2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Comparison of online algorithms for the tracking of multiple magnetic targets in a myokinetic control interface*

J. Montero, M. Gherardini, F. Clemente, C. Cipriani, *Senior Member, IEEE*

Abstract— **Magnetic tracking algorithms can be used to determine the position and orientation of magnetic markers or devices. These techniques are particularly interesting for biomedical applications such as teleoperated surgical robots or the control of upper limb prostheses. The performance of different algorithms used for magnetic tracking was compared in the past. However, in most cases, those algorithms were required to track a single magnet.**

Here we investigated the performance of three localization algorithms in tracking up to 9 magnets: two optimization-based (Levenberg-Marquardt algorithm, LMA, and Trust Region Reflective algorithm, TRRA) and one recursion-based (Unscented Kalman Filter, UKF). The tracking accuracy of the algorithms and their computation time were investigated through simulations.

The accuracy of the three algorithms, when tracking up to six magnets, was similar, leading to estimation errors varying from 0.06 ± 0.02 mm to 2.26 ± 0.07 mm within a 100 mm × 54 mm × 100 mm workspace, at the highest sampling frequency. In all cases, computation times under 300 ms for the UKF and 45 ms for the LMA/TRRA were obtained. The TRRA showed the best tracking performance overall.

These outcomes are of interest for a wide range of robotics applications that require remote tracking.

I. Introduction

Magnetic tracking deals with the determination of the position and/or orientation of a specially designed magnetic marker or device by means of its interaction with static or low-frequency magnetic fields (MF) [1]. As the human body is transparent to low-frequency MFs, this technique is particularly interesting for biomedical applications. These include tracking of body [2] and bone movements [3], of surgical needles [4] and more complex instruments like bronchoscopes [5], colonoscopes and robotic capsules for endoscopy [6].

Monitoring the position of a magnetic marker is also useful if the marker (or what it is attached to) must be teleoperated. Indeed, although open-loop control strategies exist [7], magnetic tracking allows to implement more efficient closed-loop control strategies to wirelessly control fully untethered micro-robots [8] or to perform robotic surgeries [9].

In the framework of biomedical applications, we recently proposed to track the movement of multiple magnetic markers (MM), in the form of permanent magnets, to control a dexterous hand prosthesis [10]. Indeed, MMs could be implanted in the residual muscles of an amputee to provide information on their contraction state. By concurrently monitoring the position of several magnets, these signals could potentially be used to simultaneously and proportionally control multiple DoFs of a hand prosthesis. We called this the *myokinetic* control interface [10].

In systems like the one described above, the pose (i.e. position and orientation) of passive MMs is usually retrieved by measuring the MF they generate with remote sensors [11],[12]. The problem of reconstructing the pose of a MM (or, more generally, a magnetic source) from such measurements is called the inverse problem of magnetostatics. From a theoretical standpoint, the solution of this problem entails the inversion of a model that describes the MF generated by the sources in the workspace. Several investigators tried to optimize the performance of these systems by developing (i) models that more accurately describe the MF [13],[14] or (ii) algorithms able to efficiently solve the problem (by inverting or numerically solving the particular model used) [15]-[18]. The latter are the primary focus of this work

Algorithms following different approaches were proposed so far. These span from analytical solutions [15] to neural networks [16], from optimization-based [17] to recursion-based methods [18],[19]. The reason for these efforts is that an analytical solution of the inverse problem is only available for the particular case where a single magnet is tracked [15], and thus numerical approximations are needed. In the process of finding the best candidate for magnetic tracking, the performance of different algorithms was compared in terms of computation time, localization accuracy and sensitivity to the quality of the initial conditions. Specifically, Hu and colleagues [20] compared the performance of five non-linear optimization algorithms (i.e. Powell's, Downhill Simplex, DIRECT, Multilevel Coordinate Search, and Levenberg-Marquardt). They found that the Levenberg-Marquardt algorithm (LMA) outperformed the other ones in terms of computation time, and (together with the Multilevel Coordinate Search algorithm) showed the best tracking accuracy. In fact, the LMA was exploited by several research groups to track one [3],[29] or more [10],[21],[22],[23] magnetic sources, using both desktop and embedded platforms [24]. On the other hand, the LMA was outperformed by a particle swarm optimization method in tracking the pose of a parallelepiped magnet [14]. Another example is the work from Birsan [25], that developed a magnetic tracking system based on an unscented Kalman Filter (UKF). He showed that the UKF outperformed the

*Research supported by the European Research Council under the MYKI project (ERC-2015-StG, Grant no. 679820).

J. Montero, M. Gherardini, F. Clemente and C. Cipriani (corresponding author; phone: +39050883133; fax: +39050883497; e-mail: christian.cipriani@santannapisa.it) are with The BioRobotics Institute, Scuola Superiore Sant'Anna, Pisa, Italy, and with the Department of Excellence in Robotics & AI, Scuola Superiore Sant'Anna, Pisa, Italy.

direct inversion of the magnetic dipole model in terms of tracking accuracy.

In most of these examples, the algorithms were required to track a single MM. However, we envision magnetic tracking systems able to track many more magnets, as these would enable a whole new class of applications, from the control of multi-articulated prosthetic hands to robot swarms. Focusing on the first case, we investigated and compared the performance of three viable localization algorithms in tracking up to 9 MMs in a workspace resembling the human forearm. We tested the Trust-Region Reflective algorithm (TRRA) [26], the LMA and the UKF [25].

The TRRA is an iterative optimization method for non-linear least square problems, based on the Gauss-Newton algorithm, which searches for a new solution in a "trusted region" around the current one. The TRRA implementation used in this work allowed to implement solution boundaries. The LMA, commonly used in the literature, is similar to the TRRA as it finds a solution using the Gauss-Newton algorithm. LMA differs from TRRA because it is interpolated with the method of gradient descent. Finally, the UKF falls under the class of recursive algorithms. These, in addition to the most recent observed value (i.e. MF sample), also exploit information from previous measurements to improve the prediction of the desired variable (i.e. the pose of the MMs).

Our results showed that the TRRA pose estimation error is the lowest, with the LMA following closely. The error in tracking up to 6 MMs was under half a millimeter, while 9 MMs could not be tracked with reasonable accuracy by any algorithm. The UKF showed the worst performance, with localization errors generally below 2 mm. The latter was also the most sensitive to the quality of the initial conditions. Finally, the computation time of the UKF was the largest (around 300 ms), and at least one order of magnitude larger than the LMA. These outcomes are of interest for a wide range of robotics applications in which remote tracking is necessary, and support the viability of multiple MM tracking systems.

II. Materials and Methods

A. Mathematical approximation and localization algorithms

In order to simplify the solution of the localization problem (inverse problem of magnetostatics), the point dipole model approximation can be exploited. This model approximates each MM as a point magnetic dipole located at its center. The magnetic field $\boldsymbol{B}_i = \boldsymbol{B}(\boldsymbol{x}_i)$, generated at the location $\boldsymbol{x}_i$ by a collection of n dipoles, located at $\boldsymbol{x}_j$, $j = 1, \ldots,$ n, with magnetic moment respectively equal to $M_j\boldsymbol{m}_j$ (here M_j and $\boldsymbol{m}_j$ are the magnitude and direction of the magnetic moment of the j-th MM), can be evaluated as:

$$\boldsymbol{B}(x_i) = \sum_{j=1}^{n} \frac{M_j \mu_j \mu_0}{4\pi} \left(\frac{3(\boldsymbol{m}_j \boldsymbol{x}_{ij})\boldsymbol{x}_{ij}}{|\boldsymbol{x}_{ij}|^5} - \frac{\boldsymbol{m}_j}{|\boldsymbol{x}_{ij}|^3} \right), i = 1, \ldots, N \quad (1)$$

Here xij = xi - xj, and xi represents the location of N sites (sensors) where the magnetic field is measured. This approximation is excellent in the ideal case of infinite distance between sensors and sources (far field) while it loses accuracy when this distance becomes smaller [27]. Nevertheless, it proved sufficiently accurate in several non-ideal cases [28],[29]. Thus, measuring the compound MF generated by the n MMs, using multiple sensors, allows to solve (1) in favor of $\boldsymbol{x}_{ij}$, providing the solution to the localization problem.

In this work, we performed several simulations (see below) and the resulting measurements of the MF were fed to the three numerical approximation methods under evaluation (LMA, TRRA and UKF). All these algorithms need as an input, in addition to the measurements of the MF, an estimate of the pose of all the MMs, to use as an initial guess where to start to find a solution. This was set depending on the test being carried out (described below).

Two of the numerical approximation methods tested (LMA and UKF) were implemented in C++, and executed on a desktop PC (HP ProDesk 490 G3 MT Business PC, Intel Core i7-6700 CPU at 3.40 GHz, 16 GB of RAM) running a real-time Linux OS (Ubuntu 18.10 with the PREEMPT_RT patch, Canonical). Specifically, the LMA implementation from the freely available CMinpack library was used [31]. On the other hand, the UKF was implemented from scratch (Appendix), based on the work from Birsan [25]. The MATLAB implementation of the TRRA was used.

B. Simulation setup

Five systems were simulated using COMSOL Multiphysics® (COMSOL Inc., Stockholm, Sweden), in order to compare the performance of the three algorithms. Each system comprised a different number of MMs (either 1, 2, 4, 6 or 9) in a fixed workspace mimicking the dimensions of the human forearm [30]. The workspace was modeled as a parallelepiped of 100 mm × 54 mm × 100 mm ($l \times w \times h$), with the bottom base centered at the origin of a Cartesian coordinate system, and parallel to the XY plane at $z = 0$ (Fig. 1, upper inset). The MMs were modeled as Nd-Fe-B grade disc magnets ($r = h = 2$ mm, axial remanent magnetization $B_r = 1.27$ T) with their magnetization vector pointing along the positive Z direction.

For each system, the MMs were initially arranged in a specific configuration depending on the total number of MMs (Fig. 1, middle inset). Then, we simulated the movement of a single MM (namely MM2, except in the single-magnet case in which MM1 was moved) along a pre-defined linear trajectory (along Y). Specifically, MM2 was moved back and forth by 10 mm, with 1 mm steps, for two cycles. This resulted in a total of 41 data points (Fig. 1, lower inset).

The magnetic field was sampled at specific sites, simulating the presence of magnetic field sensors. The sites were laid out on four planar and orthogonal grids of seven columns and four rows, for a total of 112 sites. Each column and row were separated by 9 mm. The four grids were spatially centered at the four sides of the workspace, two on the XZ plane, and two on the XY plane (Fig. 1, upper inset). In addition, noise was added to all the simulated data (50 repetitions). The noise was generated from white gaussian noise (standard deviation of 2 mG) that was then quantized (resolution of 4.35 mG), in order to simulate conditions as similar as possible to the ones encountered in a real system. This simulated the noise characteristics of a commercial magnetometer (HMC5983, Honeywell Inc., Morristown, NJ).

Two kinds of tests were performed: static and dynamic.

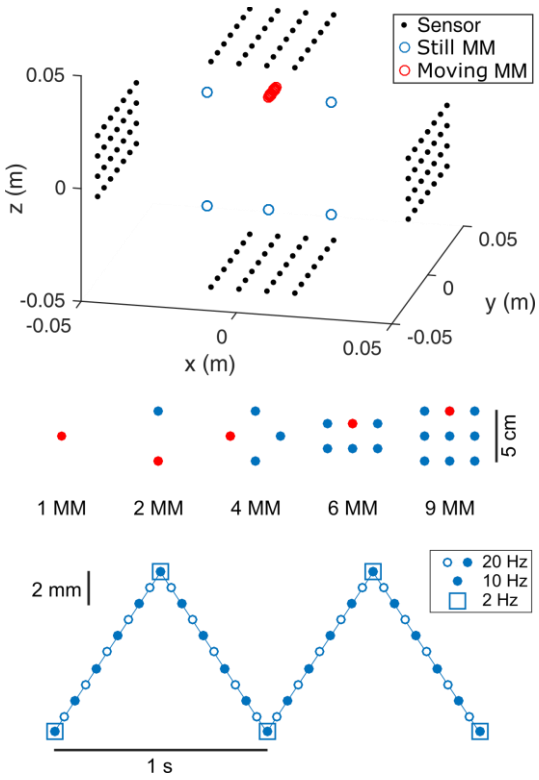

Figure 1. Experimental setup: simulated distribution of sensors and magnets (MM). Upper inset: distribution of the four groups of sensors that surround the workspace and the placement of the MMs (six MMs system). Middle inset: distribution of MMs in the five systems. The red dot indicates the moving MM, i.e., MM2. Lower inset: trajectory followed by the moving MM; different sampling frequencies were simulated, which resulted in a different amount of data points.

Static tests

In the static case, the different spatial configurations generated at each step of MM2 (MM1 in the single-magnet case) were considered as separate static localization problems to solve. In this case, the initial guess required by all algorithms was set as the actual pose of the MMs. The algorithms were required to retrieve the position of the MMs over the 50 repetitions generated by adding noise to the simulated data.

Dynamic tests

In the dynamic case, the different positions occupied by MM2 (MM1 in the single-magnet case) were considered as part of a trajectory having a 1 s period (all other MMs were kept still in the initial position). In this case, the pose retrieved by the algorithms in the previous iteration was used as the initial guess for the current iteration, so to mimic an online system. Additionally, different sampling frequencies were simulated by using subsets of the available data points. These were 20 Hz (all data points), 10 Hz and 2 Hz (five data points – Fig. 1, lower inset). This aimed to test the sensitivity of the algorithms to the quality of the initial guess. Indeed, by decreasing the sampling frequency, the distance between subsequent data points grows, as well as the distance between the initial guess and the actual position of the MMs.

Algorithms configuration

The additional information about the movement of the MMs, required by the TRRA and UKF algorithms, was set as follows. The sigma value of dimension Y (along which MM2 moved) was set to 0.1, while the other ones were set to 0.001 (APPENDIX). Correspondingly, for the TRRA boundary constraints were set. In particular, each MM was allowed to move, in the *XZ* plane, within a square with a side of 2 cm. Along *Y*, instead, the MMs were allowed to move 2 cm away from their starting position. These displacement boundaries were set symmetrically along all directions, with respect to the MM initial position, and they were equal for all MMs. Finally, the *Z* component of the MMs orientation was constrained to remain positive, ensuring a maximum angular excursion, from the initial position, of ± 90°.

C. Performance evaluation

The performance of the three algorithms was evaluated using three metrics: the computation time (CT), the localization error, and the robustness to the initial guess. In particular, operating system functions in the C++ source code were used to measure the CT needed by the algorithm to localize the MMs. Unfortunately, the CT of the TRRA algorithm could not be calculated, as it was implemented in a different platform (i.e. MATLAB). However, since Trust Region algorithms and the LMA are both based on the Newton Step method [32], they should exhibit similar convergence speeds, hence justifying the quantification of the CT for only one of the two optimization methods in this preliminary assessment.

The localization error was evaluated both in terms of position and orientation accuracy. In the former case, at each iteration, the error was defined as the Euclidean distance between the real ($\boldsymbol{x}_{real}$) and the estimated ($\boldsymbol{x}_{track}$) position:

$$E_p = \|\vec{x}_{real} - \vec{x}_{track}\| \quad (2)$$

The orientation error was defined as the angle θ between the direction vectors of the real and estimated magnetic moments (considering $|\boldsymbol{m}| = 1$):

$$E_o = \theta = cos^{-1}(\vec{m}_{real} \cdot \vec{m}_{track}) \quad (3)$$

Akin to our previous work [23], these errors were split in two components (e_m for the moving MM and e_{ct} for the others), each with an average value ($\bar{e}$), plus-minus a standard deviation (S). In particular, for each system, the errors along the trajectory were computed (conservatively) as the 95th percentile of the aggregated data. Additionally, in the dynamic test, the algorithms' performance was analyzed independently for each of the two cycles. This was performed to allow the UKF to appropriately set its covariance matrix (stationary regime).

Finally, the robustness of the algorithms to the quality of the initial guess was quantified by correlating the errors (e_m and e_{ct}) with the distance between the initial guess and the actual position of the MMs, using the Spearman correlation coefficient, r_s.

III. RESULTS

For the sake of brevity we report only the results related to the position accuracy. In fact, the orientation accuracy

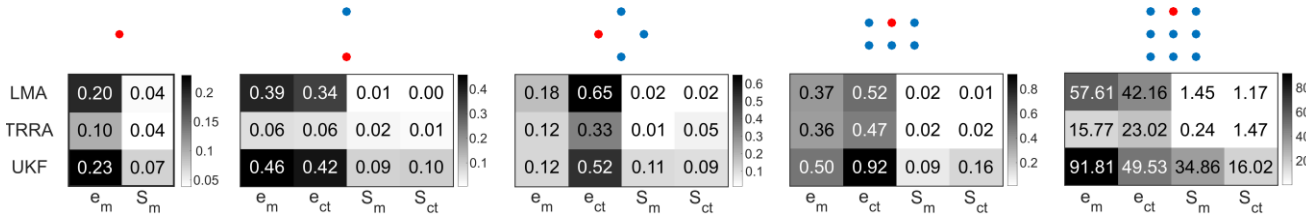

Figure 2. Localization errors (e_m and e_{ct}) and their standard deviations (S_m and S_{ct}) for the static test. All numbers are in mm.

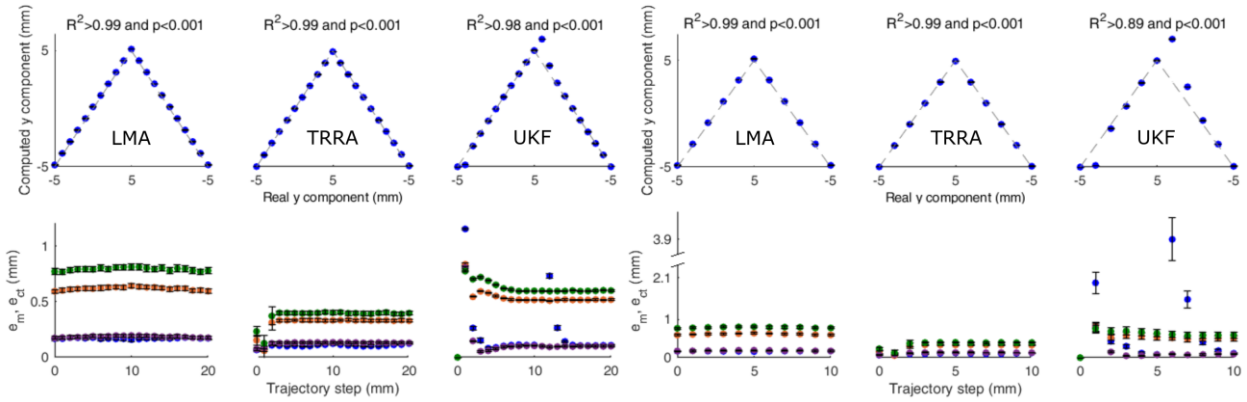

Figure 3. Performance of the three algorithms for the four-magnets system (first cycle). The upper row presents the actual and localized trajectory followed by MM2 along *Y*. The lower row shows the distribution of e_m and e_{ct} for all four magnetic markers.

proved always lower than 10° in all systems up to 6 MMs, and its trend closely matched that found for the position accuracy.

Overall, the static tests demonstrated comparable localization error across the three algorithms (Fig. 2). The TRRA exhibited the lowest e_m and e_{ct}, with standard deviations similar to those of the LMA. The UKF localization typically resulted in the largest errors, albeit the differences with respect to the other two algorithms were small (in the order of hundreds of microns). All algorithms showed considerably larger localization errors in the system with nine MMs. Specifically, the minimum e_m and e_{ct} were 0.06 mm, obtained when the TRRA was used to localize two magnets. The largest e_m (excluding the nine-magnets system) was 0.50 mm, while the largest e_{ct} was 0.92 mm. Both values were obtained when the UKF was used to localize six MMs. S_m and S_{ct} were always low, i.e. under 0.04 mm and 0.05 mm for the LMA and the TRRA, respectively. The UKF resulted in larger standard deviations (up to 0.16 mm).

For the dynamic tests, the results from a representative configuration (4 MMs, 20 Hz and 10 Hz cases) are first presented (Fig. 3). The performance of the LMA and TRRA was not affected by the change in sampling frequency, with e_m and e_{ct} being lower than 0.18 and 0.82 mm, respectively. On the other hand, the UKF performance reduced significantly, as e_m and e_{ct} increased by a factor of ~2 (i.e. from 2.01 mm to 3.92 mm). The TRRA exhibited the lowest e_m and e_{ct} at both frequencies, being these equal to 0.12 mm and 0.4 mm, respectively. For all algorithms, the relationship between the actual and the computed position along *Y* proved highly linear, with R^2 being always higher than 0.89 (p<0.001). More generally, the TRRA showed the best pose estimation accuracy also in the dynamic tests (Fig. 4). With this algorithm, excluding the system with 9 MMs, e_m was as low as 0.06 mm (two MMs system) and always lower than 0.36 mm. The TRRA and LMA e_{ct} was comparable, being always lower than 0.81 mm. The UKF pose estimation accuracy was the lowest one, leading to values of e_m and e_{ct} as large as 33.29 mm and 4.75 mm, respectively (both found for the six magnets systems at 2 Hz). In agreement with the static case, all algorithms showed considerably larger localization errors in the system with nine MMs.

The LMA and the TRRA proved robust to variations of the initial guess, being basically unaffected by a change in the sampling frequency (r_s < 0.01, p > 0.5 in both cases). On the other hand, the accuracy of the UKF in terms of e_m and e_{ct} was inversely correlated with the quality of the initial guess (r_s = 0.36, p = 0.01, Fig. 4). As an example, in the system with 4 MMs, e_m for the UKF increased from 2.01 mm to 18.54 moving from 20 Hz to 2 Hz.

As expected, no particular differences were observed between the localization errors of the LMA and TRRA during the first and the second cycle (Fig. 4). This held true also for the UKF, with the exception of the 2 Hz case, for which the performance decreased during the second cycle. Under these conditions, however, e_m was considerably large during both cycles, ranging from 14.27 ± 0.01 mm to 33.29 ± 0.05 mm (first and second cycle in the 6 MMs system).

The estimation accuracy displayed a non-monotonic trend with respect to the number of magnets. For instance, the errors for the four MMs system were among the lowest, while those for the six MMs system were among the largest. As an example, e_m for the LMA, when tracking four MMs, was as low as 0.18 mm, while it increased to 0.37 mm in the six MMs system.

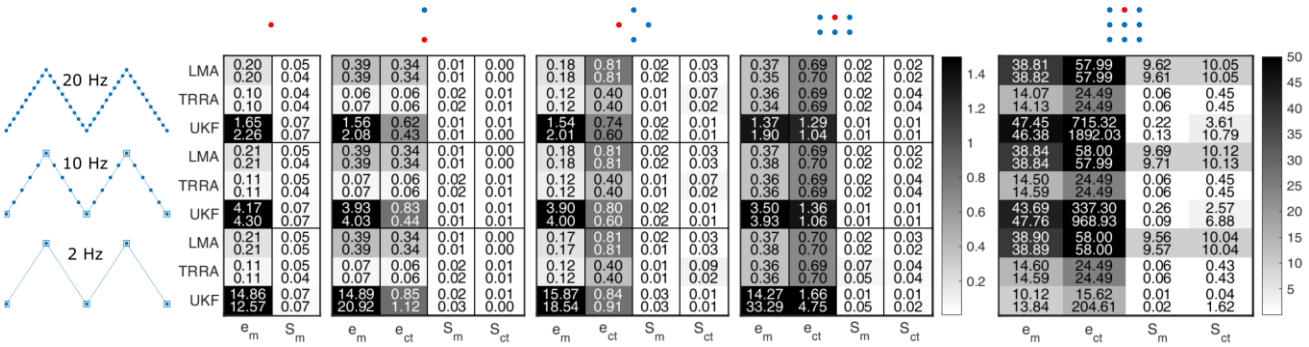

Figure 4. Localization errors (e_m and e_{ct}) and their standard deviations (S_m and S_{ct}) for the dynamic tests. All numbers are in mm. Each cell includes the results obtained during the first (above) and second (below) movement cycle.

The median CT of the LMA ranged from ~3 ms, for the single-magnet system, to ~45 ms for the six magnets system, and increased by a factor of four, i.e. up to ~216 ms, for the nine MMs system (Fig. 5). The median CT of the UKF was always larger than that of the LMA. Specifically, the LMA was a minimum of 6 (6 MMs system) to a maximum of 50 (1 MM system) times faster than the UKF.

IV. Discussion

In this work we have assessed, through simulations, the performance of a multi-magnet localizer, based on three different localization algorithms. Overall, our results show that the LMA, widely used in the literature, and the TRRA have similar performance. The latter actually showed slightly better performance in all the performed tests. As an advantage, the TRRA implementation we used also allowed to set solution boundaries. This helped in reducing the error drastically in the case of 9 MMs, where all algorithms failed to accurately track the MMs (Fig. 2 and Fig. 4). Thus, if information on the movement of the MMs is available a priori, the TRRA should be preferred. On the other hand, although the UKF algorithm takes advantage of the history of the system, it proved to be the worst solution in all respects, with the largest localization errors and CT.

We associated the consistent drop in performance, when nine magnets were used, to the particular distribution of the magnets (Fig. 1, middle inset). We argue that the magnet in the middle of the distribution was "shadowed" by the magnets surrounding it, making the algorithms unable to track it from the readings of the magnetic field. Indeed, the symmetry of the distribution makes the magnet in the middle the furthest from the sensors. Thus, the magnetic field it generates at any sensors locations is lower than the one generated by the other MMs, making the latter dominate the readings.

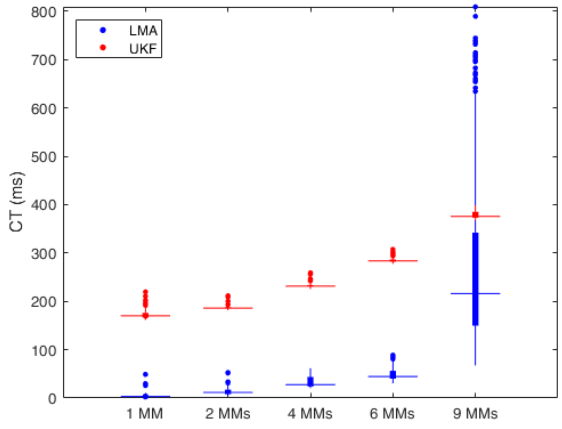

Figure 5. Distribution of the computation time (CT) for the LMA (blue) and the UKF (red) as a function of the number of magnets. The horizontal line indicates the median, the vertical thick line the interquartile-range, the vertical thin line extend to the most extreme data points, and the dots represent the outliers.

The UKF suffered from the reduction of the sampling frequency, which led to an increase in the pose estimation error. This follows from the recursive nature of the algorithm: lowering the sampling frequency implies that the filter is not able to accurately predict fast dynamics. This behavior is evident when MM2 changes direction of movement (Fig. 3). At this point, the error associated with MM2 increased by an order of magnitude. This happened because the algorithm assumes that the underlying process is a real dynamic system, which cannot display this kind of abrupt change. In fact, the derivative of the trajectory (Fig. 1, lower inset), i.e. its velocity, is discontinuous, and this negatively affects the estimation accuracy.

As previously reported [23], the CT of the LMA was largely affected by the number of tracked MMs (it increased by an order of magnitude from 1 to 6 MMs). This was also true for the UKF, but to a lower extent. Indeed, the CT only increased by a factor of 2 when moving from 1 to 6 MMs. Nonetheless, the UKF always showed CTs two orders of magnitude larger than the optimization-based method. This is inherent to the complexity of the algorithm, which requires the calculation of the inverse of a covariance matrix, which is a costly operation when its dimensionality is high. As the increased complexity is not justified by increased performance (the localization accuracy was lower than the other algorithms), the UKF might not be a good candidate in solving a multi-magnet localization problem.

Our results confirm, and extend, previous findings from the literature. Indeed, the LMA was found to be an excellent candidate for tracking multiple MMs as well. However, the TRRA showed even better performance, and could thus be a more sophisticated replacement for the LMA. Indeed, the former is a more complex algorithm, but freely available C implementations of trust region algorithms already exist [33]. These could speed up the adoption of this algorithm in both desktop and portable applications.

This work is a step forward in the development of multi-magnet tracking systems, as it demonstrates high computational efficiency and localization accuracy of the tested algorithms in solving such a complex problem. We foresee that these algorithms will enable a variety of applications based on multiple-magnets tracking systems, both in the biomedical field and beyond.

APPENDIX

A. Mathematical description of the UKF algorithm

The UKF requires a dynamical model describing the evolution of its states (the pose and speed of the MMs), as well as a model relating the states to the observations. The latter corresponds, in our work, to the magnetic dipole model, described by (1). As the underlying process is linear, the dynamics can be described by the time-invariant matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. In this case, the filter prediction is computed as:

$$\boldsymbol{\chi}^i_{k|k-1} = \boldsymbol{A}\boldsymbol{\chi}^i_{k-1} + \boldsymbol{B}\boldsymbol{u}_k, i = 0, \dots, 2L$$

$$\hat{\boldsymbol{x}}_{k|k-1} = \sum_{i=0}^{2L} W^i_S \boldsymbol{\chi}^i_{k|k-1}$$

$$\boldsymbol{P}_{k|k-1} = \left(\sum_{i=0}^{2L} W^i_C \left(\boldsymbol{\chi}^i_{k|k-1} - \hat{\boldsymbol{x}}_{k|k-1}\right)\left(\boldsymbol{\chi}^i_{k|k-1} - \hat{\boldsymbol{x}}_{k|k-1}\right)^T \right) + \boldsymbol{Q}$$

Where vector $\boldsymbol{\chi}$ corresponds to the $2L+1$ sigma points generated by the algorithm [19], and L corresponds to the number of states of the filter. The state of each MM can be described as $\boldsymbol{x_M} = [\boldsymbol{x}^T\ \boldsymbol{v}^T\ \boldsymbol{\theta}^T]^T$, where vectors $\boldsymbol{x}$, $\boldsymbol{v}$, and $\boldsymbol{\theta}$, represent, respectively, its 3D position, its 3D velocity and the two angles used to represent its orientation (spherical coordinates). Eight variables are thus required to describe the dynamics of each MM. The UKF state vector is then constructed, for N magnets, as:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x_{M,1}} \\ \vdots \\ \boldsymbol{x_{M,i}} \\ \vdots \\ \boldsymbol{x_{M,N}} \end{bmatrix}$$

Vector $\boldsymbol{x}$ has thus dimensions $8N$. Using a constant velocity model [25], the dynamics of each MM can be described as:

$$\boldsymbol{A}_i = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & T & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

This block is repeated N times, to construct matrix $\boldsymbol{A}$ as:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_2 & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}_N \end{bmatrix}$$

The entries of $\boldsymbol{A}$ for any two state vectors $\boldsymbol{x_{M,i}}$ and $\boldsymbol{x_{M,j}}$, with $i \neq j$, are equal to zero as it is assumed that the state of one MM does not influence the dynamics of all others. Matrix $\boldsymbol{A}$ has dimensions $8N \times 8N$. It is assumed that vector $\boldsymbol{u}_k$ cannot be measured, and hence all entries of matrix $\boldsymbol{B}$ are equal to zero.

Matrix $\boldsymbol{Q}$, that describes the process noise, is constructed by repeating the following block N times, analogously to matrix $\boldsymbol{A}$:

$$\boldsymbol{Q}_i = \begin{bmatrix} \frac{T^4}{4}\sigma_x^2 & 0 & 0 & \frac{T^3}{2}\sigma_x^2 & 0 & 0 & 0 & 0 \\ 0 & \frac{T^4}{4}\sigma_y^2 & 0 & 0 & \frac{T^3}{2}\sigma_y^2 & 0 & 0 & 0 \\ 0 & 0 & \frac{T^4}{4}\sigma_z^2 & 0 & 0 & \frac{T^3}{2}\sigma_z^2 & 0 & 0 \\ \frac{T^3}{2}\sigma_x^2 & 0 & 0 & T^2\sigma_x^2 & 0 & 0 & 0 & 0 \\ 0 & \frac{T^3}{2}\sigma_y^2 & 0 & 0 & T^2\sigma_x^2 & 0 & 0 & 0 \\ 0 & 0 & \frac{T^3}{2}\sigma_z^2 & 0 & 0 & T^2\sigma_x^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & T^2\sigma_\theta^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & T^2\sigma_\varphi^2 \end{bmatrix}$$

Where each σ denotes the standard deviation of the acceleration in each dimension. These values can be tuned to exploit a priori knowledge about the movement of the MMs. Low values should be used for the dimensions were no movement is expected.

Matrix $\boldsymbol{R}$ describes the uncertainty of the sensor readings and was set to:

$$\boldsymbol{R} = \boldsymbol{I}_{3P\times 3P} \begin{bmatrix} \sigma_1^2 \\ \vdots \\ \sigma_j^2 \\ \vdots \\ \sigma_L^2 \end{bmatrix}$$

Where $\boldsymbol{I}_{3Px3P}$ is the $3P \times 3P$ identity matrix, and σ describes the standard deviation of the readings of each sensor. P corresponds to the total number of 3D MF sensors. This choice of $\boldsymbol{R}$ assumes that there is no correlation between the noise signals.

Our UKF implementation used the Cholesky decomposition to calculate the square root of matrix $(L+\lambda)\boldsymbol{P}^{\alpha}_{k|k-1}$. To compute the inverse of matrix $\boldsymbol{P}_{yy}$ the LUP/LUPQ decomposition was used. The following values were used to tune the filter [19]: β = 2, κ = 0, α = 0.001.

REFERENCES

[1] V. Pasku *et al.*, "Magnetic Field-Based Positioning Systems," *IEEE Commun. Surv. Tutorials*, vol. 19, no. 3, pp. 2003–2017, 2017.

[2] J. M. Gilbert *et al.*, "Isolated word recognition of silent speech using magnetic implants and sensors," *Med. Eng. Phys.*, vol. 32, no. 10, pp. 1189–1197, 2010.

[3] E. J. Rouse, D. C. Nahlik, M. A. Peshkin, and T. A. Kuiken, "Development of a Model Osseo-Magnetic Link for Intuitive Rotational Control of Upper-Limb Prostheses," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 19, no. 2, pp. 213–220, Apr. 2011.

[4] T. Penzkofer *et al.*, "Free-hand CT-based electromagnetically guided interventions: Accuracy, efficiency and dose usage," *Minim. Invasive Ther. Allied Technol.*, vol. 20, no. 4, pp. 226–233, Jul. 2011.

[5] S. Leong *et al.*, "Electromagnetic navigation bronchoscopy: A descriptive analysis," *J. Thorac. Dis.*, vol. 4, no. 2, pp. 173–85, Apr. 2012.

[6] G. Ciuti, P. Valdastri, A. Menciassi, and P. Dario, "Robotic magnetic steering and locomotion of capsule endoscope for diagnostic and surgical endoluminal procedures," *Robotica*, vol. 28, no. 2, pp. 199-207, 2010.

[7] D. Wong, J. Wang, E. Steager, and V. Kumar, "Control of multiple magnetic microrobots," in *Proc. of the International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Boston, MA, USA, 2015.

[8] E. Diller, J. Giltinan, and M. Sitti, "Independent control of multiple magnetic microrobots in three dimensions," *The International Journal of Robotics Research*, vol. 32, no. 5, pp. 614-631, 2013.

[9] M. P. Kummer *et al.*, "OctoMag: An Electromagnetic System for 5-DOF Wireless Micromanipulation," *IEEE Trans. Robotics*, vol. 26, no. 6, pp. 1006-1017, 2010.

[10] S. Tarantino, F. Clemente, D. Barone, M. Controzzi, and C. Cipriani, "The myokinetic control interface: tracking implanted magnets as a means for prosthetic control," *Scientific Reports*, vol. 7, no. 1, 2017.

[11] O. Talcoth, and T. Rylander, "Optimization of sensor positions in magnetic tracking," *Chalmers University of Technology*, Gothenburg, pp. 1–30, 2011.

[12] A. M. Franz, T. Haidegger, W. Birkfellner, K. Cleary, T. M. Peters, and L. Maier-Hein, "Electromagnetic Tracking in Medicine - A Review of Technology, Validation, and Applications," *IEEE Trans. Med. Imaging*, vol. 33, no. 8, pp. 1702–1725, Aug. 2014.

[13] S. Song *et al.*, "6-D magnetic localization and orientation method for an annular magnet based on a closed-form analytical model," *IEEE Transactions on Magnetics*, vol. 50, no. 9, pp. 1-11, 2014.

[14] W. Yang, C. Hu, M. Q. Meng, S. Song, and H. Dai, "A Six-Dimensional Magnetic Localization Algorithm for a Rectangular Magnet Objective Based on a Particle Swarm Optimizer," IEEE Trans. Magnetics, vol. 45, no. 8, pp. 3092-3099, 2009.

[15] T. Naara, S. Suzuki, and S. Ando, "A Closed-Form Formula for Magnetic Dipole Localization by Measurement of Its Magnetic Field and Spatial Gradients," *IEEE Trans. Magnetics*, vol. 42, no. 10, pp. 3291-3293, 2006.

[16] D. Russel, and P. Goldsmith, "A Neural Network Driven Sensor Array for Tracking a Permanent Magnet," in *Proc. of the International Conference on Intelligent Systems and Control*, Calgary, Alberta, Canada, July 2017.

[17] D. W. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *J. Soc. Ind. Appl. Math*, vol. 11, no. 2, pp. 431-441, June 1963.

[18] N. Wahlström, and F. Gustafsson, "Tracking Position and Orientation of Magnetic Objects Using Magnetometer Networks," *Linköping University Electronic Press*, 2015.

[19] S. J. Julier, and J. K. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401-422, Mar. 2004.

[20] C. Hu, M. Q. Meng, and M. Mandal, "Efficient magnetic localization and orientation technique for capsule endoscopy," presented at the *IEEE International Conference on Intelligent Robots and Systems*, Edmonton, Alberta, Canada, Aug. 2005.

[21] C. R. Taylor, H. G. Abramson and H. M. Herr, "Low-Latency Tracking of Multiple Permanent Magnets," *IEEE Sensors Journal*, 2019.

[22] C. Hu *et al.*, "A new tracking system for three magnetic objectives," *IEEE Transactions on Magnetics*, vol. 46, no.12, pp. 4023-4029, 2010.

[23] S. Tarantino, F. Clemente, A. De Simone, and C. Cipriani, "Feasibility of tracking multiple implanted magnets with a myokinetic control interface: simulation and experimental evidence based on the point dipole model," *IEEE Transactions on Biomedical Engineering*, 2019.

[24] F. Clemente, V. Ianniciello, M. Gherardini, C. Cipriani, "Development of an Embedded Myokinetic Prosthetic Hand Controller," *Sensors*, vol. 19, no. 14, 2019.

[25] M. Birsan, "Recursive Bayesian Method for Magnetic Dipole Tracking with a Tensor Gradiometer", *IEEE Trans. Magnetics*, vol. 47, no. 2, pp. 409-415, 2011.

[26] A. R. Conn, N. I. M. Gould, and P. L. Toint, "Trust-Region Methods," Philadelphia: Society for Industrial and Applied Mathematics and Mathematical Programming Society, 2000.

[27] M. N. O. Sadiku, "Elements of electromagnetics", 6th ed., Oxford University Press, 2014.

[28] W. Andrä *et al.*, "A novel method for real-time magnetic marker monitoring in the gastrointestinal tract," *Phys. Med. Biol.*, vol. 45, no. 10, pp. 3081–3093, 2000.

[29] V. Schlageter, P. A. Besse, R. S. Popovic, and P. Kucera, "Tracking system with five degrees of freedom using a 2D-array of Hall sensors and a permanent magnet," *Sensors Actuators A Phys.*, vol. 92, no. 1–3, pp. 37–42, 2001.

[30] T. M.Greiner, "Hand Anthropometry of U.S. Army Personnel," Technical Report, United States Army Natick Research, Development and Engineering Center, Natick, pp. 1–434, 1991.

[31] F. Devernay, "C/C++ Minpack,", 2007. Available: http://devernay.free.fr/hacks/cminpack/

[32] F. V. Berghen, "Levenberg-Marquardt algorithms vs trust region algorithms," *IRIDIA, Université Libre de Bruxelles*, 2004.

[33] S. Agarwal *et al.*, "Ceres Solver," available: http://ceres-solver.org, last accessed: 13th September, 2019.