



Extended Dynamical Causal Modelling for Phase Coupling (eDCM PC)

**SOFTWARE
METAPAPER**

AZAMAT YELDES BAY

SILVIA DAUN

**Author affiliations can be found in the back matter of this article*

ju[ubiquity press]

ABSTRACT

We present a software tool: extended Dynamic Causal Modelling for Phase Coupling (eDCM PC). This framework is implemented in MATLAB as an extension of the *Dynamic Causal Models for phase coupling* method. eDCM PC is able to estimate effective connectivity between oscillating systems using the phase information obtained from experimental signals, such as oscillatory activities of neuronal populations in distant brain regions. With the help of a transformation function, eDCM PC can measure observable independent coupling functions within and between different frequency bands. This software is available on GitLab under the GNU General Public License (version 3 or later).

CORRESPONDING AUTHOR:

Azamat Yeldesbay

Research Centre Jülich,
Institute of Neuroscience
and Medicine, Cognitive
Neuroscience (INM-3), 52425
Jülich, Germany; Institute of
Zoology, University of Cologne,
Cologne, Germany

a.yeldesbay@fz-juelich.de

KEYWORDS:

phase oscillators; coupling
functions; phase reduction;
electroencephalography (EEG);
magnetoencephalography
(MEG); oscillatory signals;
MATLAB; Dynamic Causal
Modelling (DCM)

TO CITE THIS ARTICLE:

Yeldesbay A, Daun S. 2025
Extended Dynamical Causal
Modelling for Phase Coupling
(eDCM PC). *Journal of Open
Research Software*, 13: 33.
DOI: [https://doi.org/10.5334/
jors.412](https://doi.org/10.5334/jors.412)

(1) OVERVIEW

INTRODUCTION

An oscillatory process is a widespread phenomenon in nature that occurs in physical and biological systems [2–4]. Oscillatory and rhythmic activities play a prominent role in the interaction between biological systems, especially in the communication between brain areas [5]. Therefore it is of particular interest to understand how the interaction between the elements of these oscillating systems, such as coupled physical oscillators, or between oscillatory activities in distant brain regions, occurs. In this context phase reduction is a useful method to analyze a large oscillating network by representing every oscillating system with one variable – the phase.

In the theory of synchronization the interaction between oscillating systems is analyzed by the model of weakly coupled phase oscillators [3, 6, 7]:

$$\dot{\varphi}_i = \omega_i + Q_i(\varphi_1, \dots, \varphi_N), \quad i = 1, \dots, N, \quad (1)$$

where φ_i is the phase of an oscillating system (an oscillator) i , ω_i is the natural frequency of the oscillator i , Q_i is the interaction function (coupling function) with other oscillators. For weak coupling the coupling function Q_i in Eq. (1) can be simplified as the sum of the pairwise

coupling functions q_{ij} :

$$\dot{\varphi}_i = \omega_i + \sum_{j=1, j \neq i}^N q_{ij}(\varphi_i, \varphi_j). \quad (2)$$

The pairwise coupling functions q_{ij} , on the other hand, can be represented as 2D surfaces (Figure 1).

One can obtain the coupling between the phase oscillators in the model Eq. (2) by finding the coupling functions q_{ij} directly from the experimental data [8]. This method was actively developed in the past decades and used to find the directionality of the couplings, the causal relations and to build dynamical models [9–18].

Krilemann and colleagues [9, 10] have shown that the phase extracted from the experimental signals (e.g. by using Hilbert or Wavelet transformation) behaves differently than the phase described in the theoretical model Eq. (1). The difference appears if the limit cycle of the oscillating system, the signal which we are measuring, is not circular, and the oscillatory signal has a non-sinusoidal form. This leads to a non-linear growth of the measured phase, even if the oscillating system has no input, as demonstrated in Figure 2. Moreover, it was shown in [10] that this non-linear growth of the measured phase can cause spurious couplings in the system under investigation. The problem was resolved

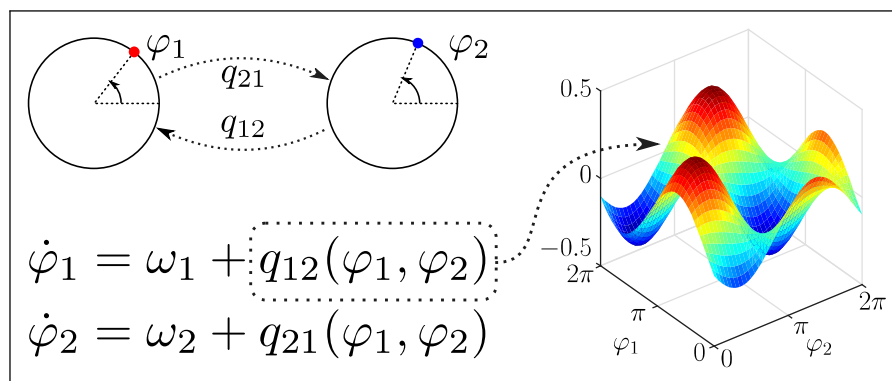


Figure 1 The model of two weakly coupled phase oscillators. A coupling function can be represented as a surface.

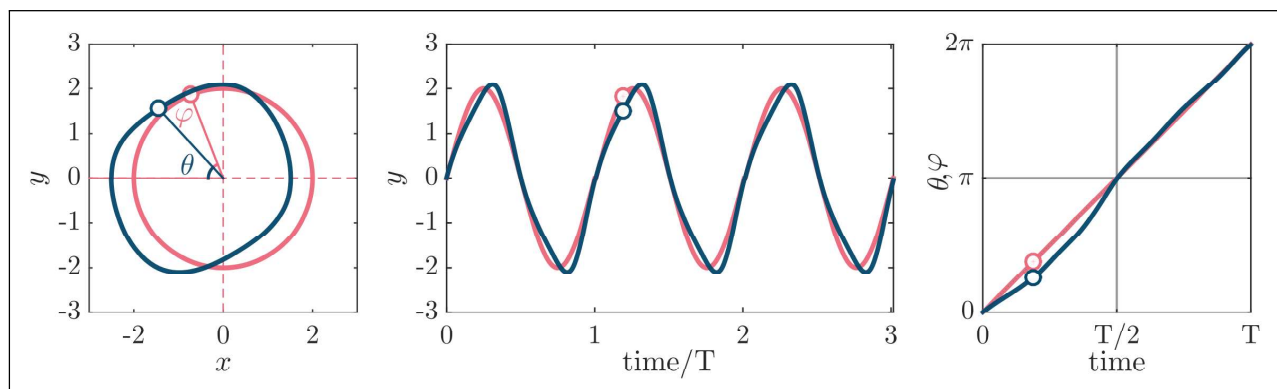


Figure 2 The difference between the theoretical and the observable (measured) phase when the limit cycle of the oscillating system is not circular (left panel). The signal of the theoretical phase is a sine wave, whereas the one of the observable phase is not (center panel). The observable phase grows non-linearly (right panel).

by introducing a transformation function between the observable and the theoretical phases (referred to as the proto- and the true phases in [10]). Kraleman and colleagues implemented this approach into their DAMOCO toolbox¹ by approximating the transformation function from the observable phase to the theoretical phase using the distribution of the observable phase over a long period of time.

The software tool presented in this metapaper – extended Dynamical Causal Modelling for Phase Coupling (eDCM PC) – addresses the problem of a potential non-linear growth in the observable phase. It is implemented as an extension to Dynamical Causal Modelling for phase coupling (DCM PC) [1]. eDCM PC aims at finding the effective coupling in a network of oscillating systems by inferring the coupling function from the measured signals. In particular, the software tool is designed to reconstruct the coupling functions for the cases when the phase is not uniformly distributed, i.e. in the case when the transformation between the observable and the theoretical phases should be taken into account. Moreover, eDCM PC extends DCM PC by allowing to find the coupling between different frequency bands, thereby making it possible to analyze n:m synchronization cases. In contrast to the DAMOCO toolbox, eDCM PC uncovers the transformation functions from the theoretical to the observable phases together with the coupling functions. It uses the capability of DCM, based on Bayesian inference, to analyze and compare several possible network structures at the same time.

The theoretical results related to eDCM PC and the numerical testing on synthetic data sets were presented in our previous work [19]. Since its first presentation in [19] eDCM PC was supplemented with a documentation, additional tests and plotting functions for data sets and results, and was rearranged in a user-friendly structure. In this work we give a detailed description on the architecture and usage of eDCM PC, as well as provide testing examples.

IMPLEMENTATION AND ARCHITECTURE

The architecture of eDCM PC is presented in Figure 3 as a data flow with different interface levels: data preprocessing level, user-interface level, eDCM PC level, and SPM/DCM level. We describe the levels in bottom up direction, namely starting from the SPM/DCM level up to the eDCM PC level by indicating specific implementation details. Thereafter, in the usage part, we describe the user-interface level.

Dynamic causal modelling

The structure of the eDCM PC is imposed by the architecture of the Dynamic Causal Modelling (DCM) [20]. DCM is an opensource toolbox within the Statistical Parametric Mapping (SPM) software [21], developed to analyze connectivity in the brain. DCM can work

with different modalities (functional magnetic resonance imaging (fMRI), EEG, MEG, local field potentials (LFP), functional near-infrared spectroscopy (fNIRS)) [20, 22–30]. The usage of a wide variety of modalities is provided by a particular feature of the DCM architecture that allows the modification of its different parts without changing the common, basic structure of the software [30].

DCM uses Bayesian inference to find the parameters of the system and the coupling between the brain regions. The elements of the common structure of DCM are the data preprocessing component (Figure 3a), the modelling component (Figure 3f), and the statistical component (Figure 3i). These elements can be modified with respect to the modality and the modelling, however the interaction between them remains the same for all versions of DCM.

Initial raw data are transformed into time courses of observables, e.g. into observable phases $\hat{\theta}$ (Figure 3a). In the modelling component (Figure 3f) synthetic signals are generated by numerically integrating the model with an initial guess for the system parameters (priors). The modelling component consists of the evolution equations (Figure 3g), a system of ODEs that represents the hidden state of the system and the connections between parts of the system, and the observation equations (Figure 3h), functions that express the observables using the hidden states. In the statistical component of DCM (Figure 3i) the observables $\hat{\theta}$ obtained from the data and the observables θ from the synthetic signals are “compared” by means of the Bayesian model inversion, namely Variational Laplace (VL) (Figure 3j). In a simplified manner, the result of this comparison gives the correction for the system parameters. Using these corrected parameter values the modelling component generates new synthetic signals. This procedure repeats until the required convergence between the measured and the synthetically generated observables is reached (Figure 3k).

Modelling component

eDCM PC introduces a new modelling component. As mentioned before the modelling component contains the evolution equation and the observation equation. The extended evolution equation is defined as

$$\dot{\varphi}_i = \omega_i + \sum_{j=1, j \neq i}^{N_q} q_{ij}(\varphi_i, \varphi_j), \quad (3)$$

where the pairwise coupling functions are constructed in the way to detect 'n : m' synchronization

$$q_{ij}(\varphi_i, \varphi_j) = \sum_{n, m = -N_q, n, m \neq 0}^{N_q} Q_{ij} e^{i(n\varphi_i + m\varphi_j)}, \quad (4)$$

with complex Fourier coefficients Q_{ij} truncated at a given number of terms N_q . In the code, however, the coupling

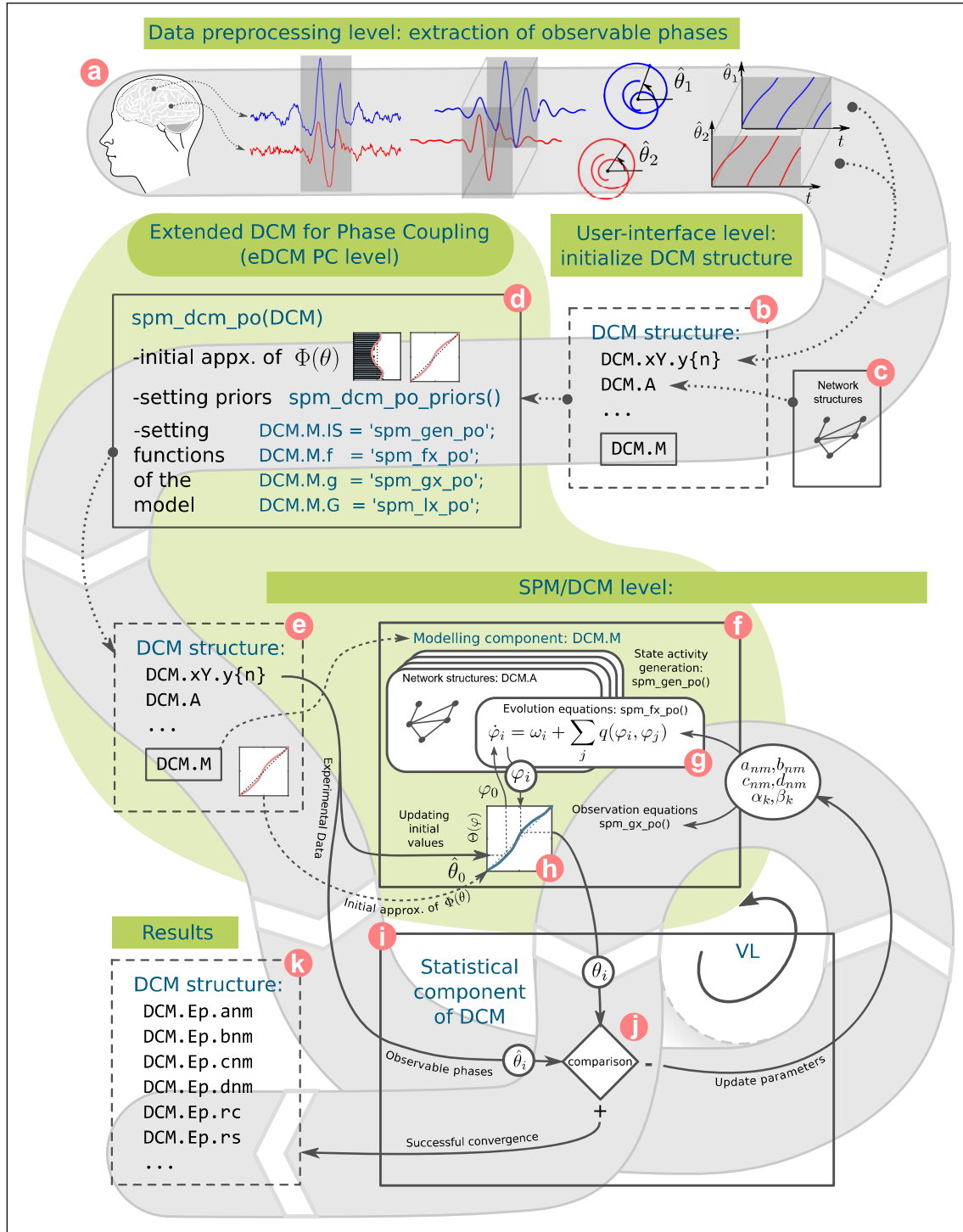


Figure 3 The data flow and the architecture scheme of eDCM PC. The green background area denotes the elements introduced in eDCM PC.

functions q_{ij} are represented with real-valued Fourier coefficients as

$$\begin{aligned}
 q_{ij}(\varphi_i, \varphi_j) = & \sum_{n=1}^{N_q} \sum_{m=1}^{N_q} \left[a_{ij}^{(n,m)} \cos(n\varphi_i) \cos(m\varphi_j) \right. \\
 & + b_{ij}^{(n,m)} \cos(n\varphi_i) \sin(m\varphi_j) \\
 & + c_{ij}^{(n,m)} \sin(n\varphi_i) \cos(m\varphi_j) \\
 & \left. + d_{ij}^{(n,m)} \sin(n\varphi_i) \sin(m\varphi_j) \right].
 \end{aligned} \quad (5)$$

To introduce the transformation function between the theoretical and observable phases (the forward transformation) we use the observation equation. Following [10] we define the transformation as follows

$$\theta_i = \Theta(\varphi_i) = \varphi_i + \sum_{k=-N_p, k \neq 0}^{N_p} (e^{ik\varphi_i} - 1), \quad (6)$$

also truncated at a given number of terms N_p . The forward transformation function is represented in the

code with real-valued coefficients:

$$\theta_i = \Theta(\varphi_i) = \varphi_i + \sum_{k=1}^{N_\rho} \frac{1}{k} \left[\alpha_i^{(k)} \sin(n\varphi_i) - \beta_i^{(k)} \cos(n\varphi_i) + \beta_i^{(k)} \right]. \quad (7)$$

Thus, in eDCM PC the hidden states are the theoretical phases φ_i and the observables are the observable phases θ_i defined for every region and every frequency band.

The resulting reconstructed coupling and transformation functions are (1) the real-valued coefficients (matrices) $a_{ij}^{(n,m)}$, $b_{ij}^{(n,m)}$, $c_{ij}^{(n,m)}$, $d_{ij}^{(n,m)}$ defined for every connected pair of regions and (2) the real-valued coefficients (vectors) $\alpha_i^{(k)}$ and $\beta_i^{(k)}$ defined for every region, respectively.

Initial approximation of the observation equation

Good convergence is provided by a good initial guess of the parameters of the system. Therefore, the initial approximation of the transformation function, i.e. the observation equation, is essential for a successful reconstruction of the system parameters and the coupling functions. eDCM PC uses the approach presented in [10] to approximate the inverse transformation function $\varphi = \Phi(\theta)$, as shown in Figure 4a–c.

The average distribution of the observable phases $\hat{\theta}_i$ over a long period of time for a region i can be approximated by a 2π periodic function $\sigma_i(\theta)$ with the mean equal to 1. (Figure 4a,b), and be written as the following Fourier series [10]

$$\sigma_i(\theta) = 1 + \sum_{k=1}^{N_\sigma} \hat{\alpha}_i^{(k)} \cos(k\varphi_i) + \hat{\beta}_i^{(k)} \sin(k\varphi_i), \quad (8)$$

where the sum is truncated at N_σ . The integral of $\sigma_i(\theta)$ over one period gives us the inverse transformation function

$$\Phi_i(\theta) = \theta_i + \sum_{k=1}^{N_\rho} \frac{1}{k} \left[\hat{\alpha}_i^{(k)} \sin(k\theta_i) - \hat{\beta}_i^{(k)} \cos(k\theta_i) + \hat{\beta}_i^{(k)} \right], \quad (9)$$

which is the inverse function of the forward transformation function $\Theta(\varphi)$ defined in Eq. (6) (Figure 4c). Note, that the truncation orders N_σ and N_ρ are, in general, different.

After obtaining the coefficients $\hat{\alpha}$ and $\hat{\beta}$ by approximating the average distribution of the observable phases $\sigma_i(\theta)$, eDCM PC finds the initial approximation of the coefficients α and β of the forward transformation $\Theta(\varphi)$.

Updating the initial conditions

In every step of the Bayesian estimation the updated parameters α and β change the transformation function $\Theta(\varphi)$. Therefore the initial conditions $\hat{\theta}_0$ of the observations should also be recalculated in every step to obtain new initial conditions for the evolution equation φ_0 (Figure 4d). For this the updated inverse

transformation function $\Phi(\theta)$ is needed. Therefore, eDCM PC recalculates the coefficients $\hat{\alpha}$ and $\hat{\beta}$ of Eq. (9) using the updated coefficients α and β after every step of the Bayesian estimation.

Usage

The data flow of the eDCM PC is presented in Figure 3. The eDCM PC package works with time courses of the observable phases obtained from oscillatory signals. The observable phase can be extracted using the Hilbert or Wavelet transformation applied to the raw signals filtered within a specific frequency band (Figure 3a).

In the user-interface level the usage of eDCM PC comes down to the initialization of the elements of the DCM structure (Figure 3b) and calling the `modcomp.spm_dcm_po(DCM)` function (Figure 3d).

The code of eDCM PC contains an example script `test.m` that runs the reconstruction procedure using a pre-simulated data set. For details of the DCM structure and calling the `modcomp.spm_dcm_po(DCM)` function we refer to the documentation included in the repository of eDCM PC. Here, we discuss only essential parts of the DCM structure.

The observable phases $\hat{\theta}$ should be stored in the DCM structure in the fields `DCM.xY.yn`, where n is the trial index (Figure 3b). All trials should have the same length with an equally distributed time step.

The supposed network structure of the system is defined in the binary adjacency matrix `DCM.A` (Figure 3c). DCM allows the definition of several possible network structures and finds the structure best fitting the data using Bayesian model comparison. In that case `DCM.A` is an array of matrix cells, and different structures should be defined in different cells. For details of the definition of the structure, please refer to the documentation of DCM in Chapters 41,42, and 43 of [21].

The elements of the modelling component are defined in the substructure `DCM.M`, such as the mean frequency ω_i (`DCM.M.freq`) and the frequency band (`DCM.M.fb`) for every region, the truncation orders of the Fourier series for the coupling functions N_q (`DCM.M.Nq`), for the forward transformation functions N_ρ (`DCM.M.Nrho`), and for the inverse transformation functions N_σ (`DCM.M.Nsig`).

The initialized DCM structure is used as an argument to call the `modcomp.spm_dcm_po(DCM)` function (Figure 3d). This function performs the initial approximation of the inverse transformation function $\Phi(\theta)$, defines the priors for all parameters of the system (`modcomp.spm_dcm_po_priors`), and assigns the functions of the model in the substructure `DCM.M` such as:

- state activities generation function `modcomp.spm_gen_po`;
- evolution equation function `modcomp.spm_fx_po`;
- observable equation function `modcomp.spm_gx_po`;

- linear observation function `modcomp.spm_lx_po`, which is a dummy function for compatibility with other versions of DCM. The function returns the values without change.

Thereafter, `modcomp.spm_dcm_po(DCM)` calls DCM routines with the DCM structure as the input argument to start the Bayesian estimation of the parameters of the system (Figure 3e).

The result of the `modcomp.spm_dcm_po(DCM)` function is an extended DCM structure that contains the estimated values of the system parameters in the substructure `DCM.Ep` (Figure 3k), namely the matrices of the Fourier coefficients of the coupling functions $a_{ij}^{(n,m)}$, $b_{ij}^{(n,m)}$, $c_{ij}^{(n,m)}$, $d_{ij}^{(n,m)}$ in `DCM.Ep.anm`, `DCM.Ep.bnm`, `DCM.Ep.cnm`, `DCM.Ep.dnm` respectively, of the forward transformations in `DCM.Ep.rc` and `DCM.Ep.rs`, which correspond to $\alpha_i^{(k)}$ and $\beta_i^{(k)}$ respectively.

Installing and testing

eDCM PC is written in MATLAB and uses Dynamic Causal Modelling (DCM), which is included in the Statistical Parametric Mapping (SPM) package [31]. The SPM package can be downloaded here <https://www.fil.ion.ucl.ac.uk/spm/software/download/>. The minimum version of SPM used by eDCM PC is SPM12. The detailed SPM installation information is provided here <https://www.fil.ion.ucl.ac.uk/spm/docs/installation/>.

eDCM PC can either be downloaded or cloned from our GitLab repository <https://gitlab.com/azayeld/edcmpe>. Add the directory containing eDCM PC package into MATLAB using `Set Path`. Use `Add Folder`, but not `Add with Subfolders...`. At restarting MATLAB will run `startup.m` script that allow to work with eDCM PC as a MATLAB-package. In the command line the message: “eDCM PC package ist initialized.” will occur as confirmation.

For installation details follow the instruction given in the documentation of the eDCM PC package <https://gitlab.com/azayeld/edcmpe/-/blob/master/Docs/Doc.md#installation>.

After installation, eDCM PC can be tested using the `test.m` script. Go to the folder, where eDCM PC is copied

or cloned, and run the testing script by running `test` in the command line or if eDCM PC folder is just added to MATLAB run:

```
startup
test
```

QUALITY CONTROL

The users can easily test whether the installation was correct by calling a `test.m` script.

The package includes three other example scripts, which are also intended to test the quality of the reconstruction.

In the first example, eDCM PC successfully reconstructs the coupling and the transformation functions from the signals of the synthetically simulated system of two weakly coupled phase oscillators. In this example the parameters of the coupling are chosen such that one coupling is zero, i.e. making the coupling unidirectional. Moreover, an artificial distortion (transformation) is added to the theoretical phases in order to simulate the effect of non-linear growing of the observable phase. Thus, the parameter values reconstructed by eDCM PC can be compared with the original parameter values used by the simulation.

The second example analyzes two uni-directionally coupled neural mass models, namely, Jansen and Rit models, which simulate EEG-like signals. In this case neither the coupling function nor the non-linear relation between the observable and the theoretical phases are known, which simulates the real case scenario. In this example, eDCM PC demonstrates successful reconstruction of the coupling functions and shows the presence of a non-linear relation between the observable and theoretical phases in the neural mass models.

The third example analyses the openly accessible Steady-State Visual Evoked Potential (SSVEP) dataset from [32], specifically examining connectivity in single-frequency SSVEPs. The provided scripts download the necessary files, extract the time courses corresponding to a given stimulation frequency and prepare the data set in a form suitable for working with the

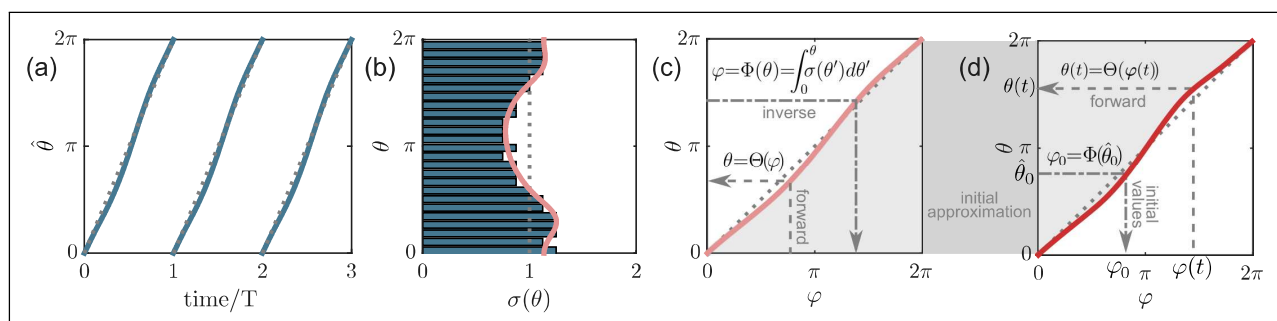


Figure 4 Initial approximation of the inverse transformation function and updating the initial conditions.

eDCM PC. The results of this example demonstrate the presence of non-linear coupling between periodic visual stimulation and the EEG signal from the occipital cortex. This example serves as a minimal script demonstrating the analysis pipeline for a dataset using eDCM PC.

The eDCM PC package includes several routines which allow testing of the data set before running the reconstruction procedure and examining the quality of the reconstruction afterwards:

1. Routines that allow the analysis of the spectral relation between two regions by plotting an approximated 2D Fourier spectrum of the observable phases. These routines can be used to analyze the data before the reconstruction procedure to set the truncation orders of the Fourier series (`DCM.M.Nq` and `DCM.M.Nrho`).
2. Routines to plot the distribution of the observable phases together with the initial approximation of the inverse transformation function and the final reconstructed inverse transformation function. These routines can be used to set the value of the parameter `DCM.M.Nsig`, and test the final reconstructed inverse function.
3. Routines to plot the reconstructed pairwise coupling functions together with the measured data points. These functions are useful to examine the convergence after the reconstruction procedure. Since the coupling functions $q_{ij}(\varphi_i, \varphi_j)$ and the measured data points (the measured observable phases $\hat{\theta}_i$ and $\hat{\theta}_j$) are in different domains, there are two possibilities to compare them: (1) by representing the data points on the domain of the theoretical phases φ using the inverse transformation functions $\hat{\varphi}_i = \Phi_i(\hat{\theta}_i)$ and $\hat{\varphi}_j = \Phi_j(\hat{\theta}_j)$, (2) or by projecting the coupling functions onto the domain of the observable phases using the forward transformation functions $q_{ij}(\theta_i, \theta_j) = q_{ij}(\theta_i = \Theta_i(\varphi_i), \theta_j = \Theta_j(\varphi_j))$.

These functions are also included in the code of the examples. For more details we refer to the documentation of the eDCM PC package.

(2) AVAILABILITY

OPERATING SYSTEM

- Windows (>=7 SP1, 32-bit or 64-bit),
- Linux (>=Kernel 2.4.x or 2.6.x and >= glibc (glibc6 2.3.4), Ubuntu 14.04 or 14.10
- MacOS (>=X 10.9.5 (Mavericks))

Please refer to the system requirements of MATLAB R2015b, which can be found here: <https://de.mathworks.com/support/requirements/previous-releases.html>.

PROGRAMMING LANGUAGE

The numerical computing language MATLAB.

ADDITIONAL SYSTEM REQUIREMENTS

Minimal requirements for MATLAB R2015b are any Intel or AMD x86 processor supporting the SSE2 instruction set, i.e. processors Intel Core-2-Duo or AMD Athlon 64 (SSE2 required) and above, Disk space >=1024MB (500MB MATLAB and 350 MB SPM), RAM >=1024MB.

For better performance use processors Intel Core i5/i7 from 2012 or newer, Disk space >=3GB, RAM >=4GB.

DEPENDENCIES

eDCM PC needs MATLAB and SPM:

1. MATLAB (>=8.6, R2015b)
2. Statistical Parametric Mapping (SPM) (SPM12, >= Update 6225)

LIST OF CONTRIBUTORS

Please refer to the list of authors.

SOFTWARE LOCATION

Archive

Name: extended Dynamic Causal Modelling for Phase Coupling (eDCM PC)

Persistent identifier: <https://doi.org/10.5281/zenodo.17406441>

License: GNU General Public License v3.0

Publisher: Azamat Yeldesbay

Version published: Version 1.0.0

Date published: 21/10/2025

Code repository

Name: GitLab

Persistent identifier: <https://gitlab.com/azayeld/edcmprc/-/tags/v1.0.0>

License: GNU General Public License v3.0 <https://gitlab.com/azayeld/edcmprc/-/blob/master/LICENSE>

Date published: 21/10/2025

LANGUAGE

English

(3) REUSE POTENTIAL

Despite being originally developed to analyze the connectivity between oscillating brain regions, eDCM PC makes no assumption about the origin of the oscillating system. Therefore, the reuse potential of eDCM PC is not restricted to brain signals, but it can also be used in other biological as well as physical and mechanical systems. In general, this software can be used to reconstruct the effective connectivity in any network of oscillating

systems. The only requirement is the presence of the phase information extracted from the rhythmic signals for every element of the network.

Although the eDCM PC offers a framework for estimating effective connectivity between oscillating systems of any kind using their phase information, its applicability is limited by the weakly coupled oscillators model. Stronger interactions between oscillating systems may require the consideration of phase as well as amplitude information. Moreover, the computational performance of the eDCM PC is limited by the Bayesian inference algorithm underlying DCM. Despite its efficient algorithmic design, the software can lead to prolonged runtimes when processing large-scale computations or high-dimensional datasets. Further development could address this by implementing the algorithm provided in our recent work [33] on the reconstruction of amplitude-phase dynamics. This could also be implemented outside of the SPM (DCM) package.

The collection of codes of eDCM PC and the testing and example scripts, together with the documentation are uploaded in the GitLab repository <https://gitlab.com/azayeld/edcmmpc>, where users can clone and modify the code. The modifications can also be included into the package by sending a merge request. Moreover, the GitLab repository has a build-in support mechanism in the form of an issue tracking system. In the issue tracking system the users of eDCM PC can post their suggestions and problems with the code by opening an issue, which is monitored and will be solved by the authors.

NOTE

- 1 The Data Analysis with Models Of Coupled Oscillators (DAMOCO) toolbox can be found here <http://www.stat.physik.uni-potsdam.de/~mros/damoco.html>.

ACKNOWLEDGEMENTS

We would like to thank Gereon Fink for useful discussions on this project.

FUNDING STATEMENT

This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 431549029 – SFB 1451.

COMPETING INTERESTS

The authors have no competing interests to declare.

AUTHOR CONTRIBUTIONS

Azamat Yeldesbay: Designing, implementing and testing the program code, writing the article and the documentation, creation of the figures.

Silvia Daun: Supervision, writing the article and the documentation, revising the text and the figures.

AUTHOR AFFILIATIONS

Azamat Yeldesbay  orcid.org/0000-0003-0800-7197

Research Centre Jülich, Institute of Neuroscience and Medicine, Cognitive Neuroscience (INM-3), 52425 Jülich, Germany; Institute of Zoology, University of Cologne, Cologne, Germany

Silvia Daun  orcid.org/0000-0001-7342-1015

Research Centre Jülich, Institute of Neuroscience and Medicine, Cognitive Neuroscience (INM-3), 52425 Jülich, Germany; Institute of Zoology, University of Cologne, Cologne, Germany

REFERENCES

1. **Penny WD, Litvak V, Fuentemilla L, Duzel E, Friston K.** Dynamic Causal Models for phase coupling. *Journal of Neuroscience Methods*. 2009;183(1):19–30. DOI: <https://doi.org/10.1016/j.jneumeth.2009.06.029>
2. **Winfree A.** *The Geometry of Biological Time*. Biomathematics, Berlin: Springer Verlag; 1980. ISBN 9783540093732. DOI: <https://doi.org/10.1007/978-3-662-22492-2>
3. **Pikovsky A, Rosenblum M, Kurths J.** *Synchronization: A Universal Concept in Nonlinear Sciences*. Cambridge nonlinear science series. Cambridge University Press; 2001. ISBN 9780511075957. DOI: <https://doi.org/10.1017/CBO9780511755743>
4. **Strogatz S.** *Sync: The Emerging Science of Spontaneous Order*. Penguin Books Limited; 2004. ISBN 9780141933184.
5. **Buzsaki G.** *Rhythms of the Brain*. Oxford University Press; 2006. ISBN 9780198041252.
6. **Kuramoto Y.** *Chemical Oscillations, Waves, and Turbulence*, volume 19. Berlin, Heidelberg: Springer Berlin Heidelberg; 1984. ISBN 978-3-642-69691-6. DOI: <https://doi.org/10.1007/978-3-642-69689-3>
7. **Hoppensteadt FC, Izhikevich EM.** *Weakly Connected Neural Networks*, volume 126 of *Applied Mathematical Sciences*. New York, NY: Springer New York; 1997. ISBN 978-1-4612-7302-8. DOI: <https://doi.org/10.1007/978-1-4612-1828-9>
8. **Rosenblum MG, Pikovsky AS.** Detecting direction of coupling in interacting oscillators. *Physical Review E. Statistical, Nonlinear, and Soft Matter Physics*. 2001; 64(4 Pt 2):045202. DOI: <https://doi.org/10.1103/PhysRevE.64.045202>
9. **Kralemann B, Cimponeriu L, Rosenblum M, Pikovsky A, Mrowka R.** Uncovering interaction of coupled oscillators from data. *Physical Review E*. 2007;76(5):1–4. DOI: <https://doi.org/10.1103/PhysRevE.76.055201>

10. **Kralemann B, Cimponeriu L, Rosenblum M, Pikovsky A, Mrowka R.** Phase dynamics of coupled oscillators reconstructed from data. *Physical Review E* 2008;77(6): 1–16. DOI: <https://doi.org/10.1103/PhysRevE.77.066205>
11. **Kralemann B, Pikovsky A, Rosenblum M.** Reconstructing phase dynamics of oscillator networks. *Chaos*. 2011;21(2): 1–10. DOI: <https://doi.org/10.1063/1.3597647>
12. **Stankovski T, Duggento A, McClintock PV, Stefanovska A.** Inference of time-evolving coupled dynamical systems in the presence of noise. *Physical Review Letters*. 2012;109(2): 1–5. DOI: <https://doi.org/10.1103/PhysRevLett.109.024101>
13. **Kralemann B, Pikovsky A, Rosenblum M.** Detecting triplet locking by triplet synchronization indices. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*. 2013;87(5):1–6. DOI: <https://doi.org/10.1103/PhysRevE.87.052904>
14. **Kralemann B, Pikovsky A, Rosenblum M.** Reconstructing effective phase connectivity of oscillator networks from observations. *New Journal of Physics*. 2014;16(8):085013. DOI: <https://doi.org/10.1088/1367-2630/16/8/085013>
15. **Stankovski T, Ticcinielli V, McClintock PVE, Stefanovska A.** Coupling functions in networks of oscillators. *New Journal of Physics*. 2015;17(3):035002. DOI: <https://doi.org/10.1088/1367-2630/17/3/035002>
16. **Stankovski T, Pereira T, McClintock PVE, Stefanovska A.** Coupling functions: Universal insights into dynamical interaction mechanisms. *Reviews of Modern Physics*. 2017;89(4):045001. DOI: <https://doi.org/10.1103/RevModPhys.89.045001>
17. **Pikovsky A.** Reconstruction of a random phase dynamics network from observations. *Physics Letters, Section A: General, Atomic and Solid State Physics*. 2018;382(4): 147–152. DOI: <https://doi.org/10.1016/j.physleta.2017.11.012>
18. **Stankovski T, Pereira T, McClintock PV, Stefanovska A.** Coupling functions: Dynamical interaction mechanisms in the physical, biological and social sciences. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*. 2019;377(2160). DOI: <https://doi.org/10.1098/rsta.2019.0039>
19. **Yeldesbay A, Fink GR, Daun S.** Reconstruction of effective connectivity in the case of asymmetric phase distributions. *Journal of Neuroscience Methods*. 2019;317(February): 94–107. DOI: <https://doi.org/10.1016/j.jneumeth.2019.02.009>
20. **Friston KJ, Harrison L, Penny W.** Dynamic causal modelling. *NeuroImage*. 2003;19(4):1273–1302. DOI: [https://doi.org/10.1016/S1053-8119\(03\)00202-7](https://doi.org/10.1016/S1053-8119(03)00202-7)
21. **Friston K, Mattout J, Trujillo-Barreto N, Ashburner J, Penny W.** Variational free energy and the Laplace approximation. *NeuroImage*. 2007;34(1):220–234. DOI: <https://doi.org/10.1016/j.neuroimage.2006.08.035>
22. **Friston K, Ashburner J, Kiebel S, Nichols T, Penny W,** editors. *Statistical Parametric Mapping*. Elsevier; 2007. ISBN 9780123725608. DOI: <https://doi.org/10.1016/B978-0-12-372560-8.X5000-1>
23. **David O, Kiebel SJ, Harrison LM, Mattout J, Kilner JM, Friston KJ.** Dynamic causal modeling of evoked responses in EEG and MEG. *NeuroImage*. 2006;30(4):1255–1272. DOI: <https://doi.org/10.1016/j.neuroimage.2005.10.045>
24. **Chen CC, Kiebel SJ, Friston KJ.** Dynamic causal modelling of induced responses. *NeuroImage*. 2008;41(4):1293–1312. DOI: <https://doi.org/10.1016/j.neuroimage.2008.03.026>
25. **Chen CC, Kiebel SJ, Kilner JM, Ward NS, Stephan KE, Wang WJ, Friston KJ.** A dynamic causal model for evoked and induced responses. *NeuroImage*. 2012;59(1):340–348. DOI: <https://doi.org/10.1016/j.neuroimage.2011.07.066>
26. **Stephan KE, Kasper L, Harrison LM, Daunizeau J, den Ouden HEM, Breakspear M, Friston KJ.** Nonlinear dynamic causal models for fMRI. *NeuroImage*. 2008;42(2):649–662. DOI: <https://doi.org/10.1016/j.neuroimage.2008.04.262>
27. **Moran RJ, Stephan KE, Seidenbecher T, Pape HC, Dolan RJ, Friston KJ.** Dynamic causal models of steady-state responses. *NeuroImage*. 2009;44(3):796–811. DOI: <https://doi.org/10.1016/j.neuroimage.2008.09.048>
28. **Tak S, Kempny AM, Friston KJ, Leff AP, Penny WD.** Dynamic causal modelling for functional near-infrared spectroscopy. *NeuroImage*. 2015;111:338–349. DOI: <https://doi.org/10.1016/j.neuroimage.2015.02.035>
29. **Stephan KE, Penny WD, Moran RJ, den Ouden HEM, Daunizeau J, Friston KJ.** Ten simple rules for dynamic causal modeling. *NeuroImage*. 2010;49(4):3099–3109. DOI: <https://doi.org/10.1016/j.neuroimage.2009.11.015>
30. **Daunizeau J, David O, Stephan KE.** Dynamic causal modelling: A critical review of the biophysical and statistical foundations. *NeuroImage*. 2011;58(2):312–322. DOI: <https://doi.org/10.1016/j.neuroimage.2009.11.062>
31. **David O, Harrison L, Friston K.** Neuronal models of EEG and MEG. In: *Statistical Parametric Mapping*, Chapter 33. Elsevier; 2007. pp. 414–440. ISBN 9780123725608. DOI: <https://doi.org/10.1016/B978-012372560-8/50033-4>
32. **Mu J, Liu S, Burkitt AN, Grayden DB.** Multi-frequency steady-state visual evoked potential dataset. *Scientific Data*. 2024;11(1):1–15. DOI: <https://doi.org/10.1038/s41597-023-02841-5>
33. **Yeldesbay A, Huguet G, Daun S.** Reconstruction of phase-amplitude dynamics from signals of a network of oscillators. *Chaos*. 2025;35(4). DOI: <https://doi.org/10.1063/5.0251072>

TO CITE THIS ARTICLE:

Yeldesbay A, Daun S. 2025 Extended Dynamical Causal Modelling for Phase Coupling (eDCM PC). *Journal of Open Research Software*, 13: 33. DOI: <https://doi.org/10.5334/jors.412>

Submitted: 20 December 2021 **Accepted:** 15 November 2025 **Published:** 09 December 2025

COPYRIGHT:

© 2025 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.

Journal of Open Research Software is a peer-reviewed open access journal published by Ubiquity Press.