Modeling Electromagnetic Navigation Systems

Journal Article

Author(s):
Charreyron, Samuel; Boehler, Quentin; Kim, Byungsoo; Weibel, Cameron; Chautems, Christophe; Nelson, Bradley

Publication date:
2021-08

Permanent link:
https://doi.org/10.3929/ethz-b-000468301

Rights / license:
In Copyright - Non-Commercial Use Permitted

Originally published in:
IEEE Transactions on Robotics 37(4), https://doi.org/10.1109/tro.2020.3047053

Funding acknowledgement:
165564 - Soft Magnetic Robots: Modeling, Design and Control of Magnetically Guided Continuum Manipulators (SNF)
168997 - Data-driven Methods for Artist-directed Physically-based Simulations (SNF)
743217 - Soft Micro Robotics (EC)
180861 - A Submillimeter Minimally Invasive System for Cardiac Arrhythmia Ablations (SNF)

Modeling Electromagnetic Navigation Systems

Samuel L. Charreyron[1], Quentin Boehler[1], Byungsoo Kim[2], Cameron Weibel[1], Christophe Chautems[1], and Bradley J. Nelson[1]

Abstract—**Remote magnetic navigation is used for the manipulation of untethered micro and nanorobots, as well as tethered magnetic surgical tools for minimally invasive medicine. Mathematical modeling of the magnetic fields generated by magnetic navigation systems is a fundamental task in the control of such tools for biomedical applications. We describe and compare several existing and newly developed methods for representations of continuous magnetic fields using interpolation in the context of remote magnetic navigation. Clinical-scale electromagnetic navigation systems feature nonlinear magnetization and magnetization interactions between electromagnets, which renders accurate magnetic field modeling challenging. We first introduce a method that can adapt existing linear models to correct for nonlinear magnetization, with similar performance to the current state-of-the-art nonlinear model. Furthermore, we present a method based on convolutional neural networks that is able to improve on the state-of-the-art method by a factor of 5.4.**

Index Terms—**Electromagnetic modeling, Magnetic field measurement, Magnetic fields, Medical robotics**

Acronyms

ANN	Artificial Neural Network
CNN	Convolutional Neural Network
CNN-DF	Divergence-free Convolutional Neural Network
eMNS	Electromagnetic Navigation System
FEM	Finite Element Method
GPU	Graphics Processing Unit
MAE	Mean Absolute Error
MNS	Magnetic Navigation System
MPEM	Multipole Electromagnet Model
N-MAE	Normalized Mean Absolute Error
N-RMSE	Normalized Root Mean Square Error
RBF	Radial Basis Function
RBF-G-3D	3D RBF Interpolation with Gaussian Kernel
RBF-G-DF	Divergence-free RBF with Gaussian Kernel
RBF-MQ-3D	3D RBF Interpolation with Multiquadric Kernel
RBF-MQ-DF	Divergence-free RBF with Multiquadric Kernel
RF	Random Forest
RMSE	Root Mean Square Error
S-MPEM	Saturated Multipole Electromagnet Model
SPL-3D	B-Spline 3D Interpolation
SPL-LPL	Laplacian Constrained 3D B-Spline Interpolation
TRI-3D	Tricubic 3D Interpolation
TRI-LPL	Tricubic Laplacian Constrained Scalar Field Interpolation

I. Introduction

REMOTE magnetic navigation is an actuation technology for minimally invasive surgery in which magnetic fields are used to wirelessly navigate devices containing magnetic material inside the body. Magnetic fields have the advantage of permeating biological tissue while being virtually harmless. Magnetic navigation results from generating magnetic torques and forces on the navigated magnetic agent by modulating the magnetic fields and the magnetic field gradients respectively, which are generated from outside the body of a patient. The reader is directed to [1] for a recent review of magnetic navigation.

Magnetic navigation of cardiac catheters is the single application that has seen successful clinical adoption with over 100,000 procedures completed [2], and spawning three competing magnetic navigation technologies by Stereotaxis Inc., Aeon Scientific AG[1], and Magnetecs Corp. Nevertheless, the field of magnetically-actuated micro and nanorobots has seen considerable research attention in the past decade [3], with potential applications in targeted drug delivery, minimally invasive surgery, and diagnostics.

A. Magnetic Navigation Systems

In remote magnetic navigation, magnetic fields are generated by a Magnetic Navigation System (MNS) that consists of magnets located around the body of the patient. These are either strong permanent magnets, which are rotated or translated in order to modulate the generated magnetic field, or electromagnets where the magnetic fields are modulated by the amount of electrical current that is running through conductive windings. A MNS using electromagnets is referred to as an Electromagnetic Navigation System (eMNS).

There are advantages and disadvantages to both approaches. Permanent magnet-based systems can be less expensive to manufacture at smaller-scales, magnetic field modeling is more straightforward, and patient access may be easier, since field control can be achieved with fewer magnets. MNSs can achieve larger field strengths, allow for independent control of magnetic gradients, and, in contrast to permanent magnets, can be "switched off."

B. Mathematical Modeling of MNS Field Generation

We define a mathematical model of a eMNS as a mathematical mapping between the electrical currents $\mathbf{i} \in \mathbb{R}^{N_e}$ running through the N_e electromagnets, and the magnetic field $\mathbf{b} \in \mathbb{R}^3$ that is generated inside the workspace at a position $\mathbf{p} \in \mathbb{R}^3$.

$$\mathbf{b}(\mathbf{p}) = g(\mathbf{p}, \mathbf{i}) \tag{1}$$

This publication focuses on *forward models* that predict the magnetic fields given the currents in the electromagnets. Conversely, *backward models* are used to determine the currents that result in a desired magnetic field. Although these will not be discussed here, strategies for inverting forward models are given in Appendix A.

[1] Multi-Scale Robotics Lab, ETH Zurich, Switzerland
[2] Computer Graphics Lab, ETH Zurich, Switzerland

[1] Aeon Scientific AG is a former spinoff company of the Multi Scale Robotics Lab

Magnetic fields induce magnetic torques on magnetized material, which cause locomotion via bending or rotation. Some untethered magnetic agents are also navigated using magnetic forces, which depend on the spatial variation or the gradient of the magnetic field.

A further simplification can be made when electromagnets have a linear relationship between the electromagnet currents and the magnetic fields. This occurs when the electromagnets do not contain ferromagnetic cores, or when the cores do not exhibit saturation or hysteresis. We refer to this category as a *linear magnetic model*. The advantage of such models is that they obey the principle of superposition, and the combined effect is simply the linear superposition of the individual electromagnet contributions, scaled by the current in each electromagnet, as seen in (2).

$$\mathbf{b}(\mathbf{p}) = \sum_{k=1}^{N_e} \mathbf{b}_k(\mathbf{p})\ i_k \tag{2}$$

The individual electromagnet contributions can be grouped into a position-dependent actuation matrix $\mathbf{A}_m \in \mathbb{R}^{3\times N_e}$.

$$\mathbf{b}(\mathbf{p}) = \mathbf{A}_m(\mathbf{p})\ \mathbf{i} \tag{3}$$

C. Motivations

This work is motivated by (1) the need of accurate models of MNSs in several applications of remote magnetic navigation, (2) the need of better guidelines to compare and choose a model for a given application. Choosing a method is often delicate due to the lack of common performance metrics among the community, and due to the variety of available types of methods.

Besides, accurate model are relevant for several applications, including for the control of robotic systems involving steerable medical continuum robots such as catheters [4], endoscopes [5] needles [6], and untethered micro or nanorobots [3]. The electromagnet currents or magnet positions of a MNS can be seen as joint parameters in the framework of robot kinematics, since they can be mapped to resultant forces and torques on the agent. MNS modeling is thus part of the development of accurate kinematic models.

Precise prediction of the magnetic fields is particularly relevant in the context of untethered devices that are controlled using magnetic field gradients, since these are much more sensitive to modeling inaccuracies. Most research in the literature has been limited to benchtop MNSs with small workspaces and relatively homogeneous magnetic fields, but that are too small for in vivo applications. Gradient control was demonstrated in a large-scale eMNS in [7], but the magnetic field model had to be corrected in real time using magnetic hall effect sensors placed in the workspace. Such solutions are likely to be impractical for applications featuring human patients, due to the difficulty of placing sensors in proximity to where the magnetic field is to be controlled.

Accurate models are also required in the context of magnetic positioning using quasi-static magnetic fields. Magnetically navigated surgical tools must often be located with respect to the anatomy in which they operate. In many applications of minimally invasive surgery, tools are not visible, since they are located inside the body, and live imaging using medical imaging modalities is either prohibitive (MRI or CT), of low-quality (ultrasound imaging), or harmful when (C-arm fluoroscopy). The use of onboard cameras in endoscopy may be used for localization, but is not possible for devices with small diameters, in confined or opaque spaces such as the vascular system, or inside tissue. There exist a host of electromagnetic trackers using arrays of coils generating dynamic electromagnetic fields which are detected by pickup coils embedded in surgical tools, but these are prone to interference from metallic objects, and large ferromagnetic bodies that are contained in eMNSs. A solution would be to use onboard magnetic sensors combined with magnetic field predictions from a MNS model, in order to estimate the tool configuration. This concept was demonstrated in a MNS featuring permanent magnets in [8], and [9], but has not yet been demonstrated in an eMNS. The accuracy of such a method is tightly coupled to the accuracy at which magnetic fields can be measured by the onboard sensors, and predicted by the model. For a MNS generating gradients on the order of $300\,\mathrm{mT/m}$, as could be expected in a larger eMNS such as the Aeon Phocus [7], modeling and sensing errors should not exceed $300\,\mu\mathrm{T}$ for sub-mm position accuracy. Such values are only rough estimates and would, of course, depend on the specifics of the implementation.

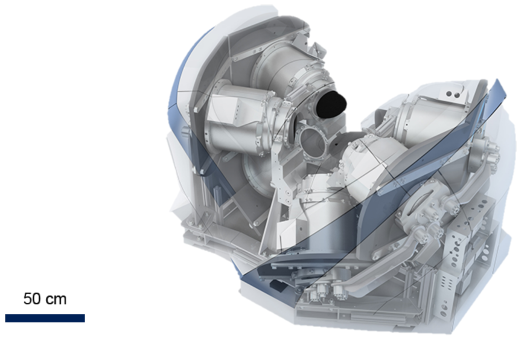

Fig. 1. The CardioMag, a large-scale eight-electromagnet eMNS

D. Contributions

The aim of this paper is to compare different methodologies for obtaining (1). *Mathematical models* can be distinguished from *electromagnetic simulations* in that the former should provide a simplified relationship between the variable parameters and the magnetic fields, and should be amenable to real-time computation, while the latter may provide accurate results at the expense of intractable computation times associated with solving the governing electromagnetic boundary-value problems. In this work, we restrict our study to mathematical models, and more particularly models which can be obtained from magnetic field data rather than from a priori information about a MNS.

We primarily focus our efforts on eMNSs with stationary electromagnets. When appropriate, we use data from the

CardioMag [10] shown in Fig. 1, a clinical-size eMNS with eight electromagnets, because it exhibits most of the properties that render modeling an eMNS complex, namely a large workspace, a large number of electromagnets, and nonlinear magnetization.

In this paper, we contrast methods that are based on mathematical interpolation, and assume that the magnetization is linearly related to electromagnet currents and that the measurements are error-free, described in III, from models that are fit to data using error minimization techniques, described in IV. In addition to a comparative analysis of eMNS modeling strategies, we introduce several modeling methods not found in the literature. Most notably, we report the first use of a Convolutional Neural Network (CNN) to obtain state-of-the-art modeling accuracy on data from the CardioMag.

We first discuss prior work in the literature in II. We then outline and compare methods for modeling general magnetic fields in free-space via interpolation in III. Methods that take into account physical constraints on the magnetic field are contrasted to fully unconstrained methods, based on interpolation of synthetic data of an eMNS. We then consider an eMNS with multiple electromagnets, accounting for electromagnet interactions in IV, and evaluating the different methods on real magnetic field data from the CardioMag. Finally, we conclude in V.

II. Related Work

A. Analytical Models

Early prototypes from Stereotaxis relied on air-core coils, which were modeled using single current loops [11]. By using the Bio-Savart law, the expression for the magnetic field magnitude $b(z)$ can be obtained in (4) for a loop of radius a with n turns and current i at a distance z of the loop center, and with μ_0 as the vacuum permeability.

$$b(z) = \frac{\mu_0}{2}\frac{na^2 i}{(a^2+z^2)^{\frac{3}{2}}} \tag{4}$$

While such a model is accurate for simple current loops, it does not hold for electromagnets with more complex designs. For uniform current distributions on simple geometries such as wires, cylinders, and solenoids, the magnetic vector potential $\mathbf{A}$ can be expressed analytically using elliptical integrals [12]. The magnetic field vector can then be obtained by taking the curl of $\mathbf{A}$.

$$\mathbf{b} = \nabla \times \mathbf{A} \tag{5}$$

In [13], lookup tables of magnetic fields were calculated at a number of discrete points by modeling eMNSs as cylinders of uniform charge and calculating the Bio-Savart integrals directly. In [14], the magnetic fields of coils of uniform charge of an eMNS were modeled using elliptical integrals.

B. Finite Element Methods

For more complex electromagnet geometries, analytic expressions often do not exist. Additionally, the problem of calculating magnetic field distributions becomes difficult in the presence of ferromagnetic materials that are magnetized in ambient fields, since they have the property of "shaping" magnetic fields. Nevertheless, ferromagnetic materials are often used in electromagnets, where current-carrying material is wound around a ferromagnetic core, because they can increase the magnet strength by several orders of magnitude. It is possible to calculate the magnetic field of any arbitrary geometry of current distributions and ferromagnetic material using Finite Element Method (FEM). In FEM, the magnetic vector potential $\mathbf{A}$, and the magnetic field are solved numerically by generating boundary-value problems [15]. By definition, FEM calculates physical quantities at a discrete set of positions on a mesh, for a given distribution of electrical charge on the mesh. More accurate and detailed field calculations can be performed by using a finer mesh, but the computational complexity scales with $O(N^3)$ where N is the number of nodes in the mesh [16]. While FEM is attractive due to its versatility, its computational requirements are prohibitive for real-time computations on standard computing hardware.

C. Interpolation of Magnetic Field Data

FEM modeling does not take into account certain effects such as manufacturing defects, inhomogeneous or mischaracterized materials, or the presence of unmodeled disturbances. Alternatively, one can directly measure the magnetic fields generated by a MNS using a magnetometer that is moved through space, or using several magnetometers simultaneously. The engineering effort can be higher than relying on simulated data, since the magnetometer must be reliably placed at a high number of positions using a robot-arm or other positioner, or alternatively, a large number of of accurately placed magnetometers is required. Magnetic field data that is available as discrete data points, either from FEM modeling or from physical measurements, can be interpolated in space to obtain a continuous function representation of the field. In [17], [18], and [19], field data was interpolated with trilinear interpolation, and with tricubic interpolation in [20]. B-Spline interpolation of the fields was performed in [21]. In [8], the magnetic field generated by an external permanent magnet was modeled using modal-basis functions using data from FEM simulation.

The advantage of interpolation is that it is completely independent of the MNS if the magnetic data is available. Provided the magnetic measurements are error-free and of sufficient resolution, interpolation can yield an accurate continuous representation of the magnetic field with very little effort required to parametrize the model. For eMNSs, the number of dimensions increases with the number of electromagnets. For systems with more than one or two electromagnets, and where the individual effects of the electromagnets cannot be separated due to nonlinear magnetization, these methods become intractable due to the curse of dimensionality.

D. Reduced Analytical Models

Between magnetic field calculations based on solutions of Maxwell's equations, and interpolation of magnetic field data, there exist a class of models that are based on simplified

MNS models. These are analytical models in that they are expressions of a valid magnetic field, but do not seek to be a true representation of the magnetic field distribution everywhere in space. Magnetic fields that are defined in a space without free-currents can be seen as the gradient of a magnetic scalar potential ψ_m. Furthermore, if the space is devoid of magnetic materials such as in free-space, the scalar potential is a solution of Laplace's equation. For a point source located at the origin, the potential at a given position in free-space, represented in axially-symmetric spherical coordinates by (r, θ), can be expressed using a multipole expansion, as shown in (6).

$$\psi_m(r,\theta) = \sum_{l=1}^{\infty} \frac{B_l}{r^{l+1}} \; P_l(\cos\;\theta) \qquad (6)$$

B_l is the multipole coefficient of order l, and $P_l(x)$ is the Legendre polynomial of order l. The orders $l = 1, 2, 3$ are often called the dipole, quadrupole, and octopole, respectively. By using measured or FEM data of the magnetic fields, one can find the unknown coefficients and positions and orientations of the point sources by fitting them in a least-squares sense. The higher order poles have a rapidly decreasing effect on the potential field as the distance r to the source increases. Therefore, given a sufficient distance from the source, a lower-order representation is often sufficient to represent the magnetic field distribution with a given precision. For example, electromagnets are modeled as single dipoles in [18] and [22]. This assumption is generally valid given that the electromagnets are sufficiently far away from the workspace, such that the dipole terms of the magnetic field dominate other terms, and far enough from each other such that they can be considered as independent sources. In [23], an eMNS was modeled with several sources per electromagnet, and the magnetic scalar potential from each source was modeled using a multipole expansion. We refer to this method as the Multipole Electromagnet Model (MPEM).

The advantage of reduced-analytical models is that they generate physically-consistent vector fields that are divergence-free, i.e. $\nabla \cdot \mathbf{b} = 0$, which is a property of all magnetic fields, and curl-free $\nabla \times \mathbf{b} = \mathbf{0}$, which is a property of magnetic fields in which there are no free currents, such as in free-space. Because they are fit to data in a least-squares sense, they show some robustness to measurement uncertainties, provided that the underlying assumptions of the model are accurate. The disadvantage of these models is the need to define the underlying parametrization. It is not inherently obvious how many sources and of what order are necessary to obtain accurate results for any MNS. Furthermore, such models are usually fit to data using iterative methods such as the Levenberg-Marquardt (LM) algorithm, which can be sensitive to initial values of the fitted parameters and only converge towards local minima.

E. Machine Learning

Recently, we investigated using machine learning to model an eMNS in [24]. The problem was cast as multivariate regression with the inputs being the electromagnet currents and the position at which to calculate the field, and the output being a magnetic field vector. Random Forests (RFs) and Artificial Neural Networks (ANNs) were both implemented, yielding improved magnetic field prediction over MPEM, which was used as a baseline. Such methods can handle the nonlinear relationship between the electromagnet currents and magnetic fields, which occurs when electromagnets exhibit saturation. Additionally, as black-box methods, they do not require knowledge of domain-specific parameters such as the number, order, position, and strength of the magnetic sources. A drawback is that they do not yield physically-consistent fields in contrast to reduced-analytical models, since there are no constraints that restrict the magnetic fields to be curl-free or divergence-free.

III. Magnetic Field Interpolation in Free Space

The following methods, also described in II-C, can be used to obtain a continuous representation of magnetic fields from discrete data. In addition to predicting field values at unmeasured locations from measurements on individual electromagnets, magnetic field interpolation can also be used to convert coarse measurements to finer ones, also known as *upscaling* the magnetic field data.

A. Structured Grid Methods

Multivariate interpolation methods either work on regular grids or on unstructured data. In this section, we assume that magnetic field data is available on a $n \times m \times l$ regular 3D grid.

1) Tricubic 3D Interpolation (TRI-3D): While some prior work used trilinear interpolation of magnetic fields [17] [20], we consider tricubic interpolation, since it not only provides continuity of the function but also of the three first derivatives of the function. This is of particular importance if one wishes to have continuous magnetic field gradients. Tricubic interpolation is a local method, because an interpolated value only depends on the eight values at the corners of the voxel containing that point. We interpolate a separate tricubic function for each component of the magnetic field, resulting in a 3×64 coefficient matrix $\mathbf{c}$. The expression of the interpolated magnetic field $\mathbf{b}$ at position (x, y, z) is

$$\mathbf{b}(x,y,z) = \sum_{i,j,k=0}^{3} \mathbf{c}_{ijk} \; x^i \; y^j \; z^k \qquad (7)$$

The 64 coefficients of each interpolant are found by solving a linear system that results from constraining the values of the field, the first derivatives of the field, and an additional set of higher order derivatives, as described in [25]. The values of the magnetic field derivatives are obtained by finite difference approximations. For the vector-valued magnetic field, each voxel of the grid is associated with three vectors of 64 coefficients, which can be precomputed and stored in a lookup-table resulting in $(n-1) \times (m-1) \times (l-1) \times 192$ coefficients. Computing the magnetic field gradient is straightforward and follows from the gradient of (7).

2) Tricubic Laplacian Constrained Scalar Field Interpolation (TRI-LPL): We can also use the fact that the magnetic field should be the gradient of a scalar potential to instead perform a tricubic interpolation of the scalar potential (TRI-LPL).

$$\psi_m(x,y,z) = \sum_{i,j,k=1}^{3} a_{ijk}\ x^i\ y^j\ z^k \tag{8}$$

The gradient of the scalar potential is constrained to equal the magnetic field values at the grid points.

$$\mathbf{b}(x,y,z) = -\nabla\psi_m \tag{9}$$

This results in an interpolant that is naturally curl-free but not necessarily divergence-free. The gradient of the magnetic field is obtained by taking the Hessian of the interpolant. It is symmetric but does not have zero trace.

The Laplacian of the interpolated potential $\nabla^2\psi_m$ is constrained to be zero at the grid points and, we also constrain the second derivatives of the interpolated potential $\frac{\partial B_x}{\partial y}$, $\frac{\partial B_x}{\partial z}$, $\frac{\partial B_y}{\partial z}$, $\frac{\partial^2 B_x}{\partial yz}$, which are obtained using finite difference approximations. The number of coefficients is $(n-1) \times (m-1) \times (l-1) \times 63$[2].

3) B-Spline 3D Interpolation (SPL-3D): Similar to polynomial interpolation, B-splines generate continuous function representations of discrete data through piecewise polynomial functions [26]. In contrast to tricubic interpolation, which only considers the values of the function on the corners of the voxel surrounding the interpolation position, B-splines have higher support, in that the adjacent voxels also contribute to the interpolation value. This increases the interpolation accuracy at an increased computational cost.

A B-spline function $f : \mathbb{R} \to \mathbb{R}$ of degree d is uniquely defined by a sequence of knots $\{t_1, \cdots, t_q\}$, and coefficients $\{c_1, \cdots, c_q\}$. The basis functions can be calculated using the recursive Cox-de Boor formula.

$$N_{i,1}(x) = \begin{cases} 1 & \text{if } t_i \le x < t_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

$$N_{i,d}(x) = \frac{x - t_i}{t_{i+d} - t_i} N_{i,d-1}(x) + \frac{t_{i+d+1} - x}{t_{i+d+1} - t_{i+1}} N_{i+1,d-1}(x) \tag{11}$$

The interpolant is a weighted linear combination of the basis functions

$$f(x) = \sum_{i=1}^{q} c_i\ N_i(x) \tag{12}$$

For multivariate functions, one can use the tensor product of B-splines. For a function of three variables, we define the sets $\{N_i(x)\}_1^q$, $\{M_j(y)\}_1^r$, $\{P_k(z)\}_1^s$ as basis functions with respective knot sequences $\{k_i\}_1^q$, $\{k_j\}_1^r$, and $\{k_k\}_1^s$. Similarly to TRI-3D, vector-valued functions are represented using vectors of coefficients for each basis function. In the SPL-3D method, we represent $\mathbf{b} : \mathbb{R}^3 \to \mathbb{R}^3$ with

$$\mathbf{b}(x,y,z) = \sum_{i,j,k=1}^{q,r,s} \mathbf{c}_{ijk}\ N_i(x)\ M_j(y)\ P_k(z) \tag{13}$$

The coefficients $\{\mathbf{c}_{ijk}\}_{i,j,k=1}^{q,r,s}$ can be obtained by solving a linear system. We stack the field values at the grid points in a matrix $\mathbf{D}_b \in \mathbb{R}^{nmp \times 3}$. We also stack the basis-function values in $\mathbf{N}_b \in \mathbb{R}^{n \times q}$, $\mathbf{M}_b \in \mathbb{R}^{m \times r}$, and $\mathbf{P}_b \in \mathbb{R}^{p \times s}$. We obtain the tensor product of the basis-function values in matrix $\mathbf{Z}_b \in \mathbb{R}^{nmp \times qrs}$.

$$\mathbf{Z}_b = (\mathbf{N}_b \otimes \mathbf{M}_b) \otimes \mathbf{P}_b \tag{14}$$

We then solve the following linear system with $\mathbf{C}_b \in \mathbb{R}^{qrs \times 3}$ as the stacked coefficients

$$\mathbf{D}_b = \mathbf{Z}_b\ \mathbf{C}_b \tag{15}$$

4) Laplacian Constrained 3D B-Spline Interpolation (SPL-LPL): In [21], a constrained version of SPL-3D was introduced, such that the divergence and curl of the interpolated magnetic field was zero at points on a separate $e \times f \times g$ grid. We call this method SPL-LPL. The system (15) is modified, such that it becomes a linearly-constrained quadratic program.

$$\begin{array}{ll} \min\limits_{\mathbf{C}_b} & ||\mathbf{D}_b - \mathbf{Z}_b\ \mathbf{C}_b||^2 \\ \text{s.t.} & \nabla \cdot \mathbf{b}(x_i, y_j, z_k) \ = \ 0 \\ & \nabla \times \mathbf{b}(x_i, y_j, z_k) \ = \ \mathbf{0} \\ & \forall i \in 1 \cdots e,\ \forall j \in 1 \cdots f,\ \forall k \in 1 \cdots g \end{array} \tag{16}$$

The $e \times f \times g$ grid of constraints can be chosen to be arbitrarily fine at the expense of the accuracy of the interpolation of field values. In this work, we use the same grid for both the field measurements and for the constraints.

For both SPL-3D and SPL-LPL, we found that using $d = \min\{n, m, l\}$ provided the best results. Knots were placed using the MATLAB curve fitting toolbox's `aptknt` acceptable knot sequence function.

B. Unstructured Methods

The magnetic field data can be specified at arbitrary positions and does not need to be on a regular grid. This is particularly useful when one cannot specify where the data is obtained, for example when a dataset has already been collected, if it contains "holes" in the measurements, or when the data is not located on a grid.

1) Scalar Radial Basis Functions: Radial Basis Functions (RBFs) are linear combinations of smooth basis functions that are centered at N control points $\{\mathbf{r}_i\}_1^N$. The 3D multivariate vector-valued RBF $\mathbf{b} : \mathbb{R}^3 \to \mathbb{R}^3$ can be expressed as

$$\mathbf{b}(\mathbf{p}) = \sum_{i=1}^{N} \phi(||\mathbf{p} - \mathbf{r}_i||)\ \mathbf{c}_i \tag{17}$$

where $\mathbf{c}_i \in \mathbb{R}^3$ are the weights associated with each basis function. $\phi : \mathbb{R} \to \mathbb{R}$ is called a *kernel* and several options can be found in the literature [27]. In this work we limit ourselves

[2] the constant parameter a_{000} of the tricubic interpolant can be set to any value, since it is not be affected by the derivatives.

to two of the most popular, the Gaussian and multiquadric kernels. The Gaussian kernel is defined as

$$\phi_g(x) = e^{-\epsilon |x|^2} \tag{18}$$

while the multiquadric kernel has the following expression

$$\phi_{mq}(x) = \sqrt{1 + \epsilon\ |x|^2} \tag{19}$$

Both kernels are infinitely differentiable with C^∞ continuity, and have global support, since they are always non-zero. A *shape parameter* ϵ controls the amount of "spread" between the control points. If ϵ is small, the basis functions are flatter and spread to neighboring control points. We call 3D RBF interpolation with a Gaussian and multiquadric kernel RBF-G-3D and RBF-MQ-3D, respectively.

The measurement values are grouped into a matrix $\mathbf{D}_r \in \mathbb{R}^{N\times 3}$, and the kernels are evaluated at all the measurement positions to form an interpolation matrix $\mathbf{A}_r \in \mathbb{R}^{N\times N}$. The coefficients $\mathbf{C}_r \in \mathbb{R}^{N\times 3}$ can be found by solving the following linear system.

$$\mathbf{D}_r = \mathbf{A}_r\ \mathbf{C}_r \tag{20}$$

2) Divergence-free Radial Basis Functions: Matrix-valued RBFs can be used to represent vector fields with physical properties such as zero divergence or curl [28]. NB. there is no known matrix-valued kernel that has both properties, so we focus on divergence-free kernels, since it is a property of all magnetic fields. A divergence-free matrix-valued RBF $\Phi(x) \in \mathbb{R}^{3\times 3}$ can be obtained from a scalar kernel by using the following operation

$$\Phi(x) = (-\nabla^2\ \mathbf{I}_3 + \nabla\nabla^T)\ \phi(x) \tag{21}$$

The expression of the interpolant is similar to the scalar case

$$\mathbf{b}(\mathbf{p}) = \sum_{i=1}^{N} \Phi(||\mathbf{p} - \mathbf{r}_i||)\ \mathbf{c}_i \tag{22}$$

The coefficients are also found by solving a linear system (20). In this case, $\mathbf{A}_r \in \mathbb{R}^{N\times 3\times 3\times N}$ is a tensor with four dimensions and is contracted with $\mathbf{C}_r \in \mathbb{R}^{N\times 3}$ over the third and fourth dimensions to obtain $\mathbf{D}_r \in \mathbb{R}^{N\times 3}$. We refer to the two matrix-valued methods as Divergence-free RBF with Gaussian Kernel (RBF-G-DF) and Divergence-free RBF with Multiquadric Kernel (RBF-MQ-DF).

C. Results on Synthetic Data

To evaluate and compare the different proposed magnetic field interpolation methods, we used synthetic data that was obtained from an existing eMNS model. We did so because it allowed us to easily generate large quantities of physically-consistent magnetic field and gradient data from an eMNS without resorting to simulations or measured data, and because we were concerned with the interpolation of general free-space magnetic fields rather than accurately modeling the magnetic field generation of a particular eMNS.

1) Dataset: The data was generated using a MPEM model [23] of the CardioMag using a set of 100 randomly-generated electromagnet current vectors on regular position grids of varying size. The workspace consisted of a cube of size 20 cm. For sake of brevity, we denote a grid of size N_g as a grid of size $N_g \times N_g \times N_g$. For evaluation, we used data on a grid of size $N_g = 16$.

2) Performance Metrics: The following metrics were used for evaluating magnetic field prediction accuracy. We denote a dataset of size M with scalar prediction $\mathbf{x} \in \mathbb{R}^M$ and corresponding observations $\mathbf{y} \in \mathbb{R}^M$. For vector data such as 3D fields, scalar metrics are evaluated component-wise unless specified otherwise. The Root Mean Square Error (RMSE) is

$$\text{RMSE}(\mathbf{x}, \mathbf{y}) = \sqrt{\frac{1}{M}\ \sum_{i=1}^{M} (x_i - y_i)^2} \tag{23}$$

We also include a Normalized Root Mean Square Error (N-RMSE) where the RMSE is normalized by the range of the observations. This is useful for making comparisons between datasets where the range of the data is different.

$$\text{N-RMSE}(\mathbf{x}, \mathbf{y}) = \text{RMSE}(\mathbf{x}, \mathbf{y})\ /\ (\max_i \mathbf{y} - \min_i \mathbf{y}) \tag{24}$$

The Mean Absolute Error (MAE) is

$$\text{MAE}(\mathbf{x}, \mathbf{y}) = \frac{1}{M}\ \sum_{i=1}^{M} |x_i - y_i| \tag{25}$$

and corresponding Normalized Mean Absolute Error (N-MAE) is

$$\text{N-MAE}(\mathbf{x}, \mathbf{y}) = \text{MAE}(\mathbf{x}, \mathbf{y})\ /\ (\max_i \mathbf{y} - \min_i \mathbf{y}) \tag{26}$$

The RMSE is easily affected by variability in the data, with larger errors contributing more to the combined error than small errors. It is therefore more sensitive to outliers and is always higher than the MAE [29]. The MAE in contrast gives less weight to outliers and is our preferred metric.

We also sometimes report the coefficient of determination (R^2)

$$R^2(\mathbf{x}, \mathbf{y}) = 1 - \frac{\sum_{i=1}^{M} (x_i - y_i)^2}{\sum_{i=1}^{M} (y_i - \bar{\mathbf{y}})^2} \tag{27}$$

where $\bar{\mathbf{y}}$ is the mean of all observations. The R^2 describes the "goodness of fit," and how well a model's predictions approximate the real data. Values closer to 1 indicate a strong performance while low values indicate a poor performance.

3) Choosing RBF Shape Parameters: The performance of an RBF interpolant depends on the value of its shape parameter ϵ. Due to the well-known "uncertainty relation," there is a trade-off between the numerical stability of (20) and the interpolation performance [30]. The optimal value of the shape parameter depends on the set of interpolation points and the interpolant kernel. We evaluated the effect of the shape parameter on the mean N-RMSE across all components and data points using the previously described dataset. We also recorded the condition number $\lambda_c = \text{cond}(\mathbf{A}_r)$. An example is shown in Fig. 2. According the the literature, the optimal shape parameter often lies close to the limits of machine precision

[30]. An option is to use a safe value of ϵ at the cost of a loss in performance. For regular grids and positions normalized to vary between 0 and 1, our results on the CardioMag suggest using values on the order of $\epsilon = 1$ for the RBF-MQ kernels, $\epsilon = 2.5$ for RBF-G-3D and $\epsilon = 4$ for RBF-G-DF for grid sizes below $N_g = 8$. A better option is to perform leave-one-out cross validation in order to estimate the optimal shape parameter [31] for a given dataset.

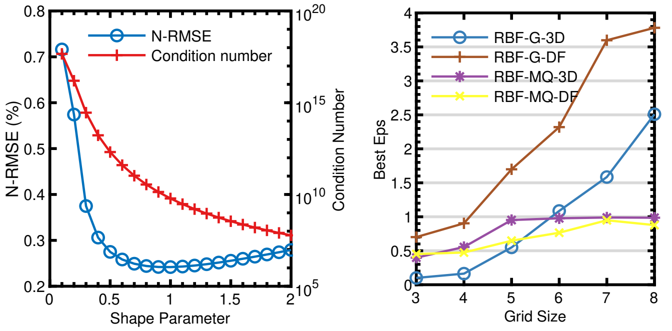

Fig. 2. Left: Illustration of the uncertainty relation between the shape parameter and the interpolation performance for RBF-MQ-3D with $N_g = 5$. The condition number is also shown as the red line. Right: Comparison of optimal shape parameters for different grid sizes for the different RBF kernels. Note that for both plots, the position values were normalized using `min-max` scaling to lie between 0 and 1.

4) Comparison of Field Interpolation Performance: We directly compared all methods. For the RBF methods, we used optimized shape parameters for the dataset positions. We performed the measurements with increasing grid sizes between 3 and 6. The average N-MAE results are shown in Fig. 3. The scalar metrics reported were averaged over all components and all positions in the dataset. The numeric values for grid size $N_g = 5$ are reported in Table I.

For all methods, there is a natural increase in performance with an increase in the grid size. Our results suggest that using grid sizes of at least $N_g = 4$ are preferable in the defined workspace of the CardioMag, since they resulted in interpolation errors of below 1% in the N-MAE for all methods. Overall the best performing methods were RBF-G-3D and SPL-3D, both exceeding all other methods on almost all of the field and gradient metrics shown in Fig. 3 and Table I. Both methods are unconstrained and have nonlocal support. We also found that the divergence-free versions of the RBF methods always performed worse than their scalar-valued counterparts, owing possibly to a worse numerical conditioning in solving (20). Amongst the different RBF kernels, the Gaussian kernel performed better for scalar valued kernels, while the multiquadric kernel performed better in the divergence-free matrix valued kernel.

While for RBF, physically-constrained methods performed worse, that was not the case with the tricubic methods. TRI-LPL consistently outperformed TRI-3D, showing that the constraints imposed by the curl-free expression, and constraints on the Laplacian at the voxel corners, both served to improve the interpolation performance. Overall we found that increased support did not necessarily lead to better performance, particularly at lower N_g values. This is particularly the case at $N_g = 3$ where TRI-3D and TRI-LPL fared well compared to RBF-G-3D and SPL-3D. At all grid values, the local TRI-LPL method outperformed the nonlocal SPL-LPL, RBF-G-DF, and RBF-MQ-DF methods.

5) Computing Magnetic Field Gradients: Magnetic gradient information is necessary for estimating the magnetic forces that act on magnetic agents in magnetic navigation. Due to the difficulty in accurately measuring magnetic field gradients, gradients are often estimated from interpolated field values rather than direct interpolation of measured or simulated gradient values. The gradient is usually represented by the 3×3 gradient matrix $\nabla \mathbf{b}$.

$$\nabla \mathbf{b}(x, y, z) = \begin{bmatrix} \frac{\partial \mathbf{b}}{\partial x} & \frac{\partial \mathbf{b}}{\partial y} & \frac{\partial \mathbf{b}}{\partial z} \end{bmatrix} \tag{28}$$

Note that because $\nabla \cdot \mathbf{b} = 0$, the gradient has 8 independent parameters since $\mathrm{tr}(\nabla \mathbf{b}) = 0$. In free-space where $\nabla \times \mathbf{b} = \mathbf{0}$, the gradient matrix is symmetric and the gradient has five independent parameters. In that case one can replace (28) with the following vector

$$\mathbf{G}_5 = \begin{bmatrix} \frac{\partial \mathbf{b}_x}{\partial x} & \frac{\partial \mathbf{b}_x}{\partial y} & \frac{\mathbf{b}_x}{\partial z} & \frac{\partial \mathbf{b}_y}{\partial y} & \frac{\partial \mathbf{b}_y}{\partial z} \end{bmatrix}^T \tag{29}$$

Gaussian and multiquadric RBF kernels both have the advantage of being C^∞ smooth, and therefore all derivatives of interpolants are continuous. In contrast, polynomial interpolants including tricubic and B-spline interpolation have limited continuity. Tricubic interpolation has C^1 continuity [25] and therefore TRI-3D has continuous gradients, while TRI-LPL does not, since the interpolant must be differentiated twice to obtain magnetic field gradients. B-spline interpolants of order d have at best $\mathbf{C}^{d-1}$ continuity, and the continuity at a given knot is C^{d-k} where k is the multiplicity of that knot.

We measured the gradient interpolation performance as in III-C4 and report the results in Fig. 3 and Table I. Note that the performance metrics were averaged across all evaluation positions and the nine components of the gradient matrix. Similarly to the field results, the best performing interpolants were again RBF-G-3D and SPL-3D.

6) Physical Considerations: We computed the divergence and curl on the evaluation grid, and compared the values predicted by the interpolants. The mean of the absolute value of divergence over all evaluation positions, and the mean of the curl magnitude at all positions is shown in Fig. 4 and in Table I. For methods that are naturally divergence-free, namely RBF-G-DF and RBF-MQ-DF, or curl-free namely TRI-LPL, the quantities are naturally zero. The divergence and curl for SPL-LPL are also negligible, showing that the constraints in (16) are successful at minimizing the curl and divergence throughout the entire workspace. This is, however, at a cost in interpolation accuracy of the gradients, as can be seen by comparing the N-MAE of SPL-LPL and SPL-3D in Table I. When physical properties are not enforced, unsurprisingly the best performing methods RBF-G-3D, RBF-MQ-3D, and SPL-3D also yield lower average field divergences and curls, since they are better able to approximate magnetic field gradients.

7) Computational Complexity: While methods with global support such as RBFs benefit from generally better interpolation performance, they are associated with a higher com-

TABLE I
FIELD AND GRADIENT INTERPOLATION PERFORMANCE RESULTS WITH $N_g = 5$

		RBF-G-3D	RBF-G-DF	RBF-MQ-3D	RBF-MQ-DF	SPL-3D	SPL-LPL	TRI-3D	TRI-LPL
Field	MAE (mT)	0.1	0.6	0.2	0.4	0.1	0.7	0.3	0.1
	N-MAE (%)	0.1	0.3	0.1	0.2	0.1	0.4	0.2	0.1
	RMSE (mT)	0.2	0.9	0.4	0.6	0.2	1.2	0.5	0.3
	N-RMSE (%)	0.1	0.5	0.2	0.3	0.1	0.6	0.3	0.2
	R^2	1.000	0.999	1.000	0.999	1.000	0.998	1.000	1.000
Gradient	MAE (mT/m)	5.3	25.3	11.9	17.8	5.2	24.0	18.9	10.7
	N-MAE (%)	0.4	1.9	1.0	1.4	0.4	1.9	1.5	0.9
	RMSE (mT/m)	14.0	41.2	23.9	30.6	14.2	43.1	40.7	25.9
	N-RMSE (%)	1.2	3.2	1.9	2.4	1.2	3.5	3.2	2.0
	R^2	0.997	0.974	0.992	0.985	0.996	0.973	0.980	0.991
	$\lvert\nabla \cdot \mathbf{b}\rvert$ (mT/m)	9.6	0.0	24.1	0.0	10.0	0.0	38.7	29.3
	$\lvert\lvert\nabla \times \mathbf{b}\rvert\rvert$ (mT/m)	14.5	105.0	32.3	71.9	14.1	0.0	57.6	0.0

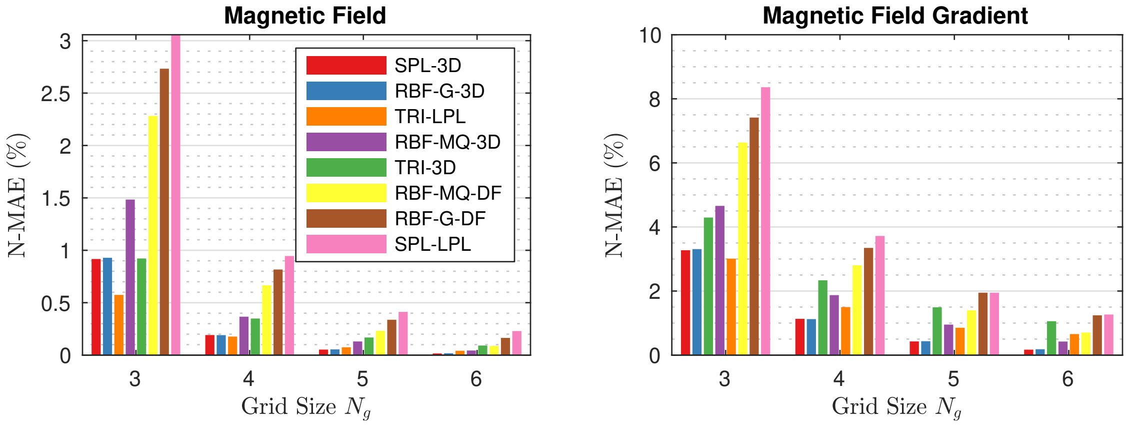

Fig. 3. Interpolation field performance comparison for all methods at increasing grid sizes

putational cost. One can distinguish the cost of setting up the interpolation and computing coefficients from the cost of computing the interpolant. In the former case, computational complexity is less of an issue in real-time control applications since it can generally be done offline. It can be a bottleneck, nonetheless, if interpolation is used to upsample a magnetic field that may come from real-time measurements, or from the output of a model yielding predictions at discrete locations. For N data locations, the setup cost of 3D RBF methods is $\mathcal{O}(N^3)$ and for large N, there can be numerical issues associated with calculating the coefficients. The interpolation cost is then $\mathcal{O}(N)$. B-spline methods generally afford from more favorable complexity over RBFs because they have more compact support and therefore sparsities in the linear algebra computations in both the set-up and interpolation. For B-spline methods of order d, the setup complexity is $\mathcal{O}(d^3N^3)$, where N is the total number of points in the grid. The interpolation cost is $\mathcal{O}(d^2N)$ [32]. For tricubic methods, the setup cost is $\mathcal{O}(N)$ and the interpolation cost is a constant, since each voxel can be considered independently from its neighbors.

The accuracy of interpolation methods can be greatly improved by increasing the grid size. Nevertheless, we have found that relatively small grid sizes $N_g < 8$ are needed to interpolate magnetic fields with high accuracy in the CardioMag, and all methods can be computed fast on modern computing hardware. In that light, we recommend using the best performing method available while disregarding computational cost. Nevertheless, in applications with finer grids, where the interpolation coefficients can not be precomputed, when upsampling a magnetic field in real-time for example, computationally favorable methods like TRI-LPL may be preferred.

IV. Modeling Electromagnet Driven Magnetic Fields

We now consider the more general nonlinear magnetic model in (1). For systems that exhibit a linear relationship between electromagnet currents and core magnetization, a simple matrix relation (2) can be used to combine the effects of multiple electromagnets. In certain systems, particularly those with large ferromagnetic cores and high current densities, as is the case for clinical scale systems such as the CardioMag, the magnetization response to electromagnet current is far from linear, since the ferromagnetic material comprising the electromagnets exhibits magnetic saturation.

A. Dataset Description

We measured magnetic field data using a 3D array of 119 Hall-effect magnetic sensors placed in the workspace of the CardioMag. The dataset contains $3,590$ distinct vectors of electromagnet currents and magnetic field readings from all

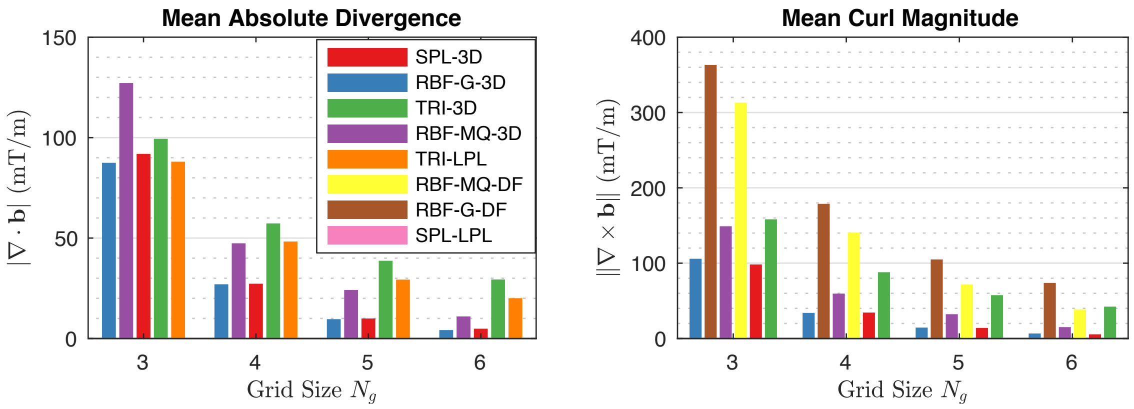

Fig. 4. Mean divergence and curl magnitude for all methods at increasing grid sizes. Methods that are curl or divergence-free are omitted from the bar plot.

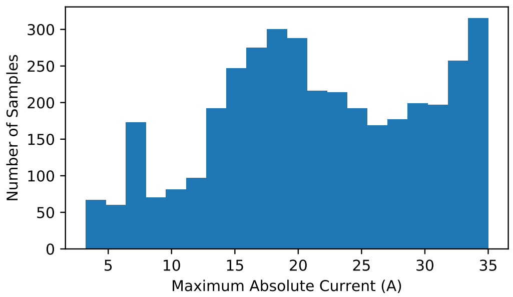

Fig. 5. Histogram of current vectors ordered by the maximum absolute value of the current in all electromagnets.

sensors. More information on the dataset can be found in our previous work [24]. We randomly split the data into training and testing datasets with a 9:1 ratio. The dataset contains current levels of varying intensity. In Fig. 5, we show the distribution of maximum absolute values in each vector of electromagnet current.

B. Methods

1) Multipole Electromagnet Model (MPEM): We trained a MPEM model of order 2 (dipole) on a subset of the previously mentioned training dataset, as it is usually considered of sufficient order to capture the fields in the workspace of an eMNS [1]. Data was discarded if the maximum electromagnet current exceeded 5 A, in order to ensure that the electromagnets did not exhibit saturation and that the linearity assumption was ensured.

2) Saturated Multipole Electromagnet Model (S-MPEM): A natural extension to linear magnetic models is to include a correction for the saturating electromagnets. For electromagnet number k, a saturation function $h_k : \mathbb{R} \to \mathbb{R}$ applies a correction based on the core saturation, and produces the *saturated current* $i_s^k = h_k(i_k)$. (2) was modified with the following scalar saturation correction.

$$\mathbf{b}_s(\mathbf{p}) = \sum_{k=1}^{N_e} \mathbf{b}_k(\mathbf{p})\; i_s^k \tag{30}$$

The saturation is minimal at low currents, so the following parametrization was used. The slope at zero current was assumed to correspond to the linear coefficients of the magnetic field at a given position $\mathbf{p}$.

$$\mathbf{b}_k(\mathbf{p}) = \left.\frac{\partial \mathbf{b}_s}{\partial i_k}\right|_{i_k=0} \tag{31}$$

The saturation function of individual magnets was determined as follows. A magnetometer was placed inside the workspace of the CardioMag. The magnetometer measured the magnitude b_s of the magnetic field $\mathbf{b}_s$ at a location $\mathbf{p}$. The currents on a given electromagnet k were ramped in steps of 0.5 A, while maintaining zero current on the other electromagnets. This was performed for each of the eight electromagnets. We tested different sigmoid functions using least-squares fitting to model $b_s(i_k)$, the relationship between the magnetic field magnitude, and the current i_k on a single electromagnet.

We selected $h_s(i_k) = \frac{\sqrt{\pi}}{2\beta_k}\,\mathrm{erf}(\beta_k i_k)$ as our preferred representation of saturation where β_k is a fitting parameter controlling the degree of saturation, since it obtained good results with a minimal number of parameters, resulting in the following expression of the saturation corrected magnetic field.

$$\mathbf{b}_s(\mathbf{p}) = \sum_{k=1}^{N_e} \mathbf{b}_k(\mathbf{p})\; \underbrace{\frac{\sqrt{\pi}}{2\beta_k}\,\mathrm{erf}(\beta_k i_k)}_{h_k(i_k)} \tag{32}$$

An example of fitting the current saturation function on the current ramp of an electromagnet is shown in Fig. 6, showing close agreement between the measurements and the saturation function that was estimated.

3) Artificial Neural Network (ANN): The modeling of an eMNS can be cast as a multivariate regression problem, which can be solved using nonlinear black-box models. In our previous work [24], we used an ANN to represent (1). The ANN consisted of three hidden layers of 100, 50, and 25 neurons with hyperbolic tangent activation functions at each hidden layer.

4) Convolutional Neural Network (CNN): Accurate simulation of 3D fluid flows using Deep Fluids, a generative CNN was shown in [33]. By modeling a vector potential, it was also shown that divergence-free fields could be obtained.

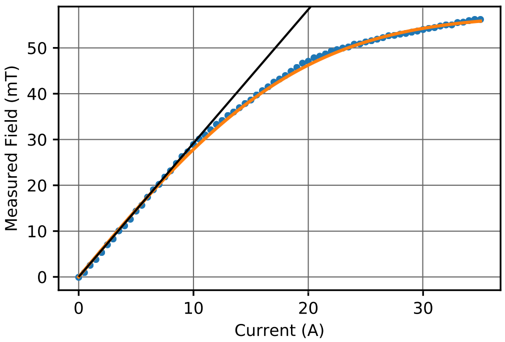

Fig. 6. Magnetic field data from ramping a single electromagnet k. The measurements using a magnetometer placed in the CardioMag are shown as the blue points. The saturation function $b_s(i_k) = a_k \text{ erf}(\beta_k i_k)$ is shown as the solid line, where β_k is the saturation parameter, i_k the current, and a_k the initial slope of the magnetization curve of that electromagnet. A straight line of equation $b_l = a_k \; i_k$ showing the linearity at low currents is shown for reference, where b_l would be the magnetic field magnitude with no saturation.

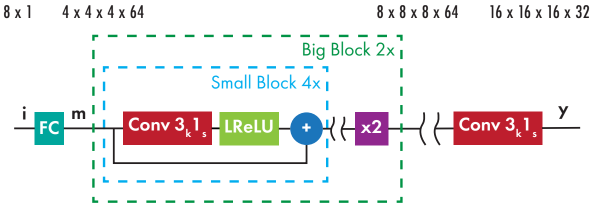

Fig. 7. Block diagram of the generative CNN used for magnetic field prediction

We used a modified version of the Deep Fluids architecture, shown in Fig. 7. The network takes as input the vector of electromagnet currents and outputs $\mathbf{y} = G(\mathbf{i})$, a discretized vector field on a regular 3D grid of size N_g. We used $N_g = 16$, since the network works with powers of two, and that level of discretization sufficiently captures the positional variations of the magnetic field over the entire workspace, as evidenced in III. We used both a regular parametrization that predicted the magnetic field directly (CNN), and a divergence-free version (CNN-DF). For CNN, the output is the predicted magnetic field $\mathbf{y} = \mathbf{b}$, while for CNN-DF it represents the magnetic vector potential $\mathbf{y} = \mathbf{A}$. In that case, the magnetic field is then obtained by taking the numerical curl of the vector potential.

The input layer is a linear fully connected layer that projects the 8×1 input vector $\mathbf{i}$ to a $4 \times 4 \times 4 \times 64$ tensor $\mathbf{m}$. Next, a convolutional layer with kernel size 3 and stride 1 (3_k1_s) was applied with 64 filters, followed by the LReLu activation function, which adds nonlinearity to the network. Skip residual networks were also used to increase training robustness by reducing the effect of “vanishing gradients.” These three elements constitute a *small block*, which is repeated four times. The output is then upsampled by a degree of two, using nearest-neighbor interpolation. This constitutes a *big block*, which is repeated twice to upsample the input tensor to size $8 \times 8 \times 8 \times 64$, followed by $16 \times 16 \times 16 \times 64$. Finally an output convolutional layer 3_k1_s produces the output tensor $\mathbf{y}$ of size $16 \times 16 \times 16 \times 3$.

We upsampled the CardioMag dataset to a $N_g = 16$ grid, using 3D RBF interpolation. The data was normalized using `min-max` normalization, such that all features varied in the range $[-1, 1]$. The same 9:1 training test split was used as in the regression models, with the same training examples being used in all methods. The only difference was the upsampling of the data to the $N_g = 16$ grid.

Deep fluids was implemented in Google’s TensorFlow library version 1.8.0 [34]. The loss function for a single training example is

$$L(\mathbf{y}) = \lambda_b \|\mathbf{b}_m - \mathbf{b}_p\|_1 + \lambda_g \|\nabla\mathbf{b}_m - \nabla\mathbf{b}_p\|_1 \tag{33}$$

where $\|\cdot\|_1$ represents the L_1 norm, $\mathbf{b}_m$ is a measurement point, and $\mathbf{b}_p$ is the magnetic field prediction output by the network, both representing magnetic fields on a $N_g = 16$ grid. λ_b and λ_g are scalar parameters that trade-off the accuracy of the magnetic field and the accuracy of the magnetic field gradient. The gradient term in the loss function serves to reduce overfitting of the network to the magnetic field values, and allows for a smoother output over space. The gradient values were approximated using numerical differentiation. Since the training data was pre-normalized, we used $\lambda_b = \lambda_g = 1$, as was done in the original Deep Fluids implementation.

The networks were trained for 350 epochs, using the Adam optimizer [35] with $\beta_1 = 0.5$ and $\beta_2 = 0.99$. We performed all computations on a computer with a NVidia Titan Xp Graphics Processing Unit (GPU) running Ubuntu 16.04. Training took approximately one hour for both CNN and CNN-DF.

Since the performance of machine learning based methods can depend on the quantity of data that is available, we evaluated the effect of the training set size on performance. Smaller sets of the training data were obtained by randomly selecting a subset between 10% and 90% of the training data, while evaluating the trained model on the same separate test set. The results are shown in Fig. 8. There is a monotonic increase in performance as more data is available for training, attested by decreasing MAE values. With up to 30% of the training data, corresponding to roughly 1000 current samples, there is a noticeable increase in performance, with the MAE being halved for both CNN and CNN-DF. The remaining 70% of training data further reduced the MAE by a factor of two, showing the diminishing returns of using more training data.

C. Field Prediction Performance

We compared magnetic field predictions to the measured magnetic fields in the test dataset. We used the same metrics as in III-C2. The metrics were averaged across all positions and field components. For CNN and CNN-DF, the data was upsampled to a grid of size $N_g = 16$ in order to match the prediction output. The results are reported in Table II. We additionally split the training set by current in order to visualize each method’s ability to handle electromagnet saturation. Data was put into a bin if the maximum absolute value of the current across all electromagnets did not exceed a threshold. We selected thresholds of 5, 10, 15, 20, 25, and 30 A. We evaluated the MAE separately for each bin and report the results in Fig. 9. The performance of the linear

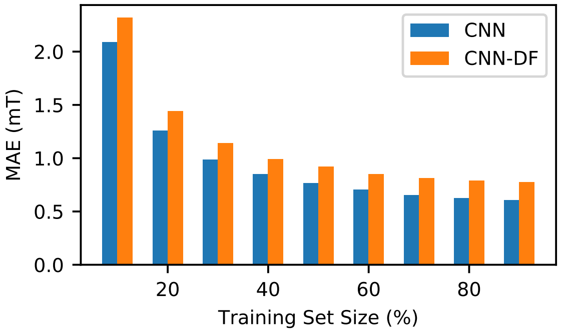

Fig. 8. Performance of the CNN-based methods depending on the training set size. The models were trained on increasingly large subsets of the initial training set. The MAE scores were averaged across all currents and positions.

TABLE II
FIELD PREDICTION PERFORMANCE COMPARISON ACROSS ALL METHODS

	MPEM	S-MPEM	ANN	CNN-DF	CNN
MAE (mT)	8.98	2.84	2.76	0.62	0.51
N-MAE (%)	2.75	0.87	0.85	0.19	0.17
RMSE (mT)	14.9	4.44	3.88	0.96	0.83
N-RMSE (%)	4.57	1.36	1.19	0.28	0.27
R^2	0.809	0.983	0.987	0.997	0.998

MPEM decreased significantly with increasing current, since it did not take into account electromagnet saturation. The ANN showed inferior performance to the S-MPEM at lower currents, possibly due to a bias towards higher current values in the dataset, as can be seen in Fig. 5. The CNN and CNN-DF showed superior performance to all methods across all current values.

The worst performing method was unsurprisingly MPEM, since it does not take into account the saturation of the electromagnets, which is significant in the case of the CardioMag. Introducing a simple saturation model in the S-MPEM significantly improved the prediction performance. One should note that the saturation model of the S-MPEM does not take into account the interactions between the electromagnets, since it is a scalar function of the electromagnet current. Nevertheless, the performance gain is significant compared to the MPEM and is close to that of the previously published ANN. The performance even exceeds the ANN at lower current values, while it is slightly worse at higher current ranges, where the interactions between electromagnets are most significant.

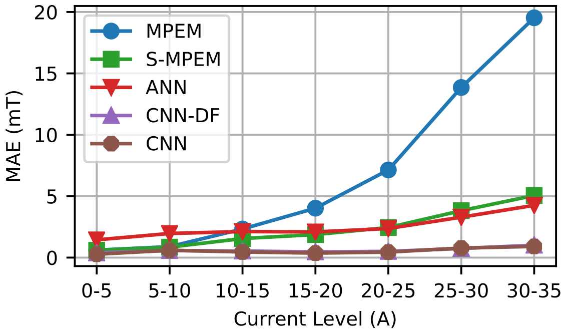

Fig. 9. MAE depending on the maximum absolute current across all electromagnets.

The ANN already improved the prediction performance by a factor of 3.3x over the linear MPEM when comparing the average MAE. In comparison, the newly introduced CNNs perform significantly better than the current state-of-the-art, with factor of improvement up to 5.4x over the ANN. This is testament to the power of the large neural architectures of "deep learning" for modeling complex physical phenomena including magnetic fields. NB. CNN-DF performed slightly worse than CNN, again suggesting that physical constraints do not improve prediction performance, given uncertainty in the dataset, due to variations between magnetic sensors, measurement noise, positional errors, or time-dependent variations. Nevertheless, it remains a method that combines performance nearly matching the CNN, with the guarantee that predicted fields are divergence-free.

D. Computing Gradients

Gradients in the MPEM and S-MPEM have analytical expressions. For the ANN, gradients can be computed using automatic differentiation, which is available in most neural-network programming frameworks since it is inherent to the backpropagation algorithm, which optimizes the parameters of the network. For the CNN, positions do not appear as quantities in the regression problem, and gradients can not be obtained directly. One can however resort to an interpolation method such as RBF-G-3D to convert the discretized field to a continuous function of position, and compute the gradients as discussed in III-C5. In contrast to III-C5, proper evaluation of gradient predictions is not possible since local gradient measurements were not available, and estimations using numeric differentiation of adjacent magnetic field measurements would be too inaccurate.

E. Prediction Speed

It is difficult to perform a rigorous and fair benchmarking of the CNN, ANN, and MPEM models, since they use different computational paradigms and run on different hardware. The CNN models used in this work contain 1.36 million floating point parameters, which is significantly higher than the number of parameters in MPEM, where the number is in the hundreds. Nevertheless CNNs can be trained and queried rapidly on GPU hardware. Our tests showed that an entire $N_g = 16$ field map could be computed in $4.37\,\mathrm{ms}$ on average. For comparison, a single computation of the MPEM at a single position took $6.85\,\mathrm{ms}$ on average, and $665\,\mu\mathrm{s}$ for the ANN method. For applications where having the whole magnetic field map is useful, the computational benefit of CNNs is clear. Moreover, for applications where the field must be computed at a single position, there does not appear to be a significant performance penalty.

V. Conclusion

Accurate modeling of the magnetic fields generated by MNSs is a fundamental task for magnetic navigation in robotics. We estimate that a field prediction accuracy on the order of $0.3\,\mathrm{mT}$ would be required for mm position tracking of devices inside a clinical-scale system like the CardioMag. Several methods were compared for generating continuous expressions of magnetic fields from measurements using interpolation. For eMNSs with several large electromagnets, as can be expected in clinical applications, the nonlinear magnetization of the electromagnets and interactions between electromagnets render the magnetic field prediction task challenging. In our previous attempts [24], we introduced the use of ANNs for nonlinear prediction of the magnetic fields generated by a large-scale eMNS. In this work, we introduced two new methods for predicting these magnetic fields. The S-MPEM is a simple addition to current linear models while exhibiting similar performance to the ANNs. We reported the first use of deep-learning methods for modeling eMNSs with significant improvement in accuracy over the ANNs. Using a CNN, we achieved a MAE of $0.5\,\mathrm{mT}$, which is on the order of the aforementioned $0.3\,\mathrm{mT}$ requirement for position tracking.

Appendix
Inverting eMNS Models

Backward models relate a desired magnetic field, magnetic field gradient, or combination thereof to a controlled parameter, and are therefore crucial in robotics applications. In the case of a linear magnetic model, there exists is a closed form inverse map to (3). If there are more than three electromagnets and (3) is underdetermined, one may express the solution that minimizes the power consumption of the eMNS by solving

$$\begin{aligned} \min_{\mathbf{i}} \quad & \mathbf{i}^T\mathbf{i} \\ \text{s.t.} \quad & \mathbf{A}_m\ \mathbf{i} = \mathbf{b}_d \end{aligned} \tag{34}$$

where $\mathbf{b}_d \in \mathbb{R}^3$ is a desired magnetic field value at position $\mathbf{p}$ and $\mathbf{i} \in \mathbb{R}^{N_e}$ is the vector electromagnet currents. The solution $\mathbf{i}_d$ is

$$\mathbf{i}_d = \mathbf{A}_m^\dagger\ \mathbf{b}_d \tag{35}$$

where $\mathbf{A}_m^\dagger$ denotes the Moore-Penrose pseudoinverse of $\mathbf{A}_m$. For the saturated linear models such as the S-MPEM introduced in IV-B2, the backward map is straightforward, provided that an inverse saturation function $h_k^{-1}(y)$ exists, as is the case for most sigmoid functions. The inverse of (30) is

$$\tilde{\mathbf{i}} = \mathbf{A}_m^\dagger\ \mathbf{b}_d \tag{36}$$

$$\mathbf{i}_d = \begin{bmatrix} {h_1}^{-1}(\tilde{i}_1) & \cdots & {h_{N_e}}^{-1}(\tilde{i}_{N_e}) \end{bmatrix}^T \tag{37}$$

Such expressions are convenient, since they can be computed in closed-form independent of the type of linear model. Note that if the electromagnets all have the same electrical resistance and the same saturation, this also minimizes the power consumption of the eMNS. For the more general nonlinear magnetic model and for other types of constraints, the backward model generally does not have a closed-form expression. In such cases one may pose the inversion problem as a constrained nonlinear optimization

$$\begin{aligned} \min_{\mathbf{i}} \quad & \|g(\mathbf{p},\mathbf{i}) - \mathbf{b}_d\|^2 \\ \text{s.t.} \quad & h(\mathbf{i}) = 0 \\ & f(\mathbf{i}) \leq 0 \end{aligned} \tag{38}$$

where $h(\mathbf{i})$ and $f(\mathbf{i})$ represent equality and inequality constraints on the current, respectively. Since the generative CNNs predict a discretized map of the magnetic field (see IV-B4), an additional interpolation step may be required to invert desired fields at arbitrary positions, but the inversion problem can also be solved using iterative optimization.

Solving for a set of currents that achieves a desired magnetic field while minimizing the power consumption of the eMNS. This can be performed with an unconstrained nonlinear solver such as Ceres [36] using the following L_2 regularized minimization

$$\min_{\mathbf{i}} \ \|g(\mathbf{p},\mathbf{i}) - \mathbf{b}_d\|^2 + \lambda\ \mathbf{i}^T\mathbf{i} \tag{39}$$

where $\lambda \in \mathbb{R}$ is a small number.

Acknowledgment

We would like to thank NVIDIA for providing us with a Titan Xp through the GPU grant program. This work was also supported by the Swiss National Science Foundation through grant numbers 200021_165564 and 200021_168997, and the ERC Advanced Grant 743217 Soft Micro Robotics (SOMBOT). This work has been partially funded by BRIDGE through grant number 180861.

References

[1] J. J. Abbott, E. Diller, and A. J. Petruska, "Magnetic Methods in Robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, 2020.

[2] Stereotaxis Inc., "One Hundred Thousand Procedure Milestone Achieved by Stereotaxis Robotic Technology," Stereotaxis Inc., St. Louis, Tech. Rep., 2017.

[3] J. Li *et al.*, "Micro/nanorobots for biomedicine: Delivery, surgery, sensing, and detoxification," 2017.

[4] S. Ernst *et al.*, "Initial Experience with Remote Catheter Ablation Using a Novel Magnetic Navigation System: Magnetic Remote Catheter Ablation," *Circulation*, vol. 109, no. 12, pp. 1472–1475, 2004.

[5] P. R. Slawinski *et al.*, "Autonomous Retroflexion of a Magnetic Flexible Endoscope," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1352–1359, 2017.

[6] A. Hong *et al.*, "3D path planning for flexible needle steering in neurosurgery," *International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 15, no. 4, 2019.

[7] C. Chautems and B. J. Nelson, "The tethered magnet: Force and 5-DOF pose control for cardiac ablation," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4837–4842, 2017.

[8] C. Di Natali *et al.*, "Jacobian-Based Iterative Method for Magnetic Localization in Robotic Capsule Endoscopy," *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 327–338, 2016.

[9] K. M. Popek, T. Schmid, and J. J. Abbott, "Six-Degree-of-Freedom Localization of an Untethered Magnetic Capsule Using a Single Rotating Magnetic Dipole," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 305–312, 2017.

[10] J. Edelmann, A. J. Petruska, and B. J. Nelson, "Magnetic control of continuum devices," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 68–85, 2017.

[11] M. S. Grady *et al.*, "Preliminary experimental investigation of in vivo magnetic manipulation: Results and potential application in hyperthermia," *Medical Physics*, vol. 16, no. 2, pp. 263–272, 1989.

[12] A. Caciagli *et al.*, "Exact expression for the magnetic field of a finite cylinder with arbitrary uniform magnetization," *Journal of Magnetism and Magnetic Materials*, vol. 456, pp. 423–432, 6 2018.

[13] A. Pourkand and J. J. Abbott, "A Critical Analysis of Eight-Electromagnet Manipulation Systems: The Role of Electromagnet Configuration on Strength, Isotropy, and Access," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2957–2962, 2018.

[14] M. Etiévant *et al.*, "An improved control-oriented modeling of the magnetic field," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May. Institute of Electrical and Electronics Engineers Inc., 5 2019, pp. 6178–6184.

[15] J. P. Bastos, N. Sadowski, and N. Sadowski, *Electromagnetic Modeling by Finite Element Methods*, ser. Electrical and Computer Engineering. CRC Press, 4 2003, vol. 117.

[16] A. George, "Nested Dissection of a Regular Finite Element Mesh," *SIAM Journal of Numerical Analysis*, vol. 10, no. 2, pp. 345–363, 1973.

[17] J. Sikorski *et al.*, "Introducing BigMag - A novel system for 3D magnetic actuation of flexible surgical manipulators," in *Proceedings - IEEE International Conference on Robotics and Automation*. IEEE, 5 2017, pp. 3594–3599.

[18] M. P. Kummer *et al.*, "Octomag: An electromagnetic system for 5-DOF wireless micromanipulation," *IEEE Transactions on Robotics*, vol. 26, no. 6, pp. 1006–1017, 12 2010.

[19] F. Niu *et al.*, "Development of an Enhanced Electromagnetic Actuation System with Enlarged Workspace," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 5, pp. 2265–2276, 2017.

[20] J. Nam *et al.*, "Magnetic Navigation System Utilizing a Closed Magnetic Circuit to Maximize Magnetic Field and a Mapping Method to Precisely Control Magnetic Field in Real Time," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5673–5681, 2018.

[21] F. Ongaro, C. M. Heunis, and S. Misra, "Precise Model-Free Spline-Based Approach for Magnetic Field Mapping," *IEEE Magnetics Letters*, pp. 1–1, 2018.

[22] D. Son, X. Dong, and M. Sitti, "A Simultaneous Calibration Method for Magnetic Robot Localization and Actuation Systems," *IEEE Transactions on Robotics*, vol. PP, pp. 1–10, 2018.

[23] A. J. Petruska, J. Edelmann, and B. J. Nelson, "Model-Based Calibration for Magnetic Manipulation," *IEEE Transactions on Magnetics*, vol. 53, no. 7, pp. 1–1, 7 2017.

[24] R. Yu *et al.*, "Modeling Electromagnetic Navigation Systems for Medical Applications using Random Forests and Artificial Neural Networks," in *to appear at IEEE Int. Conf.. Rob. Autom. 2020*, Paris, 2020.

[25] F. Lekien and J. Marsden, "Tricubic interpolation in three dimensions," *International Journal for Numerical Methods in Engineering*, vol. 63, no. 3, pp. 455–471, 2005.

[26] C. De Boor, *A Practical Guide to Splines*. New York: Springer, 1978.

[27] M. D. Buhmann, *Radial Basis Functions: Theory and Implementations*. Cambridge: Cambridge University Press, 2004, vol. 73, no. 247.

[28] F. J. Narcowich and J. D. Ward, "Generalized Hermite Interpolation Via Matrix-Valued Conditionally Positive Definite Functions," *Mathematics of Computation*, vol. 63, no. 208, p. 661, 1994.

[29] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)?-Arguments against avoiding RMSE in the literature," *Geosci. Model Dev*, vol. 7, pp. 1247–1250, 2014.

[30] R. Schaback, "Multivariate Interpolation and Approximation by Translates of a Basis Function x 2 . Optimal Recovery via Discrete Sampling," *Approximation Theory VIII*, pp. 1–25, 1995.

[31] G. E. Fasshauer and J. G. Zhang, "On choosing "optimal" shape parameters for RBF approximation," *Numerical Algorithms*, vol. 45, no. 1-4, pp. 345–368, 2007.

[32] K. Toraichi *et al.*, "Computational complexity of spline interpolation," *International Journal of Systems Science*, vol. 18, no. 5, pp. 945–954, 1987.

[33] B. Kim *et al.*, "Deep Fluids: A Generative Network for Parameterized Fluid Simulations," *Computer Graphics Forum*, vol. 38, no. 2, pp. 59–70, 2019.

[34] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283.

[35] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic gradient descent," *ICLR: International Conference on Learning Representations*, 2015.

[36] S. Agarwal, K. Mierle, and Others, "Ceres Solver."

Samuel Charreyron Samuel Charreyron was born in 1989 in Fontainebleau, France. He received a B. Eng in electrical engineering in 2012 from McGill University followed by a M. Sci in electrical engineering from ETH Zurich in 2014. He joined the Multi-Scale Robotics Lab (MSRL) at ETH Zurich in 2013 as a masters student and completed his PhD under the supervision of Prof. Brad Nelson. He is currently employed by the MSRL as a post doctorate researcher, and continues to work on applications of magnetic navigation to minimally invasive surgery.

Quentin Boehler Quentin Boehler was born in Strasbourg, France in 1990. He received a M.S. degree in mechatronics from INSA Strasbourg, France in 2013, and a Ph.D. degree in robotics from ICube laboratory, University of Strasbourg, France in 2016. He joined the Multi-Scale Robotics Lab at ETH Zrich in 2017 as a postdoctoral associate under the supervision of Bradley J. Nelson. He is currently a senior scientist and his research interests include magnetic actuation for medical robotics, the development and analysis of electromagnetic navigation systems, and the design and control of soft magnetic robots.

Byungsoo Kim Byungsoo Kim was born in 1987 in South Korea. He received BSc in Computer Science in 2009 from KAIST and MSc in Computer Science in 2016 from ETH Zurich. He completed his joint PhD at Computer Graphics Lab at ETH Zurich and Disney Research Studios, under the supervision of Prof. Markus Gross. His research mainly focuses on deep learning methods for art-directable fluid simulations.

Cameron Weibel Cameron was born in 1995 in San Diego, California. He received his bachelors degree in Computer Science and Engineering from the Eindhoven University of Technology in 2017. He obtained his master's degree in Robotics, Systems and Control from ETH Zurich and completed his thesis within their Multi-Scale Robotics Lab (MSRL). He is currently Manager of Artificial Intelligence for Digital Customer Service at Swiss International Air Lines and teaches robotics and programming across all ages through his company Hebbian.

Christophe Chautems Dr. Christophe Chautems studied mechanical engineering at ETH Zurich. Between September 2009 and June 2011, he was involved in a research project at Nasa Ames and at ESA, where he strengthened his knowledge in autonomous planetary rover navigation. He then spent 15 months working for the automotive industry before joining Aeon Scientific to develop an electromagnetic navigation system to steer a catheter for cardiac arrhythmia ablations. He joined the MSRL in June 2016 with the intention of shaping the future of magnetic navigation technology for applications in the medical field, and to support MSRL with his experience in designing robotic medical devices.

Bradley Nelson Brad Nelson received mechanical engineering degrees from the University of Illinois (B.S. 1984) and the University of Minnesota (M.S. 1987), and a Ph.D. in Robotics (School of Computer Science) from Carnegie Mellon University (1995). Prof. Nelson has been on the faculty of the University of Minnesota and the University of Illinois at Chicago, has worked at Motorola and Honeywell, and has served as a United States Peace Corps Volunteer in Botswana, Africa. He has been the Professor of Robotics and Intelligent Systems at the Swiss Federal Institute of Technology (ETH), Zurich since 2002.