TITLE PAGE

Title

MAGIC: a MATLAB toolbox for external control of transcranial magnetic stimulation devices

Authors

Forough Habibollahi Saatlou[1#], Nigel C. Rogasch[2*#], Nicolas A. McNair[3], Mana Biabani[2], Steven D. Pillen[4], Tom R. Marshall[5,6], Til O. Bergmann[4,7]

[#] Contributed equally

1. Department of Electrical Engineering, Sharif University of Technology, Azadi Avenue, Tehran 145888-9694, Iran
2. Brain and Mental Health Laboratory, Monash Institute of Cognitive and Clinical Neuroscience and Monash Biomedical Imaging, Monash University, VIC, Australia
3. School of Psychology, University of Sydney, Sydney, NSW 2006, Australia
4. Department of Neurology & Stroke, and Hertie Institute for Clinical Brain Research, Eberhard Karls University of Tübingen, Hoppe-Seyler-Str. 3, 72076 Tübingen, Germany
5. Wellcome Centre for Integrative Neuroimaging, University of Oxford, John Radcliffe Hospital, OX3 9DU, Oxford, United Kingdom
6. Department of Experimental Psychology, University of Oxford, Woodstock Rd, Oxford OX2 6GG
7. Institute for Medical Psychology and Behavioral Neurobiology, Eberhard Karls University of Tübingen, Otfried-Müller-Straße 25, 72076 Tübingen, Germany

*Correspondence

Nigel C. Rogasch
Monash University
Building 220, Clayton Campus
770 Blackburn Rd.
Clayton, VIC, 3800
Australia
nigel.rogasch@monash.edu

Keywords: TMS, remote control, serial port, MagVenture, MagStim

Abstract

The capacity to externally control transcranial magnetic stimulation (TMS) devices is becoming increasingly important in brain stimulation research. Here we introduce MAGIC (MAGnetic stimulator Interface Controller), an open-source MATLAB toolbox for controlling Magstim and MagVenture stimulators. MAGIC includes a series of MATLAB functions which allow the user to arm/disarm the stimulator, send triggers, change stimulator settings such as amplitude, interpulse intervals, and frequency, and receive stimulator setting information via a serial port connection between a computer and the stimulator. By providing external control capability, MAGIC enables greater flexibility in designing research protocols which require trial-by-trial changes of device settings to realize a priori trial randomization or interactive ad hoc adjustment of parameters during an ongoing experiment. MAGIC thus helps to prevent experimental confounds related to the block-wise variation of parameters and facilitates the integration of TMS with cognitive/sensory tasks, and the development of more adaptive brain state-dependent brain stimulation protocols.

Highlights

- MAGIC is an open source Matlab toolbox to externally control TMS devices
- MAGIC supports control of MagStim and MagVenture devices via a serial port
- MAGIC can set and get parameters enabled for external access by the manufacturers
- MAGIC functions can be integrated with Matlab based experimental control software
- MAGIC enables flexible designs, a priori trial randomization, and ad hoc adjustment

Introduction

Transcranial magnetic stimulation (TMS) is widely used in basic and clinical neurophysiology, cognitive and systems neuroscience, as well as for therapeutic applications in neurology and psychiatry. As long as standard protocols with fixed stimulation parameters are used, it is sufficient to pre-define those parameters manually via the control interface of the TMS device. This is usually the case for standard *offline* repetitive TMS (rTMS) protocols to induce plasticity-dependent changes in cortical excitability or simple *online* TMS applications using single pulses or bursts, e.g., when measuring motor-evoked potentials (MEP) or interfering with task-related neural activity, respectively. In these cases, TMS devices can be used in a stand-alone fashion, or in combination with devices that merely control the timing of the stimulation by sending TTL pulse triggers. However, there are two scenarios for which stimulation parameters need to be changed automatically from trial-to-trial: *a priori randomization* and *ad hoc adjustment*. Luckily, most TMS devices allow the remote control of stimulation parameters as well as the retrieval of the current device status, via bidirectional communication through a serial or USB port.

A priori randomization. As soon as multiple experimental conditions are involved that require a change of stimulation parameters, and block-wise application is not desired for specific scientific reasons, it is always preferable to intermingle those conditions in a (pseudo-)randomized fashion. Thereby, systematic experimental confounds such as expectation/predictability, position (order) effects, or sequence (carry-over) effects can be prevented. A trial sequence can be generated *a priori* and the respective parameters changed accordingly from trial-to-trial. Examples for this scenario include: (i) paired-pulse protocols for which the administration of single-pulse and paired-pulse trials should be intermingled; (ii) the systematic mapping of *input-output curves* (e.g., measuring MEP amplitudes as a function of stimulation intensity); or (iii) any kind of cognitive or motor task-related TMS for which stimulation parameters (e.g., intensity or burst frequency) vary across experimental conditions.

Ad hoc adjustment. When certain stimulation parameters cannot be defined a priori but are conditional on events of the very recent past, such as previous stimulation outcome, the participant's response, or the current brain state, they need to be adjusted *ad hoc* while the experiment is running. Examples for this scenario include: (i) fully-automated adaptive threshold hunting procedures for which the intensity of the next trial depends on the outcome of the last trial (e.g., MEP amplitudes for resting motor threshold, or perceptual reports for phosphene threshold) [1–4]; (ii) cognitive tasks for which certain stimulation parameters need to be iteratively adapted depending on past performance or certain decisions of the participant; or (iii) brain state-dependent brain stimulation approaches for which not only the timing [5,6] but also intensity, frequency or inter-stimulus intervals may be adapted to take the current brain state into account (e.g., accounting for the frequency, amplitude, and phase of a target oscillation with EEG-triggered TMS) [7,8].

MAGIC overview

The MAGIC (MAGnetic stimulator Interface Controller) toolbox offers a MATLAB-based unified framework for the simultaneous control of multiple stimulators of different manufacturers through a computer serial port. MAGIC currently supports MagVenture (R30, R100, X100, incl. MagOption for paired-pulse functionality) and Magstim (200^2, Bistim², Rapid²) stimulators, but control of further devices may be added in the future. Each function is organised to send commands to the stimulator, and receive

feedback on the current stimulator settings. The MAGIC toolbox complements existing TMS external control toolboxes, such as MagPy which is written in the Python programming language [9] or the RAPID2 Toolbox for MATLAB [1].

Availability

MAGIC is open source and available for download from GitHub (https://github.com/nigelrogasch/MAGIC/releases). A wiki page explaining the functionality of the toolbox in detail is also available (https://github.com/nigelrogasch/MAGIC/wiki). MAGIC is released under the GNU General Public Licence (v3). The brain stimulation community is invited to add additional code to the toolbox (e.g., for other brands of stimulator), improve the existing code, or report bugs. Please contact the study authors to contribute.

Dependencies

MAGIC is written in MATLAB, and therefore this software is required for operation. The toolbox has been tested with MATLAB versions 2014a and higher. MAGIC is designed to work with any MagVenture or Magstim stimulator, however some functionality may be restricted by the stimulator model, or by the stimulator software/firmware version. Details on model/software/firmware dependencies are included in the toolbox code. The communication with Magstim and MagVenture stimulators requires a serial port installed on the computer running MATLAB (note that the serial port may also emulated via an USB-to-serial adapter). Details regarding pin layout and cable configuration are provided in Figure 1.
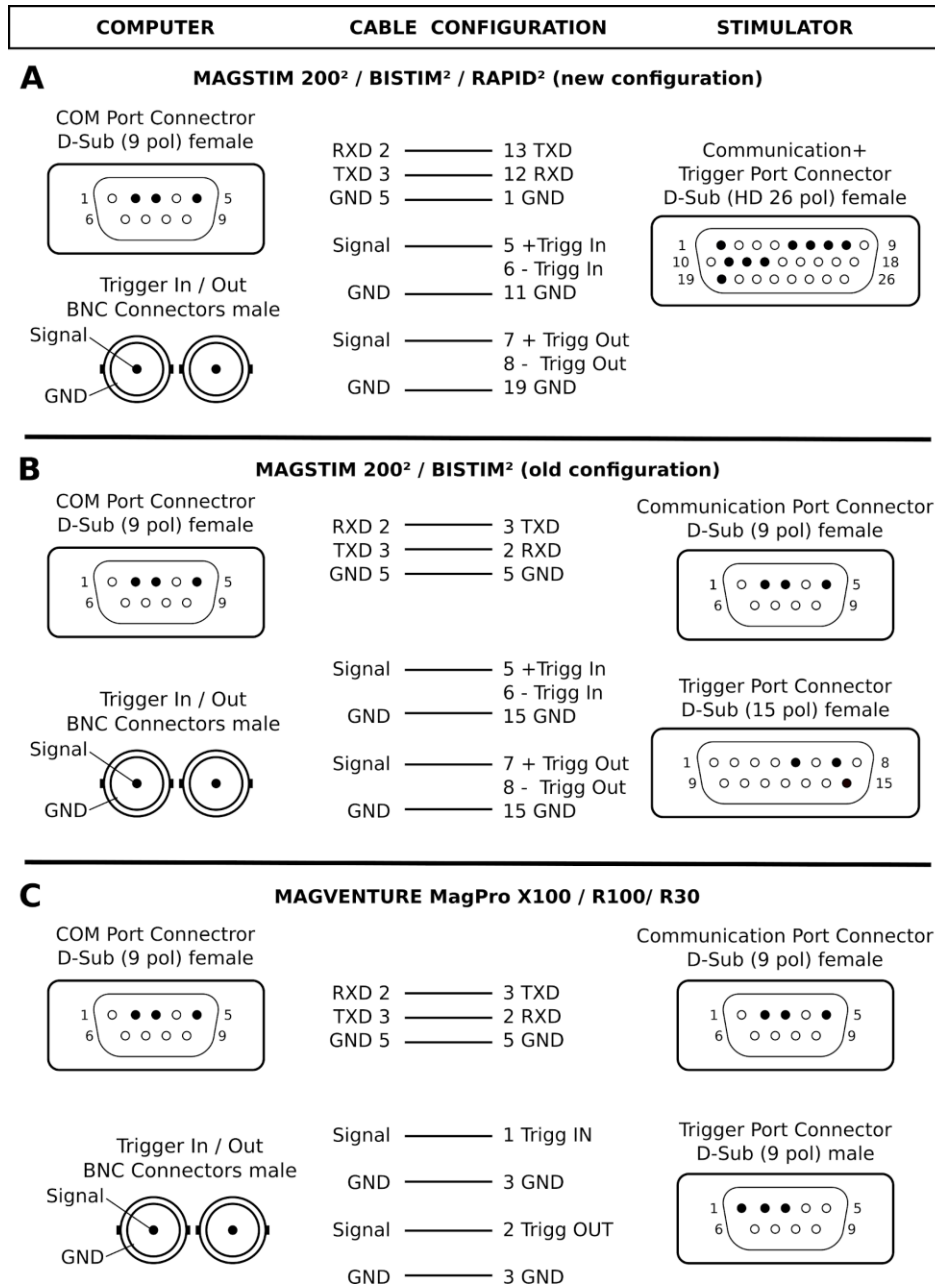
Figure 1: How to connect your computer to the stimulator. *Pin layouts are depicted for the connector to the serial port of the computer running MATLAB and BNC connectors for separate Trigger In and Out signals on the one side (left), and the connector(s) to the respective communication port and trigger port of the stimulator on the other side (right), together with the respective cable configuration (middle). Note that pin layouts refer to the connectors, not the ports. Information is provided for (**A**) the configuration for new Magstim stimulators, (**B**) old Magstim stimulators, and (**C**) MagVenture stimulators. Note that in (B) and (C) standard Null-Modem cables (RS232) are used for the connection to the communication port. For additional detail on hardware wiring please refer to the respective manufacturers' manuals. Note that all information is supplied without liability and the authors do not take responsibility for any problems resulting from incorrect wiring. In case of any doubt the manufacturer should be contacted directly.*

Using MAGIC

MAGIC consists of two parent class functions: *magventure.m* for controlling MagVenture stimulators and *magstim.m* for controlling Magstim stimulators (figure 2A,B). The magstim parent function controls basic settings for all models of Magstim stimulator (200^2, Bistim², Rapid²), however functionality specific to Bistim^2 and Rapid^2 models is controlled by child class functions: *bistim.m* and *rapid.m*.
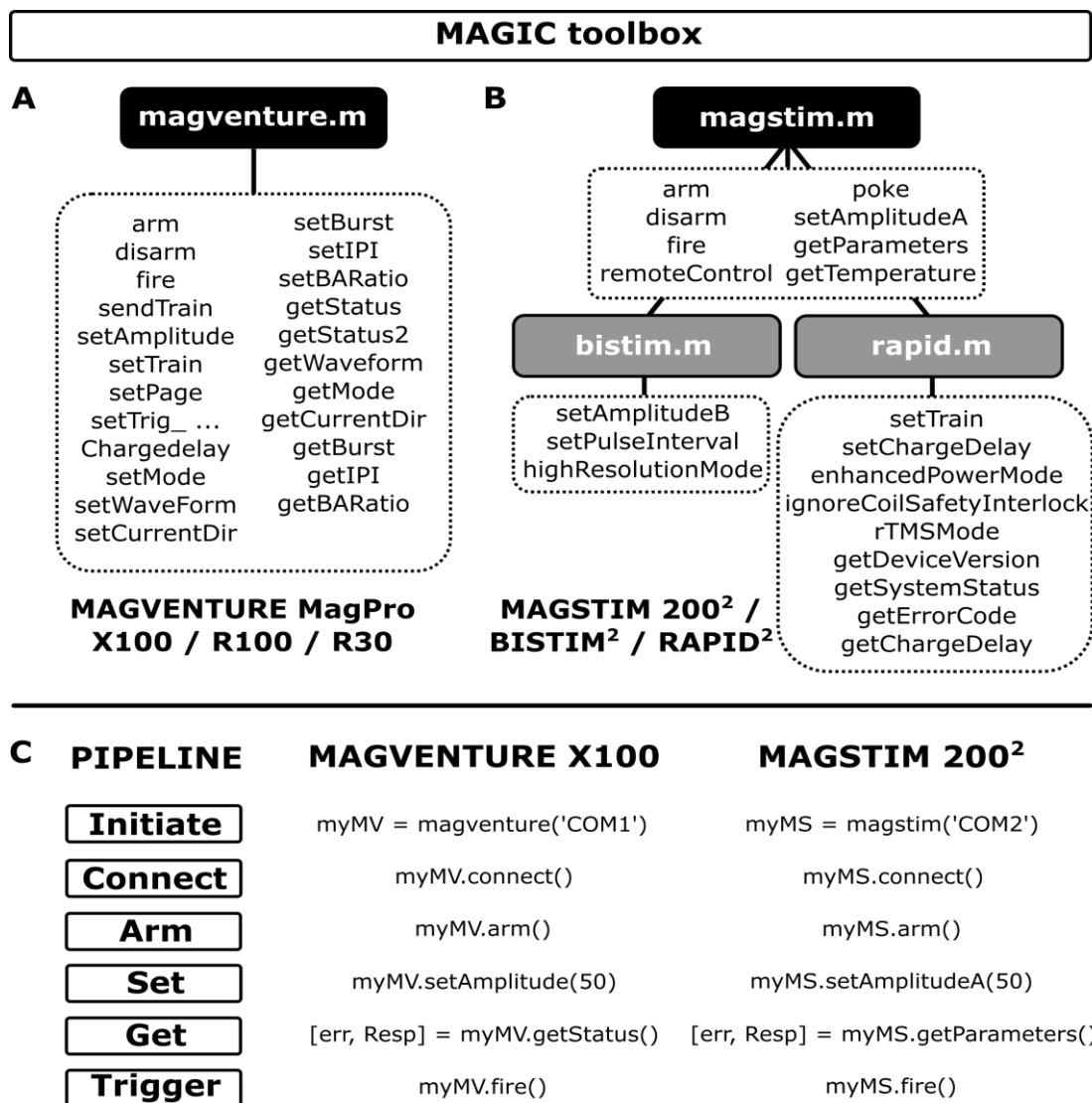
*Figure 2: MAGIC toolbox overview and examples. The MAGIC toolbox is organised into Matlab parent class functions (black boxes) which use different methods to interact with MagVenture (**A**) and Magstim (**B**) devices to alter a range of settings, and receive device information (indicated in dotted boxes). For Magstim devices, additional parameter settings required for Bistim^2 and Rapid^2 are controlled through child class functions (grey boxes), which share common settings with the 200^2 (arm, disarm etc.) through the parent function. Details of required inputs for each method are included in the MAGIC wiki (see main*

6

*text). An example of a basic stimulation pipeline is provided in (**C**) for a MagVenture X100 and Magstim 200². An object (myMV for MagVenture, myMS for Magstim in this example) is generated by calling the parent class function and identifying the serial port address in which communication will occur. Note that multiple serial ports are required to simultaneously control multiple devices. The serial port is then opened and the device armed. The amplitude of each stimulator is set to 50% of maximum stimulator output using the 'set' methods. As settings differ between stimulator brands, models and even modes within a stimulator, method names and inputs can differ slightly depending on the stimulator. In this example, the set amplitude method differs slightly between devices: for the Magstim device setAmplitudeA differentiates the test from the conditioning unit when using multiple units, such as with a Bistim² (the second unit is altered using setAmplitudeB in the bistim.m child class); for MagVenture the conditioning and test pulse amplitudes are set in the same unit by passing a vector, e.g. myMV.setAmplitude([50,60]). General device settings are returned in the Resp variable to the MATLAB workspace using the 'get' methods, which again differ slightly between devices. Finally, the devices are triggered by sending the fire method.*

Once the MAGIC toolbox is added to the MATLAB path, the user can create an object using the appropriate class constructor parent function, which identifies the stimulator to be controlled and defines the serial port address through which communication will occur (figure 2C). This design allows for simultaneous control of multiple stimulators via different serial ports. To begin sending commands to the stimulator, the serial port must first be opened using the *connect* command. Conversely, the serial port can be closed using the *disconnect* command. Note that for Rapid² stimulators with firmware versions V9.0 and above, a device-specific *unlock code* (to be obtained from the manufacturer) has to be set when initializing the stimulator object to enable remote control.

In MAGIC, commands to arm or disarm the stimulator, trigger stimulation, or adjust device settings are sent by calling the appropriate method, and providing the required input for that setting. After sending a command, feedback indicating the status of different stimulator settings can be read from the serial port and returned to the MATLAB workspace. Feedback is enabled by providing a boolean input (1 = on) after the last setting input to the object. Two variables are then returned: the first indicates whether the device successfully returned status data to the serial port (0 = success, 1+ = error); the second is either a structure with fields indicating the current status of certain settings on the device, or an error message. The status of the device, including current settings and coil temperature, can be queried independently of 'set' commands by sending 'get' commands. Again, device information is returned in a structure variable to the MATLAB workspace. See the Wiki for details on the particular field(s) returned by each method.

MAGIC permits control of most device settings unique to specific stimulator brands or models. For MagVenture devices, settings include: stimulator mode (e.g. standard, power, or twin/dual); train parameters (e.g. repetition rate); interpulse intervals in twin/dual mode; trigger and recharge delays; pulse current direction; and waveform type (e.g. monophasic, biphasic) (figure 2A). Note that only a limited number of setting commands are available for certain versions of the MagVenture stimulator firmware.

For Magstim devices, additional Bistim^2 settings include: interpulse intervals; and engaging high resolution mode (to enable submillisecond intervals). Rapid^2 settings include: stimulation frequency; number of pulses; and high power mode (figure 2B). As in the MagPy toolbox, MAGIC periodically polls Magstim devices every 500 ms in order to maintain remote control, and a *magstim.poke()* command is available to clear the serial port and prevent timing errors related to simultaneous polling and sending commands [9]. Note that Matlab functions interrupting code execution with high temporal precision, e.g., PsychToolbox's *WaitSecs()*, may also interrupt maintenance polling and abort remote control, while other, less accurate, functions, e.g., Matlab's *pause()*, do not.

Another important consideration when using MAGIC to remotely change settings is that it may take a variable delay until the change takes effect and the stimulator is again ready to be triggered with the new parameters set. This can range from a few tens of milliseconds (e.g., when changing inter-stimulus interval) to several seconds (e.g., when changing mode in the MagVenture devices). Furthermore, a delay of up to 10 ms may be present when triggering the stimulation via the serial port [9].

Summary

MAGIC provides full control over both Magstim and MagVenture TMS devices from within the MATLAB software package. The toolbox allows for the design of customised TMS protocols in which device settings change from trial-to-trial, to implement *a priori randomization* of trial types or *ad hoc adjustment* of stimulation parameters based on past events or current brain state. The capacity to control stimulator settings from MATLAB allows easier integration with MATLAB-based programs commonly used in cognitive and sensory neuroscience, such as Psychtoolbox [10]. Furthermore, MAGIC offers greater flexibility for experimental design when integrating TMS with neuroimaging modalities [8], such as concurrent TMS-EEG [11,12], EEG-guided TMS [5,7], and TMS-fMRI [13]. The external control over devices provided by MAGIC is thus extremely useful in the modern age of TMS research, as experimental designs become increasingly complex and require greater flexibility in manipulating device settings.

Competing financial interests

The authors declare no competing financial interests.

Author contributions

N.C.R., F.H.S., N.A.M, and T.O.B. designed the toolbox, F.H.S. wrote the main code; N.A.M., T.R.M., and T.O.B. contributed to earlier versions of the code; M.B., S.D.P., and N.A.M., tested, debugged, and refined the code; N.C.R. and T.O.B. wrote the manuscript; all authors jointly discussed and revised the manuscript.

Acknowledgements

References

[1] Abrahamyan A, Clifford CW, Ruzzoli M, Phillips D, Arabzadeh E, Harris JA. Accurate and rapid estimation of phosphene thresholds (REPT). PLoS One 2011;6:e22342. doi:10.1371/journal.pone.0022342 PONE-D-10-01806 [pii].

[2] Awiszus F. Fast estimation of transcranial magnetic stimulation motor threshold: is it safe? Brain Stimul 2011;4:58–9; discussion 60–3. doi:S1935-861X(10)00116-6 [pii] 10.1016/j.brs.2010.09.004.

[3] Awiszus F. TMS and threshold hunting. Suppl Clin Neurophysiol 2003;56:13–23.

[4] Borckardt JJ, Nahas Z, Koola J, George MS. Estimating resting motor thresholds in transcranial magnetic stimulation research and practice: a computer simulation evaluation of best methods. J ECT 2006;22:169–75. doi:10.1097/01.yct.0000235923.52741.72.

[5] Bergmann TO, Molle M, Schmidt MA, Lindner C, Marshall L, Born J, et al. EEG-Guided Transcranial Magnetic Stimulation Reveals Rapid Shifts in Motor Cortical Excitability during the Human Sleep Slow Oscillation. J Neurosci 2012;32:243–53. doi:32/1/243 [pii] 10.1523/JNEUROSCI.4792-11.2012.

[6] Zrenner C, Desideri D, Belardinelli P, Ziemann U. Real-time EEG-defined excitability states determine efficacy of TMS-induced plasticity in human motor cortex. Brain Stimul 2018;11:374–89. doi:10.1016/j.brs.2017.11.016.

[7] Thut G, Bergmann TO, Fröhlich F, Soekadar SR, Brittain J-S, Valero-Cabré A, et al. Guiding transcranial brain stimulation by EEG/MEG to interact with ongoing brain activity and associated functions: A position paper. Clin Neurophysiol 2017;128:843–57. doi:10.1016/j.clinph.2017.01.003.

[8] Bergmann TO, Karabanov A, Hartwigsen G, Thielscher A, Siebner HR. Combining non-invasive transcranial brain stimulation with neuroimaging and electrophysiology: Current approaches and future perspectives. Neuroimage 2016;140:4–19. doi:10.1016/j.neuroimage.2016.02.012.

[9] McNair NA. MagPy: A Python toolbox for controlling Magstim transcranial magnetic stimulators. J Neurosci Methods 2017;276:33–7. doi:10.1016/j.jneumeth.2016.11.006.

[10] Brainard DH. The Psychophysics Toolbox. Spat Vis 1997;10:433–6.

[11] Rogasch NC, Fitzgerald PB. Assessing cortical network properties using TMS-EEG. Hum Brain Mapp 2013;34:1652–69. doi:10.1002/hbm.22016.

[12] Herring JD, Thut G, Jensen O, Bergmann TO. Attention Modulates TMS-Locked Alpha Oscillations

in the Visual Cortex. J Neurosci 2015;35:14435–47. doi:10.1523/JNEUROSCI.1833-15.2015.

[13] Ruff CC, Driver J, Bestmann S. Combining TMS and fMRI: from "virtual lesions" to functional-network accounts of cognition. Cortex 2009;45:1043–9. doi:10.1016/j.cortex.2008.10.012.