Project Proposal Dynawo

General information

Name: Dynaωo

Mission statement:

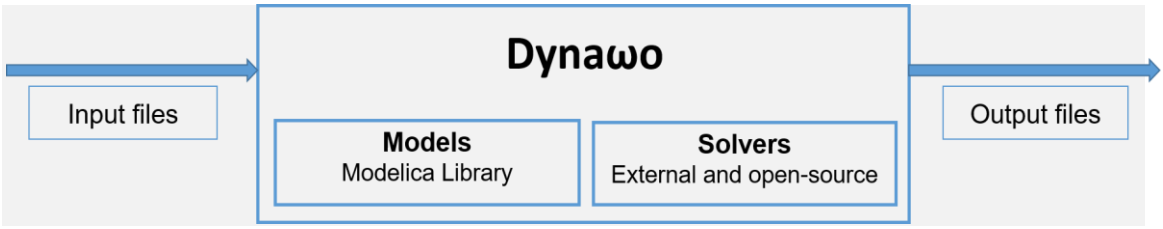
Dynaωo is an hybrid C++/Modelica open source suite of simulation tools for power systems. It aims at providing power system stakeholders with a transparent, flexible, interoperable and robust suite of simulation tools that could ease collaboration and cooperation in the power system community.

Description (what it does, why it is valuable, origin and history):

The nature of power system dynamics is deeply evolving towards a more diverse and difficult to predict behavior due to the massive changes going on in the power system (large penetration of power-electronic based components such as Renewable Energies Sources - RES - or High Voltage Direct Current - HVDC - lines, booming use of complex automata, control strategies or smart grids, ...). Due to this radical change from physically-driven to numerically-driven dynamics, being able to assess the system stability becomes harder but is still essential as any generalized incident will be unacceptable for the economy and the consumers. This requires to have access to a transparent, flexible, robust and easy to use suite of simulation tools that will allow to run collaborative studies in a very simple way by sharing not only the same data but also the same modeling and solving choices in an open-source frame. Such tools will ensure to get similar results and to agree upon optimal and shared actions on the system to accompany the ongoing changes in the best possible way. This analysis has motivated us to launch a new effort on simulation tools that finally ends up in the development of the Dynaωo's software.

To achieve this goal, Dynaωo is based on two mains principles: the use of a high-level modeling language Modelica and a strict separation between modeling and solving parts. Modelica is an equation-based, declarative and object-oriented modeling language that is easy to read and understand (the equations are written in a similar way than they are written in textbooks for example) and already used in different and various industrial sectors. Using this language enables to easily share and discuss the modeling choices done because the final models implementation is available in an understandable way, even for the end-user. It is important to mention that Modelica-based tools already exist (Dymola, OpenModelica, JModelica, etc.) but they are not efficient enough for large-scale simulation of power system, which was one of the motivation for Dynaωo. In addition to this, the Modelica language itself has some limitations that are adressed in

Dynaωo by the possibility to use C++ models in a similar way than Modelica models. The second important point in Dynaωo is the strict separation between modeler and solvers - it means that the models only expose a few methods to the solvers such as the residual evaluation, the Jacobian evaluation or the zero-crossing functions or in other words that the numerical resolution method doesn't interfere in the modeling part. This feature has several advantages: it enables to easily test or use new solvers, it eases the addition of new model and it allows modeling expert not to bother about numerical difficulties and vice versa.



Dynaωo 's primary focus has been on long-term and short-term stability studies but the very encouraging results obtained and the flexibility of the approach led to an extension of the initiative.

Dynaωo is now evolving towards a complete extensive models library sharing the same philosophy.
For sake of clarity, available models library has been classified in a suite to show which kind of simulations and studies can be run using Dynaωo:

- DynaFlow for steady-state calculations
- DySym for short-circuit calculations
- DynaWaltz for long-term stability simulations
- DynaSwing for short-term stability studies
- DynaWave for stability studies and system design with a high-penetration of power-electronics based components (quasi-EMT)

In order to provide a more complete environment around Dynaωo, a serie of companion projects are available in Dynaωo GitHub organization:

- Dynaflow-launcher: utility used to easily run unitary or systematic analysis simulation with DynaFlow using a minimal set of inputs
- Dynawo-algorithms: wrapper around Dynaωo that provides utility algorithms to calculate complex key values of a power system (unitary simulations, systematic analysis, margin calculations and load increase)
- Dynawo-large-scale-validation: a repository of scripts and utilities built during the projects carried out in 2020-2021 for the purpose of validating *DynaWaltz* and *DynaFlow*.

Is this a new project or an existing one?

Existing one started in 2015

Current lead(s)

Marco Chiaramello (RTE)

Sponsoring organization(s), along with any other key contributing individuals and/or organizations

- RTE (Réseau de Transport d'Électricité)
- AIA, Aplicaciones en Informàtic Avanzada
- ULB, Université Libre de Bruxelles
- Fraunhöfer IEE
- RWTH Aachen
- UPC / Citcea
- ...

Existing community infrastructure:

- Github: https://github.com/dynawo
- Website: http://dynawo.org
- Docs: https://dynawo.github.io/docs/
- Communication channels (mailing list): rte-dynawo@rte-france.com
- Slack: https://app.slack.com/client/T01CFV0NB5K/C01C49VDDLP

Specific infrastructure needs or requests outside of what is provided normally by LF Energy

No

Why would this be a good candidate for inclusion in LF Energy?

Stability simulations and more broadly time domain simulations are necessary for industrial utilities to be able to support operational and long-term decisions. Not only, being able to explore future power systems evolutions and physical behaviors is of great interest for academical research. Today there is no such software in the Linux Foundation.

Dynaωo is fulfilling this need with an open industrial-grade solution.

How would this benefit from inclusion in LF Energy?

As this solution becomes mature, LF Energy will provide a clear and safe governance framework and a larger visibility that will open the project to a wider community easing collaboration and cooperation.

Provide a statement on alignment with the mission in the LF Energy charter.

The proposed project is in line with the LF Energy Charter (mission statement 1.a) as open-source project related to transmission of energy.

What specific need does this project address?

This project offers a complete open-source suite of simulation tools for power system. It enables to provide a place to discuss the modelling assumptions done for one kind of simulation or another, to test new numerical schemes on transparent models, to facilitate exchanges on common results or common studies, to try, validate and improve new approaches with all the different stakeholders in a completely open environment.

It aims at providing power system stakeholders with a transparent, flexible, interoperable and robust suite of simulation tools that could ease collaboration and cooperation in the power system community

Describe how this project impacts the energy industry.

Time domain simulations and studies are commonly part of the operational activities of Transmission Systems Operators.

In this framework, Dynaωo is supporting short- and long-term operational decisions, allowing better and shared understanding of power system physical behavior, offering an option for many stakeholders that are eager to promote open-source, transparent and shared approaches.

How does this project intersect with other LF Energy projects?

Dynaωo shares with PowSyBl the data format (IIDM) representing the grid topology and static data. In addition to that, some API are available to run Dynaωo via PowSyBl framework.

Nevertheless, we consider that it makes sense to have two different projects for several reasons (strategical and technical):

- Both projects claim the possibility to be associated to other projects: PowSyBl is integrating different simulation tools, including Dynaωo but not only, and Dynaωo is integrated in different environments, including but not limited to PowSyBl.
- Potential partners are very different: academics for modeling and solving developments and end-users for expression of needs and long-term goal for Dynaωo, IT firms and software developers for PowSyBl for example.
- The technical stacks are very different: C++/Modelica against Java/Groovy notably.

Considering these elements, the choice of two different projects with a strong collaboration and an effort to promote the other one in coupled presentation or demonstration makes sense.

Who are the potential benefactors of this project? / What other organizations in the world should be interested in this project?

This project could be interesting for a lot of actors from the power system community and beyond:

- Academics can have an open, transparent and free access to an industrially-grade suite of simulation tools, facilitating their day-to-day research work (open test case, open models, open discussions)
- Manufacturers can have a standard and transparent approach to use to provide system operators with the necessary information on the material and technologies used (no need to do models in a specific closed language for example)
- System operators (TSO/DSO) have access to validated models, to a transparent frame to discuss and can conduct shared and collaborative studies in a very easy way.
- Coordination entities (RSC, ROC) or associations (ENTSO-E) that have to conduct international studies, draw guidelines and requirements in terms of data quality exchanged or prepare commonly acceptable reference data. An open-source widely recognized tool would be a nice opportunity for these entities to promote and push their ideas and opinions in a transparent way.
- Software providers and vendors that spend a lot of time to validate their results against other closed tools while an open-source reference tool can provide reference results, easy to explain and to interpret.

Beyond the power system actors mentioned below, the approach used in Dynaωo opens new doors for long-term research and possibilities for actors historically not involved in power system problems: around multi-system simulations or cyber-physical studies.

Plan for growing in maturity if accepted within LF Energy

The tool structure is still quite mature, even if improvements could always be added. The next steps will consist in capitalizing on this first effort to decline the approach on the whole range of simulation tools.

In parallel, a special emphasis should be put on providing test cases and examples (standard and validated ones plus anonymized large scale ones) in the project to illustrate the added values and provide a frame that could easily be used and reused by external actors.

The strategical plan can be divided into two main stages:

- The first stage should enable to convince academics and coordination entities to use and promote the tool and its approach.
- The second stage consists in using the already attracted partners to convince and motivate end-users to adopt and promote the tool and its strategies (TSOs, manufacturers). It is targeted for the end of 2024.

Questions for Technical Projects ONLY

Project license

MPL-2.0: Mozilla Public License 2.0

For further information, please check the LICENSE file.

Is the project's code available now? If so provide a link to the code location.

Yes, all the repositories around Dynaωo project are available on GitHub:

https://github.com/dynawo

Does this project have ongoing public (or private) technical meetings?

Today, there are only private technical meetings. Once the project is part of the LFE, the TSC will be implemented to align with LFE governing principles.

Do this project's community venues have a code of conduct? If so, what is it?

Yes, please check CODE_OF_CONDUCT file.

Describe the project's leadership team and decision-making process.

Today, the decision-making process is mainly driven internally to RTE. By January 2023, the project will launch an official public TSC.

Does this project have public governance (more than just one organization)?

No

Does this project have a development schedule and/or release schedule?

Release schedule are today mainly driven by internal RTE needs, based on best efforts.

Dynaωo repositories releases are available approximatively every three months.

For further information, please check RELEASE file.

Does this project have dependencies on other open-source projects? Which ones?

Dynaωo is leveraging on some external libraries:

- OpenModelica: a Modelica environment developed and maintained by the Open Source Modelica Consortium distributed under a GPL V3.0 or OSMC Public License V1.2. The current version used is V1.13.2.

- SUNDIALS: a suite of solvers developed and maintained by the Lawrence Livermore National Lab and distributed under a BSD-3-Clause license. The 4.1.0 version is currently used.
- SuiteSparse: and in particular KLU, a LU decomposition library that is part of the suite sparse project, developed and maintained by T. A. Davis et al. at the University of Florida distributed under a LGPL-2.1+. The version 4.5.4 of suite sparse is the one used by Dynaωo.
- Adept: an automatic differentiation library that has been developed and maintained at the University of Reading by R.J. Hogan distributed under Apache-2.0. It is the version 2.1.1 that is integrated into Dynaωo.
- Xerces-C++: a validating XML parser written in a portable subset of C++ and distributed under the Apache Software License, Version 2.0. The current version used is 3.2.2.
- Libxml2: a XML C parser and toolkit distributed under the MIT License. The current version used is 2.9.4.
- PowSyBl - iidm4cpp: a C++ implementation of the IIDM grid model and distributed under the MPL License, Version 2.0. The current version used is 1.4.0.
- NICSLU: which is another LU decomposition library. It is developed and maintained by Tsinghua University and is optional at the moment into Dynaωo. It is distributed under a GNU LGPL license.
- jQuery: that is distributed into Dynaωo to display results into a minimalistic GUI after the simulation. The current version used is the 1.3.4 distributed under both a MIT and a GPL license.
- cpplint: a tool used during Dynaωo compilation process to ensure that the C++ files follow the Google's C++ style. It is distributed under a CC-By 3.0 License.
- gperftools: a collection of a high-performance multi-threaded malloc implementations distributed under the BSD license, version 2.6.1.
- MPICH: an implementation of the Message Passing Interface (MPI) standard distributed under a BSD-like license, version 3.4.2.

Describe the project's documentation.

The documentation explains how to install Dynaωo, how to configure it to simulate a given test case, gives more details on the model library and the solvers included into the simulation tool and introduces more advanced features such as adding a new Modelica model to the library. For the companion Dynaωo projects, a dedicated pdf documentation is available on each GitHub repository.

https://github.com/dynawo/dynawo/releases/download/v1.3.0/DynawoDocumentation.zip

Describe any trademarks associated with the project.

Not applicable

Do you have a project roadmap? please attach [Are this project's roadmap and meeting minutes public posted?]

The roadmap is public and can be found here: https://dynawo.github.io/roadmap

Does this project have a legal entity and/or registered trademarks?

Registered trademark.

Has this project been announced or promoted in any press?

The project has been promoted through publications and conference papers.

The complete list of works referring Dynaωo is available here:

https://dynawo.github.io/publications/

Does this project compete with other open-source projects or commercial products?

This project covers features that compete with some in commercial and open source solutions, such as:

- DigSilent PowerFactory
- Siemens PSSE
- Etap Transmission System Planning Software
- IED Network Analysis and Planning
- Neplan / Electricity
- Open DSS
- Panda Power
- MATPOWER
- PSAT
- ..