CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	An open framework enabling electromagnetic tracking in image-guided interventions
Authors	Jaeger, Herman Alexander;Hinds, Stephen;Cantillon-Murphy, Pádraig
Publication date	2018-09-13
Original Citation	Jaeger H. A., Hinds S. and Cantillon-Murphy P. (2018)'An open framework enabling electromagnetic tracking in image-guided interventions', in Frangi A., Schnabel J., Davatzikos C., Alberola-López C., Fichtinger G. (eds) Medical Image Computing and Computer Assisted Intervention – MICCAI 2018. 21st International Conference on Medical Image Computing & Computer Assisted Intervention, Granada, Spain, 16-20 September. Lecture Notes in Computer Science, vol 11073. Cham: Springer International Publishing, pp. 168-175. doi:10.1007/978-3-030-00937-3_20
Type of publication	Conference item
Link to publisher's version	10.1007/978-3-030-00937-3_20
Rights	© 2018, Springer International Publishing AG. The final authenticated publication is available at Springer via https://doi.org/10.1007/978-3-030-00937-3_20
Download date	2026-10-13 12:17:01
Item downloaded from	https://hdl.handle.net/10468/7081

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

An Open Framework Enabling Electromagnetic Tracking in Image-Guided Interventions

Herman Alexander Jaeger[1], Stephen Hinds[1] and Pádraig Cantillon-Murphy[123]

[1] University College Cork, Cork, Ireland
[2] Tyndall National Institute, Cork, Ireland
[3] Institute of Image Guided Surgery, Strasbourg, France

Abstract. Electromagnetic tracking (EMT) is a core platform technology in the navigation and visualisation of image-guided procedures. The technology provides high tracking accuracy in non-line-of-sight environments, allowing instrument navigation in locations where optical tracking is not feasible. Integration of EMT in complex procedures, often coupled with multi-modal imaging, is on the rise, yet the lack of flexibility in the available hardware platforms has been noted by many researchers and system designers. Advances in the field of EMT include novel methods of improving tracking system accuracy, precision and error compensation capabilities, though such system-level improvements cannot be readily incorporated in current therapy applications due to the 'blackbox' nature of commercial tracking solving algorithms. This paper defines a software framework to allow novel EMT designs and improvements become part of the global design process for image-guided interventions. In an effort to standardise EMT development, we define a generalised cross-platform software framework in terms of the four system functions common to all EMT systems; *acquisition*, *filtering*, *modelling* and *solving*. The interfaces between each software component are defined in terms of their input and output data structures. An exemplary framework is implemented in the Python programming language and demonstrated with the open-source Anser EMT system. Performance metrics are gathered from both Matlab and Python implementations of Anser EMT considering the host operating system, hardware configuration and acquisition settings used. Results show indicative system latencies of 5 ms can be achieved using the framework on a Windows operating system, with decreased system performance observed on UNIX-like platforms.

1 Introduction

The development of new image guided therapies relies heavily on intelligently combining data from multiple hardware sources. New techniques combining ultrasound and electromagnetic tracking (EMT) [1, 2] are among techniques which combine multiple data sources to enhance the safety and accuracy of procedures. Progress in these areas is made possible by the standardised open protocols [3, 4] that govern how hardware and software should interact with one another. Fig. 1 shows the generalised design flow of many image guided interventions (IGI). IGI

applications interact with hardware through vendor authored application programming interfaces (APIs) or interface toolkits such as PLUS and IGSTK[5]. Such toolkits provide standardised methods through which IGI applications and hardware can interact. From a software perspective, this standardised approach enables developers to prototype and apply their work in a manner that can be distributed and replicated. That said, IGI application development typically falls short of incorporating custom innovations in tracking hardware. Electromagnetic tracking systems in particular are very much considered 'blackboxes' from the perspective of IGI research.

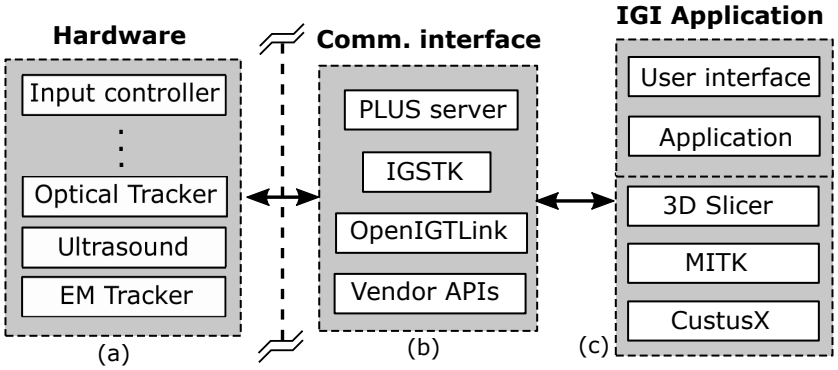

Fig. 1. A standard approach for developing cross-platform IGI applications. (a) Proprietary hardware is interfaced using an API or open communications interface (b). (c) Guided therapy applications use this interface to ensure cross-platform compatibility.

This paper outlines a framework to encourage integration of new electromagnetic tracking hardware into the current IGI design flow. The resulting framework was implemented in the Python programming language and applied to the open-source Anser EMT system [6] shown in Fig. 2. Preliminary cross-platform functionality of the framework is demonstrated with important performance metrics reported.

2 Framework Design

Electromagnetic tracking systems are complex electronic systems that incorporate advanced analog circuit design, signal processing and optimisation techniques. While the precise topology of such systems will vary depending on the core design and manufacturer, all EMT systems can be distinguished by four common processing steps outlined in Fig. 3: *acquisition*, *filtering*, *modelling*, and *solving*. Each of these processing steps can be treated as a discrete, independent stage in the tracking system software pipeline. The designed framework is structured according to these stages.

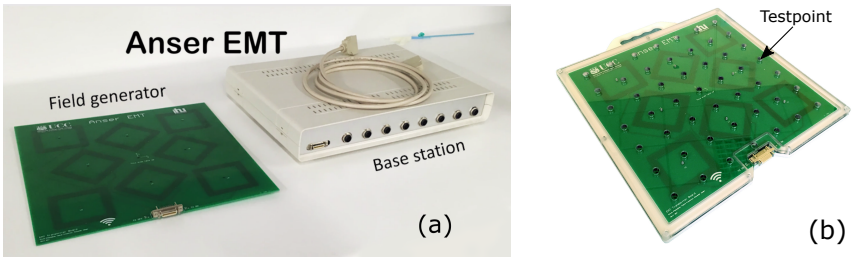

Fig. 2. (a) The Anser EMT system v1.0. (b) Field generator enclosure.

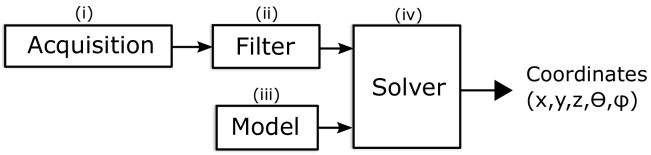

Fig. 3. The basic components which comprise all electromagnetic tracking systems are shown. The 5 degree-of-freedom sensor coordinate $[x, y, z, \theta, \varphi]$ is the most basic case for a symmetrical sensor coil, where θ and φ correspond to yaw and pitch angles respectively.

(i) Acquisition. The most fundamental step in any measurement system is data acquisition. It is the process by which physical signals are digitised into forms suitable for computation. In the case of EMT systems, the signals being measured are typically electric voltages. Such signals are induced on a tracking sensor coil when it is placed in the tracking volume of an EMT system. Acquisition hardware in the form of an analogue-to-digital converter (ADC) converts the electrical signal into a digital data stream by sampling at fixed periodic intervals. Manufacturers of acquisition systems include National Instruments (Austin, TX, U.S.A.) and Measurement Computing Corp. (Norton, MA, U.S.A).

(ii) Filtering. The sampled sensor signal contains all the necessary physical information required to resolve the sensor's position in space. EMT systems typically operate using multiple transmitters operating at distinct frequencies, thus the sampled sensor signal is a linear sum of the individual frequency components generated by each transmitting coil, as well as noise from the surrounding environment. The filter extracts the relevant signal content. A combination of digital filtering and Fourier methods are typically employed.

(iii) Model. An accurate model of the tracking system's generated magnetic field is a necessary component in all system designs. Models are typically defined as analytical expressions which define the spacial distribution of magnetic fields in the volume around the field transmitter. Each constituent transmitting coil

can be characterised by a vector equation relating the magnetic flux density vector $\boldsymbol{B}$ to a point in space, $\boldsymbol{p}$:

$$\boldsymbol{B}(\boldsymbol{p}) = [B_x^p, B_y^p, B_z^p] \tag{1}$$

Commonly used models include variations of the magnetic dipole approximation [7], Biot-Savart law [8, 9] and mutual inductance models [10]. Numerical models may also be used in cases where no accurate closed-form solution for the magnetic field exists.

(iv) Solver. Magnetic tracking systems resolve sensor positions through a process of non-linear optimisation in which a cost function is minimised to yield the best-fit solution for the position and orientation of the tracking sensor. The cost function is generally formulated such that the squared difference between the magnetic model of the sensor coil and acquired sensor measurements are minimised. Full formulations of the sensor model can be found in [8]. A non-linear least-squares approach is usually required to yield accurate results. Examples of general solving methods include the well known Levenburg-Marquard [11] and trust-region algorithms [12]. The general form of the optimisation problem is shown in (2):

$$\begin{aligned} \underset{\mathbf{p}}{\text{minimise}} \quad & \sum f_i(\mathbf{p})^2 \\ \text{subject to} \quad & lb \leq \mathbf{p} \leq ub, i = 1, \ldots, n \end{aligned} \tag{2}$$

where f_i is the i^{th} cost function relating a single frequency component of the tracking sensor signal to the field model of the corresponding transmission coil, lb and ub are upper and lower bound constraints for the solving algorithm (if applicable) and $\mathbf{p}$ is the vector argument representing the position and orientation of the tracking sensor.

3 Framework Implementation

The proposed EMT framework is composed of four Python modules representing each of the signal processing steps shown in Fig. 1. An expansion of this design showing the data-flow between modules is shown in Fig. 4. The framework is divided into four modules labelled (i) to (iv). Analog signals from the EMT sensing electronics are acquired through a data acquisition module *daq.py*. This module provides facilities to abstract the acquisition hardware's specific API into a standard interface. The acquired digital samples are fed into the filter module *filter.py* where the frequency components of interest are conditioned and extracted from the digital waveform. The filter module allows easy configuration of filter parameters while providing routines for efficient matrix multiplications required during filtering operations. The extracted signal information is then fed to the solver module *solver.py*. Simultaneously the magnetic model of the system *model.py* is compared with the extracted signal data to minimise the system cost function. The solver module provides access to the tolerance and parameter

settings for the minimisation process. The resulting sensor coordinates from the solver can be streamed to the user application using OpenIGTLink [4].

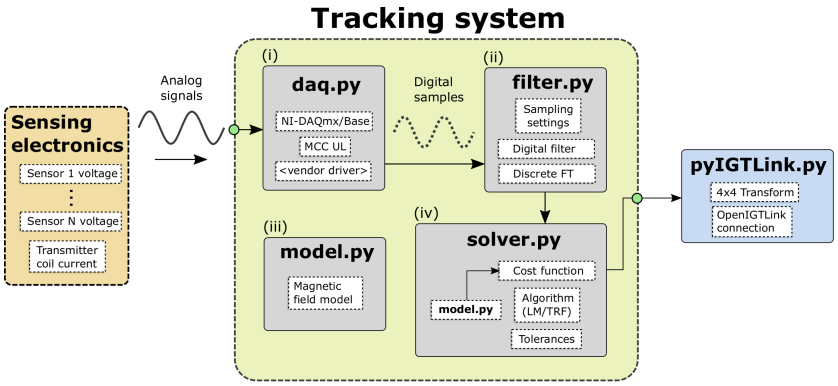

Fig. 4. (i) Analogue signals are sampled by the data acquisition module. (ii) The samples are filtered in software with relevant frequency components extracted. (iii)-(iv) A cost function utilises the component magnitudes to yield a position vector which can be transmitted using OpenIGTLink as a 4x4 transformation matrix.

The proposed framework was applied to the Anser EMT project [6]. The original open-source codebase, which was originally implemented in Matlab[§], was fully converted to Python in order to conform with the framework. The PyDAQmx driver [13] was used to provide a cross-platform interface with the NI-DAQmx driver acquisition system (National Instruments, Austin, Texas). Filtering and solving operations were performed using NumPy and SciPy libraries [14]. OpenIGTLink connectivity was achieved using PyIGTLink [15].

4 Experiments and Results

EMT framework's performance and accuracy were tested relative to the original Windows-only Matlab implementation. Testing of the framework was performed on three operating systems: Windows 10, MacOS 10.13 and Cent-OS 7.0 Linux. All reported metrics result from tests performed on a laptop PC configured with an Intel i7 4810HQ 3.5GHz CPU and 16GB of RAM, utilising both a Windows 10 and CentOS 7.0 installation. Compatibility with the MacOS 10.13 operating system was confirmed using a separate machine, but comparative results were not possible due to significant differences in the laptop hardware configuration. The variable sampling frequency of the acquisition system was set at 100kHz for all experiments.

[§]Available at https://openemt.org

4.1 Performance benchmark

Performance testing measured the framework's ability to stream position measurements as quickly as possible with minimal latency over an OpenIGTLink connection to CustusX and 3DSlicer. Acquisition latency, maximum update frequency (with both a stationary and moving sensor) were recorded for multiple acquisition frame sizes shown in Tab. 1 and 2. An acquisition frame constitutes the number of samples gathered by the acquisition system per single resolved position. A finite acquisition time for each frame puts a limit on the minimum latency figure. Static refers to a slow moving sensor speeds of <5cm per second while 'Dynamic' refers to speeds >50cm per second.

Table 1. Performance measurements using Matlab on Windows.

Matlab	Frame size	Acq. Latency (ms)	Max. Static (Hz)	Max. Dynamic (Hz)
-	250	2.5	72	64
-	500	5	70	61
-	1000	10	65	62
-	2000	20	50	50
-	5000	50	20	20

Table 2. Performance measurements using Python framework (Windows/Linux).

Python	Frame size	Acq. Latency (ms)	Max. Static (Hz)	Max. Dynamic (Hz)
-	250	2.5/5000+*	138/120	84/70
-	500	5/2000+	142/115	95/83
-	1000	10/1000+	102/80	66/62
-	2000	20/100	51/45	52/45
-	5000	50/50	20/21	20/20

* Due to operating system driver, see discussion.

4.2 Accuracy benchmark

Accuracy testing consisted of comparing errors between sensor positions obtained from a 7x7 plane test x-y grid providing a total of 49 points. 150 position acquisitions at a height of $z = 70$mm from the transmitter board (Fig. 2 (b)) were recorded per grid point from which the mean x, y and z coordinate of each point was calculated. Measurements were obtained from both Matlab and Python implementations. Maximum, minimum and root-mean-square (RMS) errors were calculated between the two obtained point grids. The grid obtained using the Matlab implementation was used as the reference since its performance has already been characterised in [6]. Maximum and minimum grid errors were measured as 3.1mm and 0.1mm respectively with an RMS error of 0.9mm with a standard deviation of 0.75mm.

5 Discussion

Tab. 2 shows how performance of the Anser EMT system varies with acquisition sample size over Windows and Linux operating systems. Benchmark results

on Windows are clearly favourable to Linux particularly at low sample sizes. The high latencies in Linux were found to be caused by a low-level buffer issue caused by limitations in the NI-DAQmx Base kernel driver for Linux. Forcing the acquisition time to be greater than the solving time prevents this latency issue occurring. Empirically setting the acquisition frame to 2000 (i.e. a frame acquisition time of 20ms) mitigates the issue, although this limits the maximum effective position update rate.

It can also be seen that the update speeds vary significantly depending on the movement speed of the sensor. This is due to the previous sensor position being used as the initial condition for the solver during each position update. A moving sensor causes previous sensor positions to lie further from the current true sensor position, resulting in the solver requiring more iterations to converge to the global minimum. Artificially forcing periodicity by limiting the update rate of the system prevents this issue from occurring. This approach must be implemented with care in order to avoid increasing the overall system latency.

The purpose of the accuracy benchmark is to showcase any significant differences between the two EMT software implementations. The benchmark is limited since it uses the Matlab implementation results as the reference standard, since no gold standard was available for the experiment. The reported mean error value of 0.9mm falls within the mean error value of the Matlab implementation of 1.14mm [6]. From this we can conclude that the Python implementation is of similar accuracy to the original implementation. Characterisation of this system according to the Hummel protocol [16, 17] is necessary in order to fully validate the accuracy of the system under the new framework.

Conclusion

An open source framework for designing electromagnetic tracking systems has been proposed. The framework was applied to a previously characterised tracking system with performance results reported. It is hoped that this work will assist in the translation of new EMT modules and platforms from research into the clinical setting.

References

1. Franz, A.M., Seitel, A., Bopp, N., Erbelding, C., Cheray, D., Delorme, S., Grünwald, F., Korkusuz, H., Maier-Hein, L.: First clinical use of the EchoTrack guidance approach for radiofrequency ablation of thyroid gland nodules. Int. J. Comput. Assist. Radiol. Surg. **12**(6) (2017) 931–940
2. Paolucci, I., Schwalbe, M., Prevost, G.A., Lachenmayer, A., Candinas, D., Weber, S., Tinguely, P.: Design and implementation of a dynamic navigation technique for laparoscopic ablation of liver tumors based on electromagnetic tracked ultrasound. Surg. Endosc. **32**(7) (2018) 3410–3419
3. Lasso, A., Heffter, T., Rankin, A., Pinter, C., Ungi, T., Fichtinger, G.: PLUS: Open-source toolkit for ultrasound-guided intervention systems. IEEE Trans. Biomed. Eng. **61**(10) (2014) 2527–2537

4. Tokuda, J., Fischer, G.S., Papademetris, X., Yaniv, Z., Ibanez, L., Cheng, P., Liu, H., Blevins, J., Arata, J., Golby, A.J., Kapur, T., Pieper, S., Burdette, E.C., Fichtinger, G., Tempany, C.M., Hata, N.: OpenIGTLink: An open network protocol for image-guided therapy environment. Int. J. Med. Robot. Comput. Assist. Surg. **5**(4) (2009) 423–434
5. Enquobahrie, A., Cheng, P., Gary, K., Ibanez, L., Gobbi, D., Lindseth, F., Yaniv, Z., Aylward, S., Jomier, J., Cleary, K.: The Image-Guided Surgery Toolkit IGSTK: An open source C++ software toolkit. J. Digit. Imaging **20**(SUPPL. 1) (2007) 21–33
6. Jaeger, H.A., Franz, A.M., O'Donoghue, K., Seitel, A., Trauzettel, F., Maier-Hein, L., Cantillon-Murphy, P.: Anser EMT: the first open-source electromagnetic tracking platform for image-guided interventions. Int. J. Comput. Assist. Radiol. Surg. **12**(6) (2017) 1059–1067
7. Li, M., Bien, T., Rose, G.: FPGA based electromagnetic tracking system for fast catheter navigation. Int. J. Sci. Eng. Res. **4**(9) (2013)
8. O'Donoghue, K., Eustace, D., Griffiths, J., O'Shea, M., Power, T., Mansfield, H., Cantillon-Murphy, P.: Catheter position tracking system using planar magnetics and closed loop current control. IEEE Trans. Magn. **50**(7) (2014) 1–9
9. Sonntag, C.L.W., Sprée, M., Lomonova, E.A., Duarte, J.L., Vandenput, A.J.A.: Accurate Magnetic Field Intensity Calculations for Contactless Energy Transfer Coils. In: Proc. 16th Int. Conf. Comput. Electromagn. Fields, Achen, Ger., Aachen, Germany (2007) 1–4
10. Bien, T., Li, M., Salah, Z., Rose, G.: Electromagnetic tracking system with reduced distortion using quadratic excitation. Int. J. Comput. Assist. Radiol. Surg. **9**(2) (2014) 323–332
11. Levenberg, K.: A Method for the Solution of Certain Non-Linear Problems in Least Squares. Q. Appl. Math. **2** (1944) 164–168
12. Byrd, R., Schnabel, R., Shultz, G.: A Trust Region Algorithm for Nonlinearly Constrained Optimization. SIAM J. Numer. Anal. **24**(5) (1987) 1152–1170
13. Cladé, P.: PyDAQmx : a Python interface to the National Instruments DAQmx driver (2018)
14. Oliphant, T.E.: Python for scientific computing. Comput. Sci. Eng. **9**(3) (2007) 10–20
15. Hiversen, D.: PyIGTLink: A Python implementation of OpenIGTLink. (2016)
16. Hummel, J., Figl, M., Birkfellner, W., Bax, M.R., Shahidi, R., Maurer, C.R., Bergmann, H.: Evaluation of a new electromagnetic tracking system using a standardized assessment protocol. Phys. Med. Biol. **51**(10) (2006) N205–N210
17. Franz, A.M., Haidegger, T., Birkfellner, W., Cleary, K., Peters, T.M., Maier-Hein, L.: Electromagnetic tracking in medicine -A review of technology, validation, and applications. IEEE Trans. Med. Imaging **33**(8) (2014) 1702–1725