are both visual and inertial measurements that are used to estimate the motion and the environment. Parametrisation of a platform's motion is expressed with its position and rotation relative to a world fixed frame for each time step, and the landmarks are parametrised with their 2D position relative to the same world fixed frame

$$p_t = [p_t^x \; p_t^y]^T \tag{1a}$$
$$R_t = \begin{bmatrix} R_t^{11} & R_t^{12} \\ R_t^{21} & R_t^{22} \end{bmatrix} \tag{1b}$$
$$l_i = [l_i^x \; l_i^y]^T \tag{1c}$$
$$t \in \{0, \ldots, N\} \tag{1d}$$
$$i \in \{1, \ldots, M\} \tag{1e}$$

where p_t is the position, R_t is the orientation of the platform, l_i is the landmark vector, and the number of time steps is $N+1$ and the number of landmarks is M. In visual-inertial setups, the parameters are usually not directly measured, for example, projections of landmarks via images and motion parameters measured with inertial consiting of the derivatives, i.e., velocities and accelerations. To simplify the further analysis in this paper, we will assume that the angular, ω, and linear, $v = [v^x \; v^y]^T$, velocities in the body fixed frame are measured. The connection between the velocity and the position is described by a simple first order dynamics as

$$p_t = p_{t-1} + T_s v_t, \; t \in \{1, \ldots, N\} \tag{2}$$

where T_s is the sampling time. This can be rewritten in a more compact form containing only initial position and the velocities as

$$p_t = p_0 + T_s \sum_{k=1}^{t} v_k, \; t \in \{1, \ldots, N\}. \tag{3}$$

Without loss of generality, we shall assume the the initial position is put in the origin, i.e., $p_0 = [0 \; 0]^T$, since it is arbitrary.

The rotation in (1) is parametrised with a rotation matrix that is dependent on only one parameter, rotation angle θ, and the rotation matrix is uniquely determined by it as

$$R_t = \begin{bmatrix} R_t^{11} & R_t^{12} \\ R_t^{21} & R_t^{22} \end{bmatrix} = \begin{bmatrix} \cos(\theta_t) & \sin(\theta_t) \\ -\sin(\theta_t) & \cos(\theta_t) \end{bmatrix}. \tag{4}$$

With this in mind, we can use the 2D unit quaternion defined as

$$q_t = [q_t^1 \; q_t^2]^T \tag{5a}$$
$$q_t^1 = \cos(\theta_t) \tag{5b}$$
$$q_t^2 = \sin(\theta_t) \tag{5c}$$
$$\|q_t\| = 1 \tag{5d}$$

to parametrise rotation matrix as

$$R_t = \begin{bmatrix} q_t^1 & q_t^2 \\ -q_t^2 & q_t^1 \end{bmatrix}. \tag{6}$$

This parametrisation is advantageous when the relationship between the measured angular velocity, ω_t, and the rotations is established. To obtain it, we will start with the rotational dynamics in continuous time that can be expressed as

$$\dot{q}_t^1 = \frac{\mathrm{d}}{\mathrm{d}t}\cos(\theta_t) = -\sin(\theta_t)\dot{\theta}_t = -q_t^2 \omega_t \tag{7a}$$
$$\dot{q}_t^2 = \frac{\mathrm{d}}{\mathrm{d}t}\sin(\theta_t) = \cos(\theta_t)\dot{\theta}_t = q_t^1 \omega_t \tag{7b}$$

where the relationship $\omega_t = \dot{\theta}_t$ is used. Now we can express the dynamics as

$$\dot{q}_t = \begin{bmatrix} 0 & -\omega_t \\ \omega_t & 0 \end{bmatrix} q_t = S(\omega_t) q_t. \tag{8}$$

Note that this bi-linear relationship also holds for general rotation matrix $\dot{R}_t = S(\omega_t) R_t$, but this is not a minimal representation. We can now employ exact sampling to obtain the dynamics for the discrete time (given that the angular velocity is assumed constant between the sampling times) as

$$q_t = e^{T_s S(\omega_t)} q_{t-1}. \tag{9}$$

By using the Euler forward sampling approximation, the dynamics in (9) can be written as

$$q_t = q_{t-1} + T_s S(\omega_t) q_{t-1} = \big(I + T_s S(\omega_t)\big) q_{t-1}. \tag{10}$$

It is assumed that the velocities are measured without any bias, and that the projection of the landmarks in an image plane with the 2D correspondence are bearing (or visual) measurements. The measurements are assumed to be influenced by additive white Gaussian. Measurements are denoted y with superscript denoting the measured quantity an a model is thus defined as

$$y_t^v = v_t^b + e_t^v = [v_t^{b,x} \; v_t^{b,y}]^T + [e_t^{v,x} \; e_t^{v,y}]^T \tag{11a}$$
$$y_t^r = \omega_t + e_t^r \tag{11b}$$
$$y_t^{j_t} = \frac{u_t^{j_t,y}}{u_t^{j_t,x}} + e_t^L \tag{11c}$$
$$u_t^{j_t} = R_t(l_{j_t} - p_t) = R_t \delta_t^{j_t} \tag{11d}$$
$$e_t^v \sim \mathcal{N}\left(0, \Sigma^v\right) \tag{11e}$$
$$e_t^r \sim \mathcal{N}\left(0, \Sigma^r\right) \tag{11f}$$
$$e_t^L \sim \mathcal{N}\left(0, \Sigma^L\right) \tag{11g}$$
$$t \in \{1, \ldots, N\} \tag{11h}$$
$$j_t \subseteq \{1, \ldots, M\}. \tag{11i}$$

Note that all landmarks are not necessarily observed at each time step and hence landmark index j_t that is time dependent. Which landmarks are observed at a certain time by which measurement is encoded by the data association, which we assume is given in this paper. The total number of landmark measurements as is given by

$$J = \sum_{t=1}^{N} |j_t| \tag{12}$$

where $|j_t|$ denotes the number of landmark measurements at time t. The Gaussian noise is assumed i.i.d. and white with time independent covariance. Furthermore, the world related velocity v_t is not the measured body related one, v_t^b, but they are connected via the rotation as

$$v_t^b = R_t v_t. \tag{13}$$

Given all the measurements, a non-linear equation system can be formulated by stacking the equations from (11) and the resulting non-linear least squares problem is

$$\min_{v_{1:N},\omega_{1:N},l_{1:M}} \|Y - F(v_{1:N}, \omega_{1:N}, l_{1:M})\|^2_{\Sigma^{-1}} \tag{14a}$$

where Y and F are resulting stacked left and right hand sides of (11), and Σ is block diagonal matrix consisting of measurement covariances. Note that the system dynamics from (2) and (10), which can be interpreted as constraints, are embeded into the minimisation formulation above. This problem can be solved with, for instance, the Levenberg-Marquardt method [17], [18].

III. Linear Problem Formulation

In this section it is shown how the system in (11) can be rewritten by observing that the general form of the considered systems are with measured quotients of the unknowns. For example, a quotient system with unknown $x = [x_1, x_2]^T$, given some functions f_1 and f_2, measurement y, and noise e, can be expressed and rewritten as

$$y = \frac{f_1(x_1)}{f_2(x_2)} + e \Longrightarrow f_2(x_2)y = f_1(x_1) + f_2(x_2)e. \tag{15}$$

In the simplistic linear case with $f_1 = f_2 = I$ then (15) becomes

$$\begin{bmatrix}0 & y\end{bmatrix} x = \begin{bmatrix}1 & 0\end{bmatrix} x + \begin{bmatrix}0 & e\end{bmatrix} x \Longleftrightarrow Yx = Hx + Ex \tag{16}$$

where the noise, Ex, now becomes correlated with the parameters. In [1] observations are made that relate (15) and (11), i.e., that the rearranged equation system is conditionally linear given the rotations. In this case the functions $f_1 = u_t^{j_t,y}$ and $f_2 = u_t^{j_t,x}$ from (11c) becomes linear (or affine) function of the unknowns. This follows directly by rearranging the equations that are not related to angular velocity in (11) as

$$y_t^v = v_t^b + e_t^v = R_t v_t + e_t^v \tag{17a}$$

$$\begin{aligned} y_t^{j_t} &= \frac{u_t^{j_t,y}}{u_t^{j_t,x}} + e_t^L \;\Rightarrow \\ y_t^{j_t} u_t^{j_t,x} &= u_t^{j_t,y} + u_t^{j_t,x} e_t^L \Rightarrow \\ y_t^{j_t}(q_t^1 \delta_t^{j_t,x} + q_t^2 \delta_t^{j_t,y}) &= -q_t^2 \delta_t^{j_t,x} + q_t^1 \delta_t^{j_t,y} + u_t^{j_t,x} e_t^L \end{aligned} \tag{17b}$$

$$\delta_t^{j_t,x} = l_{j_t}^x - p_t^x = l_{j_t}^x - T_s \sum_{k=1}^{t} v_k^x \tag{17c}$$

$$\delta_t^{j_t,y} = l_{j_t}^y - p_t^y = l_{j_t}^y - T_s \sum_{k=1}^{t} v_k^y \tag{17d}$$

resulting in an equation system that is linear in parameters, and where rotations and measurements are now coefficients in the system matrix. We can now define all matrices and vectors based on (17) where Y^L are the visual measurements, Y^v are the inertial body velocity measurements, X are the linear motion parameters, Q are the rotation parameters and E^v and $E^L(X,Q)$ are the (possibly linear and rotation parameter dependent) noise terms defined as

$$Y^L = [y_1^{j_1}, \ldots, y_N^{j_N}]^T \in \mathbb{R}^J \tag{18a}$$
$$Y^v = [y_1^{v,x}, \ldots, y_N^{v,y}]^T \in \mathbb{R}^{2N} \tag{18b}$$
$$E^L = [u_1^{j_1,x} e_1^L, \ldots, u_N^{j_N,x} e_N^L]^T \in \mathbb{R}^J \tag{18c}$$
$$E^v = [e_1^{v,x}, \ldots, e_N^{v,y}]^T \in \mathbb{R}^{2N} \tag{18d}$$
$$Q = [q_0^1, q_0^2, \ldots, q_N^1, q_N^2]^T \in \mathbb{R}^{2N} \tag{18e}$$
$$X = [v_1^x, v_1^y, \ldots, v_N^x, v_N^y, l_1^x, l_1^y \ldots, l_M^x, l_M^y]^T \in \mathbb{R}^{2(N+M)}. \tag{18f}$$

This leads to a new, this time linear, equation system

$$A(Y^L, Q)X = [Y^v \;\; 0]^T + [E^v \;\; E^L(X,Q)]^T = \bar{Y}^v + \bar{E}^v \tag{19}$$

where $A(Y^L, Q) \in \mathbb{R}^{(2N+J)\times 2(N+M)}$ is the system matrix with coefficients that depend on the visual measurements and the rotations. In order to show the principal structure of this matrix some short-hand definitions obtained by rearranging terms in (17) are

$$g_t^{j_t,x} = y_t^{j_t} q_t^1 + q_t^2 \tag{20a}$$
$$g_t^{j_t,y} = y_t^{j_t} q_t^2 - q_t^1 \tag{20b}$$
$$h_t^{j_t,x} = -T_s(y_t^{j_t} q_t^1 + q_t^2) \tag{20c}$$
$$h_t^{j_t,y} = -T_s(y_t^{j_t} q_t^2 - q_t^1) \tag{20d}$$

The matrix A in (19) then becomes

$$\left[\begin{array}{cccccc|ccccc}
q_1^1 & q_1^2 & & & & & & & & & \\
-q_1^2 & q_1^1 & & & & & & & & & \\
 & & \ddots & & & & & & & & \\
 & & & q_N^1 & q_N^2 & & & & & & \\
 & & & -q_N^2 & q_N^1 & & & & & & \\
\hline
h_1^{1,x} & h_1^{1,y} & & & & & g_1^{1,x} & g_1^{1,y} & & & \\
\vdots & \vdots & \ddots & & & & \vdots & \vdots & \ddots & \vdots & \vdots \\
h_N^{M,x} & h_N^{M,y} & \cdots & h_N^{M,x} & h_N^{M,y} & & & & & g_1^{M,x} & g_1^{M,y}
\end{array}\right]. \quad (21)$$

Different blocks are related to inertial and visual measurements as well as velocity and landmark parameters. The covariance of the stacked noise is denoted as $\bar{\Sigma}_v = \text{diag}(\Sigma^v,\ \Sigma^L(X, Q))$. Since the equation system in (19) is linear in the unknown parameters an ordinary Weighted Least Squares (WLS) method can be usedt to solve for them as

$$\hat{X} = (A^T \bar{\Sigma}_v^{-1} A)^{-1} A^T \bar{\Sigma}_v^{-1} \bar{Y}^v \quad (22)$$

where some arguments are dropped for the sake of readability. In [1] all the algebraic manipulations are done without considering the measurement noise present, i.e., setting $\bar{\Sigma}_v = I$, which can be a fair approximation. However, since the noise for the visual measurements becomes parameter-dependent in the linear estimator, it can influence the performance.

Another issue that should be investigated is errors in the assumed rotation and its influence on the solution. This is an interesting question, since it cannot always be assume that the true rotation is available, but instead some kind of estimate or, in the worst case, pure guess must be used. In order to explore this an empirical analysis of the estimator is performed.

Ideally we would like to calculate the expected value of the estimate $\hat{X}$ with respect to both the measurements and the rotations, i.e., $\mathbb{E}(\hat{X}|Y^L, Y^v, R)$, both using, and without, the covariance weighting. Unfortunately, the expectation does not have a simple analytical solution, and instead a Monte Carlo (MC) simulation is implemented for empirical analysis in a simple setup. This will serve as an illustration of the estimator performance, and not a general result. In Section V similar performance will be seen in other examples, further strengthening the assumed properties of the estimator consistency.

A. The Importance of Noise

In the example setup, only one landmark is present and it is observed from three positions. The initial position and rotation are assumed known. 1000 randomly generated noise realisations for the visual and inertial measurements, as well as, 1000 randomly generated rotations around the true one. Te noise covariance used as a weight, i.e., $\bar{\Sigma}_v$ in (22), and the true positions of the platform and the landmark are used. This is, of course, not possible to do in the real case, but is used here for the sake of analysis. The result of the simulation is illustrated in Fig. 1 where the estimate of the landmark is depicted. It can be seen that the estimate that uses the covariance is consistent and has a mean that is close to the true landmark position. The estimate that is not using the covariance weighting has a mean

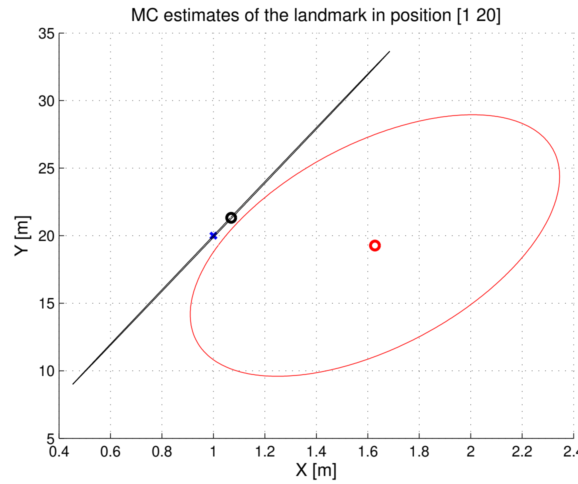


Figure 1: MC estimated mean and covariance (50% 2D-confidence) of the linear system with covariance based weights (in black) and without (in red).

that is biased, its covariance is much larger and without covering the true landmark position, i.e., an inconsistent estimate. Unsurprisingly, the analysis shows that the noise covariance improves the estimator and should be considered further on. However, in the example above, the true parameter values in the noise covariance, which of is not necessarily available. A solution to this issue via an iterative method is introduced in the next section.

B. Conditionally Linear Rotation Dependent System

It can be observed that the system in (17) is conditionally linear in the rotation variables q_t^1 and q_t^2 conditioned on the landmark and position variables. The angular velocity measurements can be used by modifying (10) with the addition of the noise as in

$$\begin{aligned}
q_t =& q_{t-1} + T_s S(\omega_t) q_{t-1} = \\
& q_{t-1} + T_s S(y_t^r - e_t^r) q_{t-1} = \\
& \big(I + T_s S(y_t^r)\big) q_{t-1} - T_s S(e_t^r) q_{t-1} = \\
& \big(I + T_s S(y_t^r)\big) q_{t-1} + T_s [q_{t-1}^2 \;\; -q_{t-1}^1]^T e_t^r \\
& \big(I + T_s S(y_t^r)\big) q_{t-1} + \bar{e}_t^r
\end{aligned} \quad (23)$$

where new noise $\bar{e}_t^r$ is now dependent on rotations q_{t-1}. This implies that we can formulate a new linear system where the rotations are unknown by combining visual measurements from (17) and (23), this time with fixed positions and landmarks, i.e., fixed δ_t. Similarly as before, define

$$Y^r = [y_1^r, \ldots, y_N^r]^T \in \mathbb{R}^N \quad (24a)$$

$$E^r = [\bar{e}_1^{r,1}, \ldots, \bar{e}_N^{r,2}]^T \in \mathbb{R}^{2N} \quad (24b)$$

and the system matrix $B(Y^r, Y^L, X) \in \mathbb{R}^{(J+2N) \times 2N}$. This system will have very similar structure to (19) as in

$$\begin{aligned}
B(Y^r, Y^L, X) Q &= [(Y_0^r)^T \;\; 0]^T + [E^r(Q) \;\; E^L(X, Q)]^T \\
&= \bar{Y}^r + \bar{E}^r
\end{aligned} \quad (25)$$

where $Y_0^r = (I + T_s S(y_1^r))[q_0^1 \; q_0^2]^T$ since we, exactly as for the position, assume that the initial rotation is known. Define, similar to above,

$$m_t^{1,j_t} = y_t^{j_t} \delta_t^{j_t,x} - \delta_t^{j_t,y} \tag{26a}$$
$$m_t^{2,j_t} = y_t^{j_t} \delta_t^{j_t,y} + \delta_t^{j_t,x} \tag{26b}$$

The matrix B will then have the principal structure

$$\left[\begin{array}{cccccccc}
1 & & & & & & & \\
 & 1 & & & & & & \\
-1 & T_s y_2^r & 1 & & & & & \\
-T_s y_2^r & -1 & & 1 & & & & \\
 & & & & \ddots & & & \\
 & & & & -1 & T_s y_N^r & 1 & \\
 & & & & -T_s y_N^r & -1 & & 1 \\
\hline
m_1^{1,1} & m_1^{2,1} & & & & & & \\
\vdots & \vdots & & & \ddots & & \vdots & \vdots \\
 & & & & & & m_N^{1,M} & m_N^{2,M}
\end{array}\right]. \tag{27}$$

This system can now be solved according to WLS method as

$$\hat{Q} = (B^T \bar{\Sigma}_r^{-1} B)^{-1} B^T \bar{\Sigma}_r^{-1} \bar{Y}^r \tag{28}$$

where $\bar{\Sigma}_r = \text{diag}(\Sigma^r(Q),\ \Sigma^L(X,Q))$.

IV. The Iterative Method

The main idea now is to use the conditional linearity of the two interconnected systems,

$$A(Y^L, Q)X = \bar{Y}^v + \bar{E}^v \tag{29a}$$
$$B(Y^r, Y^L, X)Q = \bar{Y}^r + \bar{E}^r \tag{29b}$$

and solve for all the parameters, X and Q, by iterating solutions between the systems. This can be done using Block Coordinate Descent (BCD). However, the parameter dependent noise must be accounted for. We suggest IRBCD as combination BCD [19], [20], and Iteratively Reweighted Least Squares method, see e.g., [16], [21], where the weights are represented by the (square root of the) covariance matrix. In this way, the values from previous iteration are used as weights and both systems are solved until convergence. This method is shown in Algorithm 1. Although both main steps are solved as linear equation systems, the algorithm still needs initial values for the parameters. This is not so strange, since the underlying problem is still a non-linear and as such require a starting point for the iterations. Furthermore, Algorithm 1 is implemented in a way that can be altered, e.g.,, each linear solution, i.e., steps 4. and 6. are done with the previous iterate in the weights. An alternative is to iterate step 4. (and 6.) with the update of the weights until convergence and then move on. In other words, there are inner iterations around steps 4. and 6. This method is shown in Algorithm 2. Which method is preferred is not obvious at first glance, and numerical comparison will be done in Section V. Step 7. (and 17. in Algorithm 2) is needed in order to keep quaternion length constraint, since it is not guaranteed after solving for it. Also, if no (or constant) noise assumption is done, the only modification to the Algorithm 1 is to set noise covariances $\bar{\Sigma}$ to I or constant values

Algorithm 1 Iteratively Reweighted Block Coordinate Descent (IRBCD).

Require: X^0, Q^0, Y^L, Y^r, Y^v, Σ^v, Σ^r, Σ^L, ε
Ensure: $\hat{X}$, $\hat{Q}$

1. Set $i = 1$
2. Set *Terminate* = **false**

while not *Terminate*
  3. Create $A(Y^L, Q^{i-1})$, $\bar{Y}^v$ and $\bar{\Sigma}_v(X^{i-1}, Q^{i-1})$
  4. Solve $X^i = (A^T \bar{\Sigma}_v^{-1} A)^{-1} A^T \bar{\Sigma}_v^{-1} \bar{Y}^v$
  5. Create $B(Y^L, Y^r, X^i)$, $\bar{Y}^r$ and $\bar{\Sigma}_r(X^i, Q^{i-1})$
  6. Solve $Q^i = (B^T \bar{\Sigma}_r^{-1} B)^{-1} B^T \bar{\Sigma}_r^{-1} \bar{Y}^r$
  7. Normalise Q^i
  if $\|Q^i - Q^{i-1}\| < \varepsilon$ **and** $\|X^i - X^{i-1}\| < \varepsilon$ **then**
    8. Set *Terminate* = **true**
  else
    9. Set $i = i + 1$
  end if
end while
10. Set $\hat{X}$ = X^i and $\hat{Q} = Q^i$

independent of the parameters. A comparison of results with these assumptions will also be performed in the same section.

V. Numerical Results

Numerical evaluation of the methods above is performed on a very simple simulated setup with three landmarks and 10 trajectory positions. The landmarks are positioned in $[1 \; 100]^T$, $[5 \; 65]^T$ and $[3 \; 30]^T$ and the platform is mainly moving along the X-axis pointing along the Y-axis. The platform's movement is not completely uniform, since some variation in the velocity and rotation is applied. The measurements are the linear and angular velocities, and the landmark observations as in (11). The noise covariances are set to $\Sigma^L = 10^{-6}$ [m^2], $\Sigma^v = 10^{-2} I$ [m^2/s^2] and $\Sigma^r = 10^{-4}$ [deg^2/s^2]. 100 MC runs were performed and comparison of the performance for different noise assumptions is evaluated. The initial values of the parameters were $[0 \; 50]^T$ for all three landmarks and all zeros for the linear velocity. The rotations are initialised with random values around the true ones and the termination threshold ε is set to 10^{-4}.

A comparison of the performance between the IRBCD according to Algorithm 1 and 2, and with BCD for constant noise and noise-free assumptions. As a reference, BCD with "true" noise is also run, where true here means that the true parameter values are used in the covariance matrices. The performance is mainly evaluated for the landmarks due to simplicity of illustration. The results are given in Table I as the error between the true landmark position and the MC based mean, $\tilde{l}_i$, as well as its the covariance, Cov($\tilde{l}_i$). It can be seen that the performance of the IRBCD methods are basically identical and quite close to the true position which can be compared to the BCD estimate with the true noise assumption. However, if we compare to the estimates from BCD with the noise-fre assumption, we can see that it is biased and the

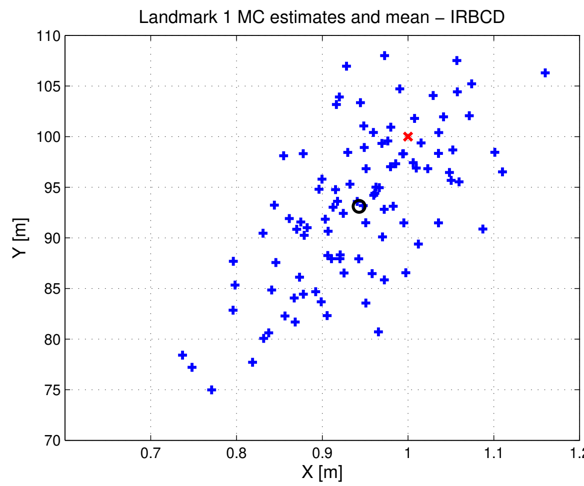
Algorithm 2 Iteratively Reweighted Block Coordinate Descent with inner iterations.

Require: X^0, Q^0, Y^L, Y^r, Y^v, Σ^v, Σ^r, Σ^L, ε
Ensure: $\hat{X}$, $\hat{Q}$

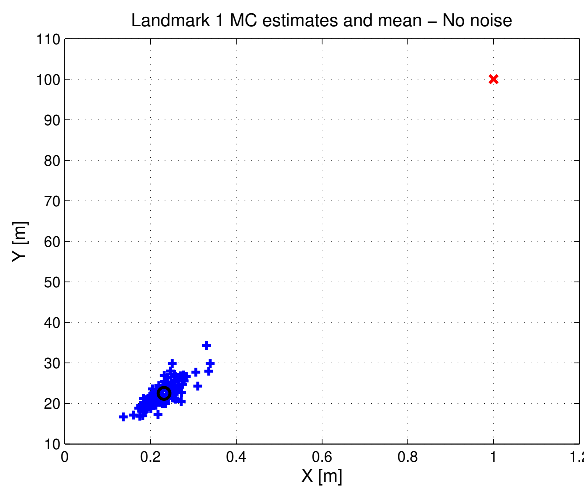
1. Set $i = 1$
2. Set *Terminate* = **false**
while not *Terminate*
  3. Set $j = 1$
  4. Set *Terminate_inner* = **false**
  5. Set $X^{j-1} = X^{i-1}$
  6. Create $A(Y^L, Q^{i-1})$ and $\bar{Y}^v$
  while not *Terminate_inner*
    7. Create $\bar{\Sigma}_v(X^j, Q^{i-1})$
    8. Solve $X^j = (A^T \bar{\Sigma}_v^{-1} A)^{-1} A^T \bar{\Sigma}_v^{-1} \bar{Y}^v$
    if $\|X^j - X^{j-1}\| < \varepsilon$ **then**
      9. Set *Terminate_inner* = **true**
      10. Set $X^i = X^j$
    else
      Set $j = j + 1$
    end if
  end while
  11. Set $j = 1$
  12. Set *Terminate_inner* = **false**
  13. Set $Q^{j-1} = Q^{i-1}$
  14. Create $B(Y^L, Y^r, X^i)$ and $\bar{Y}^r$
  while not *Terminate_inner*
    15. Create $\bar{\Sigma}_r(X^i, Q^{j-1})$
    16. Solve $Q^j = (B^T \bar{\Sigma}_r^{-1} B)^{-1} B^T \bar{\Sigma}_r^{-1} \bar{Y}^r$
    17. Normalise Q^j
    if $\|Q^j - Q^{j-1}\| < \varepsilon$ **then**
      18. Set *Terminate_inner* = **true**
      19. Set $Q^i = Q^j$
    else
      20. Set $j = j + 1$
    end if
  end while
  if $\|Q^i - Q^{i-1}\| < \varepsilon$ **and** $\|X^i - X^{i-1}\| < \varepsilon$ **then**
    21. Set *Terminate* = **true**
  else
    22. Set $i = i + 1$
  end if
end while
23. Set $\hat{X} = X^i$ and $\hat{Q} = Q^i$

Table I: Results for the landmark estimates

	True noise	IRBCD	IRBCD (inner)	No noise	Const. noise
$\tilde{l}_1$	$[.05\ 6.0]^T$	$[.06\ 6.9]^T$	$[.06\ 6.9]^T$	$[.8\ 77]^T$	$[1.0\ 99]^T$
$\mathrm{Cov}(\tilde{l}_1)$	$\begin{bmatrix} .006 & .3 \\ .3 & 43 \end{bmatrix}$	$\begin{bmatrix} .007 & .4 \\ .4 & 57 \end{bmatrix}$	$\begin{bmatrix} .007 & .4 \\ .4 & 57 \end{bmatrix}$	$\begin{bmatrix} .001 & .5 \\ .5 & 9.7 \end{bmatrix}$	$\begin{bmatrix} .0001 & .007 \\ .007 & .7 \end{bmatrix}$
$\tilde{l}_2$	$[.3\ 3.8]^T$	$[.3\ 4.4]^T$	$[.3\ 4.4]^T$	$[3.5\ 45]^T$	$[4.9\ 64]^T$
$\mathrm{Cov}(\tilde{l}_2)$	$\begin{bmatrix} .09 & 1.1 \\ 1.1 & 16 \end{bmatrix}$	$\begin{bmatrix} .1 & 1.6 \\ 1.6 & 21 \end{bmatrix}$	$\begin{bmatrix} .1 & 1.6 \\ 1.6 & 21 \end{bmatrix}$	$\begin{bmatrix} .04 & .5 \\ .5 & 6.2 \end{bmatrix}$	$\begin{bmatrix} .002 & .02 \\ .02 & .3 \end{bmatrix}$
$\tilde{l}_3$	$[.2\ 1.7]^T$	$[.2\ 2.0]^T$	$[.2\ 2.0]^T$	$[1.5\ 15]^T$	$[3.0\ 30]^T$
$\mathrm{Cov}(\tilde{l}_3)$	$\begin{bmatrix} .03 & .3 \\ .3 & 3.5 \end{bmatrix}$	$\begin{bmatrix} .04 & .4 \\ .4 & 4.6 \end{bmatrix}$	$\begin{bmatrix} .04 & .4 \\ .4 & 4.6 \end{bmatrix}$	$\begin{bmatrix} .02 & .2 \\ .2 & 2.2 \end{bmatrix}$	$\begin{bmatrix} .0006 & .006 \\ .006 & .06 \end{bmatrix}$



(a) Estimates from the IRBCD.



(b) Estimates from the BCD with noise-free assumption.

Figure 2: MC sstimates of the landmark 1 from IRBCD and noise-free assumption in blue. Black circle is the mean of the estimates and red cross is the true landmark position.

covariance is not covering the true position of the landmark. The same is applicable to the constant noise assumption, where the results are even worse. Further, the number of iterations that are performed on average for both IRBCD methods are compared. Both methods converges after 45 outer iterations on average while IRBCD with inner iterations has slower convergence since it requires more iterations in total. In this cse the average nuber of inner iterations was 30. Since the performance is basically the same, there is no need to use Algorithm 2 since it needs more iterations in total. As a further illustration Figure 2 shows MC estimates for the IRBCD and noise-free assumption for the landmark 1.

The rotation estimate is evaluated by comparing the true rotation matrix R_t^* and the MC estimated ones $\hat{R}_t^i$, $i =$

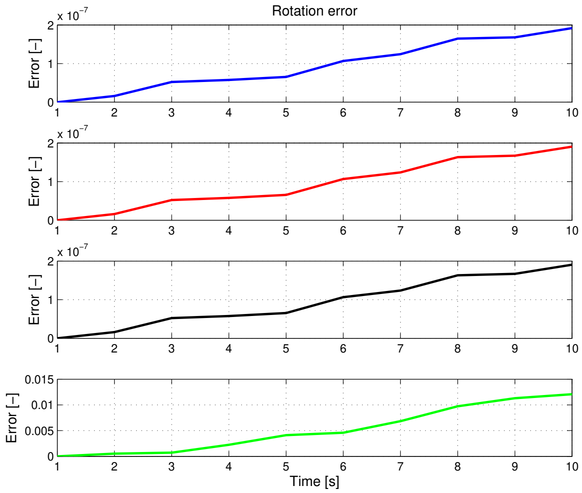


Figure 3: Error of the rotation estimate, BCD with true noise assumption in blue, IRBCD with inner iterations in red, IRBCD in black and BCD with noise-free assumption in green.

$\{1,\dots,N_{MC}\}$ as in

$$R_t^{\text{err}} = \sqrt{\frac{1}{2N_{MC}} \sum_{i=1}^{N_{MC}} \text{trace}\Big(\big((R_t^*)^T \hat{R}_t^i - I\big)^2\Big)} \tag{30}$$

Basically, this is a scalar measure of how aligned these matrices are. The result is shown in Figure 3. Here it can be seen that all but green estimate are similar and very close to zero, while the estimate from noise-free assumption is much larger. This also shows that taking the noise into account is important.

VI. Treatment of the 3D case

So far we have considered a 2D setup where we could utilise the linear dependence between parameters in the rotation parametrisation. Unfortunately, this property does not translate to the 3D case. The connection between the rotation matrix and the quaternion is non-linear in this case implying that the rotation part of the system must be solved with a non-linear method, e.g., non-linear least squares (NLS) (the linear velocity and landmarks part is still conditionally linear and can be solved as before). This basically means that the Algorithm 1 is modified in steps 5. and 6. by replacing the WLS with the N(W)LS step. The important part of NLS is the Jacobian of the equation system and it should be beneficial if the calculation of the Jacobian is efficient. We shall here formulate an approach to calculate the Jacobian in a recursive fashion, which increases the efficiency of the calculations.

To do this, we will change the parametrisation of the rotation and use angular velocity, $\omega_t = [\omega_t^x, \omega_t^y, \omega_t^z]^T,\ t = \{1,\dots,N\}$, as a parameter instead. This means that we need to express the rotation as a function of the angular velocity. Similar to the 2D case, the 3D rotation dynamics in the continuous time can be expressed as

$$\dot{R}_t = \widehat{\omega}_t R_t \tag{31a}$$

$$\widehat{\omega}_t = \begin{bmatrix} 0 & -\omega_t^z & \omega_t^y \\ \omega_t^z & 0 & -\omega_t^x \\ -\omega_t^y & \omega_t^x & 0 \end{bmatrix}. \tag{31b}$$

see e.g., [12]. Note also that this change of parameters implies that the step 7. from Algorithm 1 can be removed since angular velocities do not need normalisation. In the same way as before this dynamics can be discretised, under the constant angular velocity between the sampling times assumption, as

$$R_t = e^{T_s \widehat{\omega}_t} R_{t-1}. \tag{32}$$

The dynamics can now be used to express a rotation matrix at any time t as a function of the initial rotation R_0 and the angular velocities up to time t, $\omega_t,\ t = \{1,\dots,t\}$ as

$$R_t = \Big(\prod_{k=1}^{t} e^{T_s \widehat{\omega}_k}\Big) R_0. \tag{33}$$

For the Jacobian calculation, terms for the partial derivatives of the rotation matrices with respect to angular velocities, $\partial R_i / \partial \omega_j,\ j \le i$ are needed. These are defined as a $3 \times 3 \times 3$ matrices and can be calculated by observing that

$$\begin{aligned} \frac{\partial R_i}{\partial \omega_j} &= e^{T_s \widehat{\omega}_i} e^{T_s \widehat{\omega}_{i-1}} \cdots \frac{\partial e^{T_s \widehat{\omega}_j}}{\partial \omega_j} \cdots e^{T_s \widehat{\omega}_1} R_0 = \\ &= e^{T_s \widehat{\omega}_i} \frac{\partial R_{i-1}}{\partial \omega_j}, j < i \end{aligned} \tag{34}$$

where all partial derivatives are calculated in the previous recursive step and the last term is calculated as

$$\frac{\partial R_i}{\partial \omega_i} = \frac{\partial e^{T_s \widehat{\omega}_i}}{\partial \omega_i} R_{i-1}. \tag{35}$$

For example, the first three time steps these calculations are as follows:

$$R_1 = e^{T_s \widehat{\omega}_1} R_0 \tag{36a}$$

$$\frac{\partial R_1}{\partial \omega_1} = \frac{\partial e^{T_s \widehat{\omega}_1}}{\partial \omega_1} R_0 \tag{36b}$$

$$R_2 = e^{T_s \widehat{\omega}_2} R_1 \tag{36c}$$

$$\frac{\partial R_2}{\partial \omega_1} = e^{T_s \widehat{\omega}_2} \frac{\partial R_1}{\partial \omega_1} \tag{36d}$$

$$\frac{\partial R_2}{\partial \omega_2} = \frac{\partial e^{T_s \widehat{\omega}_2}}{\partial \omega_2} R_1 \tag{36e}$$

$$R_3 = e^{T_s \widehat{\omega}_3} R_2 \tag{36f}$$

$$\frac{\partial R_3}{\partial \omega_1} = e^{T_s \widehat{\omega}_3} \frac{\partial R_2}{\partial \omega_1} \tag{36g}$$

$$\frac{\partial R_3}{\partial \omega_2} = e^{T_s \widehat{\omega}_3} \frac{\partial R_2}{\partial \omega_2} \tag{36h}$$

$$\frac{\partial R_3}{\partial \omega_3} = \frac{\partial e^{T_s \widehat{\omega}_3}}{\partial \omega_3} R_2. \tag{36i}$$

In this example, it can be seen that in order to calculate the rotation matrix and its derivative at time step t, $e^{T_s \widehat{\omega}_t}$ and

$\partial e^{T_s \widehat{\omega}_3}/\partial \omega_3$ must be calculated once and all other terms are available from the previous time step calculations.

Since the new parametrisation is used, we need to use measurements of the angular velocity y_t^r from (11) directly in the equation system instead of (23). It should also be mentioned that there is an closed form expressions for both the exponential $e^{T_s \widehat{\omega}_t}$ and the for the partial derivative $\partial e^{T_s \widehat{\omega}_t}/\partial \omega_t$ which can be calculated by using Rodrigue's formula, [12],

$$e^{T_s \widehat{\omega}} = I + \frac{\widehat{\omega}}{\|\omega\|} \sin(T_s \|\omega\|) + \frac{\widehat{\omega}^2}{\|\omega\|^2} (1 - \cos(T_s \|\omega\|)) \tag{37}$$

but for brevity the derivative expression is not explicitly given here.

VII. Conclusions and Future Prospects

In this work we have analysed the conditionally linear structure of the visual-inertial estimation problem in 2D. A fomulation was proposed by splitting the problem into into two (interconnected) conditionally linear systems. The first system assumes known rotations while the linear motion and landmarks are unknown. The second system has the opposite assumption. The main benefit of doing this split is the efficiency of the solution. Furthermore, the measurement noise becomes parameter dependent in both systems which is treated with WLS that result in bias reduction and improved consistency.

We also give a suggestion on how the total system can be solved in an iterative manner, by combining the Block Coordinate Descent and the Iteratively Reweighted Least Squares approaches. Comparison of the estimation results from these methods under different noise assumptions shows that parameter dependence in the noise must be used if good estimates are sought.

So far, we have treated only the 2D case because of the (conditional) linearity for both linear and rotational parameters. For the 3D case, this is not as simple and the rotation estimation step must be exchanged for the non-linear method. No experiments are done in this case, but we show how the Jacobian can be efficiently calculated in a recursive manner.

In the future, experiments with both simulated and real 3D data should be performed and the performance of both parameter accuracy as well as execution time analysed in more detail.

References

[1] A. Martinelli, "Vision and IMU Data Fusion: Closed-Form Solutions for Attitude, Speed, Absolute Scale, and Bias Determination," *Robotics, IEEE Transactions on*, vol. 28, no. 1, pp. 44–60, Feb. 2012.

[2] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart, "Fusion of IMU and vision for absolute scale estimation in monocular SLAM," *Journal of Intelligent Robotics Systems*, vol. 61, pp. 287–299, Jan. 2011. [Online]. Available: http://dx.doi.org/10.1007/s10846-010-9490-z

[3] L. Kneip, A. Martinelli, S. Weiss, D. Scaramuzza, and R. Siegwart, "Closed-Form Solution for Absolute Scale Velocity Determination Combining Inertial Measurements and a Single Feature Correspondence," in *International Conference on Robotics and Automation*, Shanghai, China, May 2011. [Online]. Available: http://hal.inria.fr/hal-00641772

[4] T. Lupton and S. Sukkarieh, "Visual-Inertial-Aided Navigation for High-Dynamic Motion in Built Environments Without Initial Conditions," *Robotics, IEEE Transactions on*, vol. 28, no. 1, pp. 61–76, Feb. 2012.

[5] H. Strasdat, J. M. M. Montiel, and A. Davison, "Real-time monocular slam: Why filter?" in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 2657–2664.

[6] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, November 2007.

[7] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2011.

[8] A. Mourikis and S. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Roma, Italy, 10-14 Apr. 2007, pp. 3565–3572.

[9] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2017.

[10] K. Eckenhoff, P. Geneva, and G. Huang, "Closed-form preintegration methods for graph-based visual–inertial navigation," *The International Journal of Robotics Research*, vol. 38, no. 5, pp. 563–586, 2019.

[11] Y. Liu, Z. Chen, W. Zheng, H. Wang, and J. Liu, "Monocular visual-inertial slam: Continuous preintegration and reliable initialization," *Sensors*, vol. 17, no. 11, p. 2613, 2017.

[12] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer Verlag, 2003.

[13] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[14] R. I. Hartley and P. Sturm, "Triangulation," *Computer Vision and Image Understanding*, vol. 68, no. 2, pp. 146 – 157, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314297905476

[15] Z. Zhang and T. Kanade, "Determining the Epipolar Geometry and its Uncertainty: A Review," *International Journal of Computer Vision*, vol. 27, pp. 161–195, 1998.

[16] Å. Björck, *Numerical Methods for Least Squares Problems*. SIAM, 1996.

[17] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly Journal of Applied Mathmatics*, vol. II, no. 2, pp. 164–168, 1944.

[18] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM Journal on Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[19] C. A. Hildreth, "A Quadratic Programming Procedure," *Naval Research Logistics Quarterly*, vol. 4, pp. 79–85, 1957, see also Erratum, Naval Research Logistics Quarterly, Vol. 4, p. 361, 1957.

[20] J. Warga, "Minimizing Certain Convex Functions," *SIAM Journal on Applied Mathematics*, vol. 11, pp. 588–593, 1963.

[21] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.