Environment. Technology. Resources. Rezekne, Latvia
Proceedings of the 15th International Scientific and Practical Conference. Volume III, 352-356

Computer Modeling and Simulation of Mechanisms using Free and Open-source Software

Mihail Zagorski
Technical University of Sofia,
Faculty of industrial technology;
"AI and CAD systems Lab",
R&D&I Consortium,
Sofia Tech Park
Sofia, Bulgaria
mihail.zagorski.tu@gmail.com

Radoslav Miltchev
Technical University of Sofia,
Faculty of industrial technology;
"AI and CAD systems Lab",
R&D&I Consortium,
Sofia Tech Park
Sofia, Bulgaria
rmiltchev@tu-sofia.bg

Mario Semkov
Technical University of Sofia,
Faculty of industrial technology;
Lab. "CAD/CAM/CAE in Industry",
Sofia, Bulgaria
msemkov@3clab.com

Abstract. **The present paper aims to examine how free and open-source software (FOSS) can be applied in the processes of computer modeling and simulations of mechanisms. Successful examples of the application of free software for CAD (computer-aided design) and simulation of mechanisms are presented and discussed. A case study is used to exemplify how engineering tasks, related to position function determination, first transfer function and output link's velocity of a four-bar linkage, can be resolved with the help of an open-source scientific programming language – GNU Octave. Recommendations about the applicability of free open-source software in mechanism analysis are provided.**

Keywords: open-source software, GNU Octave, four-bar linkage, CAD software

I. INTRODUCTION

Free and open-source software (FOSS) is software for which the owner of the copyright of the source code grants the rights to study, modify and distribute the software to anyone and for any purpose (or software with an open-source license for short) [1,2]. Open-source software can also be developed in a cooperative public manner [3]. This type of free software is the most common example of open-source development and is often compared to user-generated content (technical definition) or open content movements (legal definition) [4]. GNU (General Public License) is the project within which the free software concept originated. Richard Stallman, the founder of the project, views GNU as a "technical means to a social end" [5]. Relatedly, Lawrence Lessig states in his introduction to the second edition of Stallman's book "Free Software, Free Society" that in it Stallman has written about "the social aspects of software and how Free Software can create community and social justice" [6].

The open-source model, or collaborative development from many independent sources, generates a much more diverse scope in terms of code design and structure than development by a single firm, allowing it to survive over a long period of time [7]. A report by the Standish Group (from 2008) says that the adoption and use of the open-source software model has resulted in savings of $60 billion per year to users [8]. Various companies are benefiting from different open-source solutions. A research estimates that in 2018 FOSS had at least 260 000 contributors in the EU (around 8% of all employees in software development sector). They produced a volume of software code equivalent to the full-time work of 16 000 developers. For the same year the economic impact of open source software stands between $77.8 billion and $113.7 billion [9].

Free and open-source software has several key advantages and disadvantages compared to paid solutions.

The main advantages of FOSS are the following [10]:

- The software is available for use and modification for any purpose, including educational and professional, completely free and unlimited;
- Some open-source solutions require less computing resources than paid alternatives, often because of the lack of graphical user interface;
- A large database of training materials for working with the software is available;
- There is a large community of users;
- The software provides flexibility and freedom for the user.

Print ISSN 1691-5402
Online ISSN 2256-070X
https://doi.org/10.17770/etr2024vol3.8143

The main disadvantages of open-source software are the following [11]:

- The software often has a limited graphical interface, which makes its use more difficult than that of the paid alternatives;
- Usually its use is related to writing commands/code in a given programming language;
- Due to the possibility of anyone working on the software code, there is a risk of "bugs" and unexpected results;
- Some open-source solutions are not compatible with Windows and/or MacOS, so the user needs a dedicated machine or a virtual machine with a specific version of Linux;
- FOSS typically lacks dedicated technical support;
- A dedicated technical support team is often non-existing.

Industrial CAD (computer-aided design) and CAE (computer-aided engineering) solutions for mechanical design and simulation are often very expensive. Open-source solutions are existent and some of them could be a good alternative compared to paid software, especially for scientific researches and education.

The present paper aims to examine different free softwares for computer modeling and simulations of mechanisms. Successful examples of the applications of such solutions are presented and discussed. Case study is used to exemplify how engineering tasks, related to several important parameters of a four-bar linkage, can be resolved with the help of an open-source scientific programming language - GNU Octave.

II. MATERIALS AND METHODS

A. MBDyn

MBDyn is the first and one of the few full-featured free general purpose Multibody Dynamics analysis software, released under GNU's GPL 2.1. It has been developed at the Dipartimento di Scienze e Tecnologie Aerospaziali (formerly Dipartimento di Ingegneria Aerospaziale) of the University "Politecnico di Milano", Italy. MBDyn simulates the behavior of heterogeneous mechanical, aeroservoelastic systems based on first principles equations. The software is being actively developed and used in the aerospace, wind energy, automotive and mechatronic fields (industrial robots, parallel robots, micro aerial vehicles) for the analysis and simulation of the dynamics of complex systems [12].

MBDyn is structured in terms of statements. Statements are logically divided in blocks. Each block is opened by a begin statement and it is closed by an end statement:

```
begin: data;
       # global simulation data
       problem: initial value;
end: data
begin: initial value;
# problem-dependent data
end: initial value;
begin: control data;
# global model data
end: control data;
begin: nodes;
# nodes
end: nodes;
begin: drivers;
# drivers
end: drivers;
begin: elements;
# elements
end: elements;
```

B. Integration of MBDyn to other softwares

MBDyn can be integrated to Blender, FreeCAD (Dynamics Workbench) and MATLAB through add-ons in order to be used for model preparation and visualization.

FreeCAD is an open-source software, which is fully compatible with Python, a widely used free and open-source programming language [13]. FreeCAD is a platform which can be used not only for 3D modelling, but also for various analysis and simulations when the proper add-ons are installed. It has a steep learning curve but provides a completely free alternative to various CAD, CAM and CAE solutions.

Dynamics Workbench project aims at integrating FreeCAD and MBDyn. The objective is to use FreeCAD as pre-post processor for MBDyn, simplifying the process of successfully simulating multibody-dynamics systems and post-processing the simulation results [14]. An exemplary application of Dynamics Workbench is shown in Fig. 1.

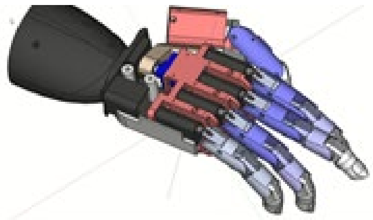

Fig. 1 An exemplary application of Dynamics Workbench.

Blendyn is another interesting project which aims to integrate MBDyn physics to Blender graphics – a so-called graphical post-processor [15]. Blendyn is built as an add-on of Blender which allow user to visualise MBDyn output datas with 3D animations. Blendyn was first implemented in 2017 [16]. ATTILA model, an example of a real-life application of Blendyn, is shown in Fig .2 [17].

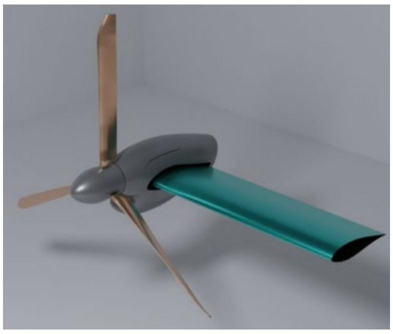

Fig. 2 Semispan Wing ATTILA Tiltrotor Model.

Another example of an integration between MBDyn and engineering software is MBS3D: an easy-to-use open-source general-purpose program for dynamic simulation of multibody systems. It is entirely programmed as plain-text in MATLAB and uses a very efficient and tested mathematical semi-recursive formulation. It covers open-chain and closed-loop systems [18].

MBDyn Analysis can be implemented in MATLAB Graphics, as shown in Fig. 3 [19].

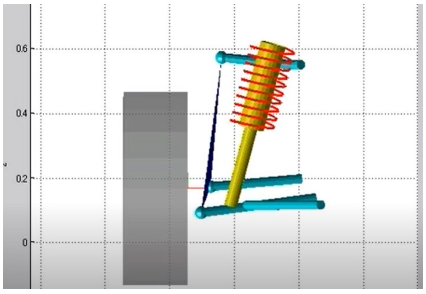

Fig. 3 Double Wishbone Suspension simulation using MBDyn Analysis and MATLAB Graphics.

C. OpenFOAM

OpenFOAM (Open Field Operation And Manipulation) is a C++ toolbox for the development of customized numerical solvers, and pre-/post-processing utilities for the solution of continuum mechanics problems, most prominently including computational fluid dynamics (CFD) [20]. OpenFOAM's finite volume method uses a co-located methodology on an unstructured polyhedral grid with arbitrary grid elements [21]. An exemplary application of the toolbox is shown in Fig. 4 [22].

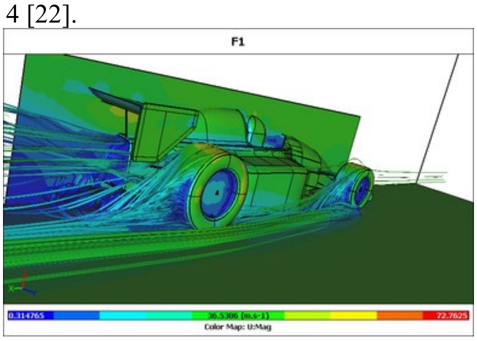

Fig. 4 Flow simulation using OpenFOAM and ParaView for visualization.

D. Linkage

Linkage is a free computer-aided design program for Microsoft Windows used for prototyping of mechanical linkages. The mechanism is edited and animated in the same window allowing for quick analysis and modification while working on a design. It is developed by David Rector – Developer, Designer, and Software Engineer [23].

E. GNU Octave

GNU Octave is a scientific programming language for scientific computing and numerical computation. Octave helps in solving linear and nonlinear problems numerically, and for performing other numerical experiments using a language that is mostly compatible with MATLAB. As part of the GNU Project, it is free software under the terms of the GNU General Public License [24].

III. RESULTS AND DISCUSSION

A case study is developed to exemplify how engineering tasks, related to position function determination, first transfer function and output link's velocity of a four-bar linkage, can be resolved with the help of an open-source scientific programming language - GNU Octave. Linkage is used for visualisation.

A. Visualisation with Linkage

The kinematic scheme of a four-bar linkage – crank-rocker mechanism, is shown in Fig. 5.

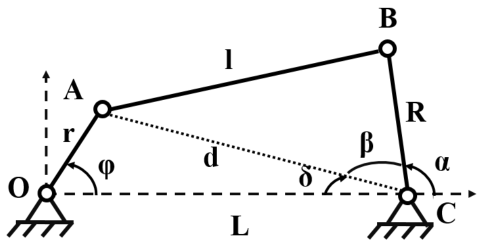

Fig. 5 Kinematic scheme of a crank-rocker mechanism.

The visualisation of the designed mechanism in Linkage is shown in Fig. 6.

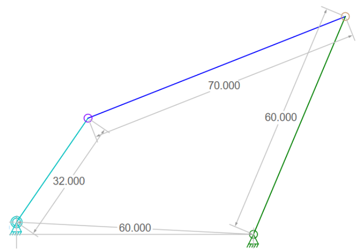

Fig. 6 Visualisation of the designed four-bar mechanism in Linkage.

The dimensions of the different links are as follows:

- Crank (OA = r) = 0.032 m;
- Intermediate link (AB = l) = 0.07 m;
- Rocker (BC = R) = 0.06 m;
- Fixed link (OC = L) = 0.06 m.

B. Calculations and visualizations with Octave

The rocker oscillates between two limiting angles (*α* or *alpha*). The angle between the input link and the fixed link is *φ* or *phi*. The position function of the angle of the output link with respect to the angle of the input link can be calculated in Octave using trigonometric functions with the following code:

```
%input data
r = 0.032;
l = 0.07;
R = 0.06;
L = 0.06;
%defining the angle of the input link
phi = 0:pi/180:2*pi;
%defining the diagonal
d = sqrt(L.^2 + r.^2 - 2.*L.*r.*cos(phi));
%position function alpha = phi(alpha)
alpha = pi - acos((R.^2 + d.^2 -
l.^2)./(2.*R.*d)) - asin(r.*sin(phi)./d);
%radians to degrees
phideg = rad2deg(phi);
alphadeg = rad2deg(alpha);
```

```
%plot of the position function alpha(phi)
plot(phi_deg,alpha_deg,'k');
xlabel('phi',"fontsize",20);
ylabel('alpha', "fontsize", 20);
xlim([0 360]);
set(gca,'XTick',0:30:360);
grid on;
```

The resulting diagram is shown in Fig. 7.

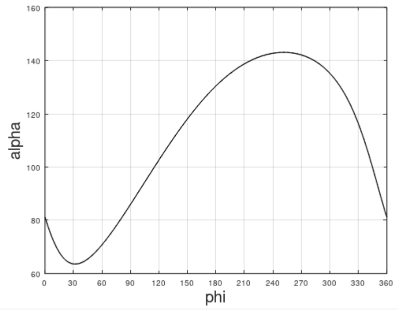

Fig. 7 Position function of the mechanism.

The first transfer function of the mechanism can be calculated by differentiating the position function with the help of the following code:

```
%first transfer function
aDiff = diff(alpha_deg)./diff(phi_deg);

%plot of the first transfer function
plot(aDiff,'k');
xlabel("phi", "fontsize", 20);
ylabel("a'", "fontsize", 20);
set(gca,'XTick',0:30:360);
xlim([0 360]);
```

The resulting diagram is shown in Fig. 8.

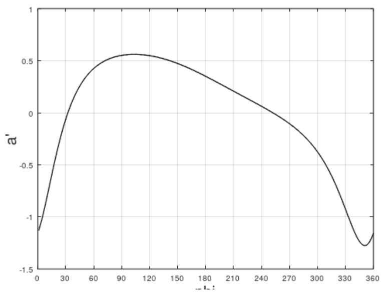

Fig. 8 First transfer function of the mechanism.

The velocity of the output link can be calculated by multiplying the first transfer function by the angular velocity of the input link:

```
%first transfer function
w = 500; %angular velocity of the output link
v = aDiff*w;
plot(v,'k');
xlabel('phi',"fontsize",20);
ylabel("v", "fontsize", 20);
set(gca,'XTick',0:30:360);
xlim([0 360]);
grid on;%plot of the first transfer function
```

The resulting diagram is shown in Fig. 9.

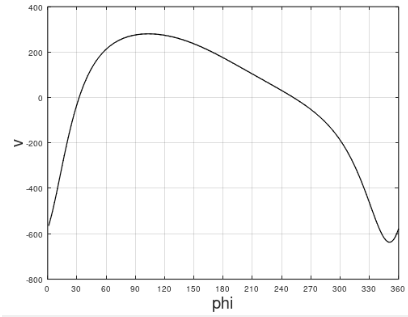

Fig. 9 Velocity of the output link.

The present case study shows some of the advantages of using free and open-source software for engineering tasks. However, there are also disadvantages related to FOSS. For example most of the software is not equipped with user-friendly GUI (Graphical User Interface) and it requires additional add-ons and/or libraries for visualization.

IV. CONCLUSIONS

The software reviewed in the paper can be used for solving of various engineering tasks including but not limited to:

- Simulation of different mechanisms and machines using multibody dynamics simulation software;
- Visualization and computation of various theoretical problems;
- Various tasks including computational fluid dynamics, for example in the field of energetics, for optimization of geometries etc.
- Education and scientific researches.

The licenses for paid engineering software can be worth tens of thousands of dollars. However, paid solutions often provide a better user interface, they are usually much easier to get up and running with substantially less time spent and have better compatibility with other products – hardware or software.

The choice of suitable software – paid, free or open-source, cannot always be considered unambiguously. It is for this reason that the engineer must be familiar with the various software products and select the right one for solving the specific engineering problem.

ACKNOWLEDGMENT

This research work has been accomplished with the financial support of the National Science Fund of Bulgaria under Project KP-06-N67/8 “Development of a fluid-structural methodology for the study and modernization of HYDRAulic turbomachines, through the TECHnologies of virtual prototyping – HydraTech”.

The authors would like to thank the Artificial Intelligence and CAD Systems Laboratory at Research & Development & Innovation Consortium at Sofia Tech Park for supporting of the presented research.

REFERENCES

[1] R. Stallman, “The Free Software Movement and the GNU/Linux Operating System”, 22nd IEEE International Conference on Software Maintenance, 2006.

[2] “License information,” [Online]. Available: https://www.debian.org/legal/licenses/. [Accessed: Feb. 19, 2024].

[3] "Cooperatives Europe developing open source software," Nov. 25, 2019. [Online]. Available: https://coopseurope.coop/news_article/coopseurope-developing-open-source-software/. [Accessed: Feb. 19, 2024].

[4] M. Santos, "The "so-called" UGC: an updated definition of user-generated content in the age of social media," Online Information Review, Vol. 46, 2022.

[5] J. Simpson, "Expanding Human Capabilities Through the Adoption and Utilization of Free, Libre, and Open Source Software", University of the Incarnate Word, 2014.

[6] R. Stallman, "Free software, free society : selected essays of Richard M. Stallman", GNU Press, 2002. [E-book] Available: www.gnupress.org.

[7] D. A. Tamburri, F. Palomba, A. Serebrenik and A. Zaidman, "Discovering community patterns in open-source: a systematic approach and its evaluation," Empirical Software Engineering, Vol. 24, 2019, pp. 1369 – 1417.

[8] R. Rothwell, "Creating wealth with free software," Free Software Magazine, 2008.

[9] "How much are open-source developers really worth? Hundreds of billions of dollars, say economists," Feb 8, 2021. [Online]. Available: https://www.zdnet.com/article/how-much-are-open-source-developers-really-worth-hundreds-of-billions-of-dollars-say-economists/. [Accessed: Feb. 20, 2024].

[10] H. Hamidovic, A. Buljubasic, M. Djulic, E. Sejic, "BENEFITS AND RISKS OF FOSS STRATEGIES", Conference: DIEC 2022, 2022.

[11] M. Heron, V. Hanson and I. Ricketts, "Open Source and Accessibility: Advantages and Limitations", Journal of Interaction Science, Vol. 1, 2013.

[12] P. Masarati, M. Morandini and P. Mantegazza, "An Efficient Formulation for General-Purpose Multibody/Multiphysics Analysis", Journal of Computational and Nonlinear Dynamics, Vol. 9, 2014.

[13] F. Machando, N. Malpica and S. Borromeo, "Parametric CAD modeling for open source scientific hardware: Comparing OpenSCAD and FreeCAD Python scripts", PLOS ONE, Vol. 14, 2019.

[14] "FreeCAD-MBDyn dynamics workbench," [Online]. Available: https://gitlab.com/josegegas/freecad-mbdyn-dynamics-workbench. [Accessed: Feb. 20, 2024].

[15] "Blendyn," [Online]. Available: https://github.com/zanoni-mbdyn/blendyn. [Accessed: Feb. 20, 2024].

[16] "Blendyn Development," [Online]. Available: https://github.com/BryanGsep/Blendyn-Development/blob/main/README.md. [Accessed: Feb. 20, 2024].

[17] A. Cocco, S. Mazzetti. P. Masarati, S. Hoff and B. Timmerman, "Numerical Whirl–Flutter analysis of a tiltrotor semi-span wind tunnel model", CEAS Aeronautical Journal, Vol. 13, 2022, pp. 923 – 938.

[18] C. Caccia and P. Masarati, "Coupling multi-body and fluid dynamics analisys with preCICE and MBDyn", 9th edition of the International Conference on Computational Methods for Coupled Problems in Science and Engineering, 2021.

[19] "MBDyn Tutorial," [Online]. Available: https://www.sky-engin.jp/en/MBDynTutorial/. [Accessed: Feb. 20, 2024].

[20] H. Medina, A. Beechook, J. Saul, S. Porter, S. Alexandrova and S. Benjamin, "Open source Computational Fluid Dynamics using OpenFOAM," Light Aircraft Design: Methods and Tools 2015, London, 2015.

[21] H. Jasak,, "OpenFOAM: Open source CFD in research and industry", International Journal of Naval Architecture and Ocean Engineering, Vol. 2, 2009, pp. 89 – 94.

[22] "OpenFOAM+ParaView_flow_simulation.png," [Online]. Available: https://upload.wikimedia.org/wikipedia/en/e/e9/OpenFOAM%2BParaView_flow_simulation.png. [Accessed: Feb. 22, 2024].

[23] "Linkage Mechanism Designer and Simulator," [Online]. Available: https://blog.rectorsquid.com/linkage-mechanism-designer-and-simulator/. [Accessed: Feb. 22, 2024].

[24] J. Eaton, "GNU Octave and reproducible research", Journal of Process Control, Vol. 22, 2012, pp. 1433 – 1438.