

## COMMUNICATION OF MATLAB GUI AND ANSYS MAXWELL: AN EDUCATION TOOL FOR TUBULAR LINEAR GENERATOR

S. Arslan<sup>1\*</sup>, O. Gürdal<sup>2</sup>, S. Akkaya Oy<sup>3</sup>

<sup>1</sup>Department of Electrical, Harran University, Urfa, 63400, Turkey

<sup>2</sup>Gazi University, Faculty of Technology, Department of Electrical and Electronic Engineering,  
Ankara, 06500, Turkey

<sup>3</sup>Fatsa Faculty of Marine Sciences, Ordu University, 52400, Turkey

Received: 23 September 2018 / Accepted: 06 December 2018 / Published online: 01 January 2019

### ABSTRACT

In this study, analytical calculations of a tubular linear generator model are performed by using Matlab Gui for free-piston applications. A training tool including the processes of design of linear generator, sizing with the analytical equations and analyzing with the finite elements methods was suggested. Geometric dimensions in Matlab Gui interface are solved by transferring automatically to finite element model Ansys-Maxwell 2D-rz plane. The results of analytical analysis are in concordance with the results of finite elements. In addition, Ansys-Maxwell Script and Matlab codes are given in the appendix.

**Keywords:** Linear generator; magnetic field; ansys maxwell; matlab gui; visual basic script

Author Correspondence, e-mail: [serdalarslan@harran.edu.tr](mailto:serdalarslan@harran.edu.tr)

doi: <http://dx.doi.org/10.4314/jfas.v11i1.9>

### 1. INTRODUCTION

Even though the operating principles of the electrical machines we use in our daily lives have not changed, significant changes in the materials used and the structure geometries have been made. Thus, new perspectives in machine designs have occurred and the application fields have

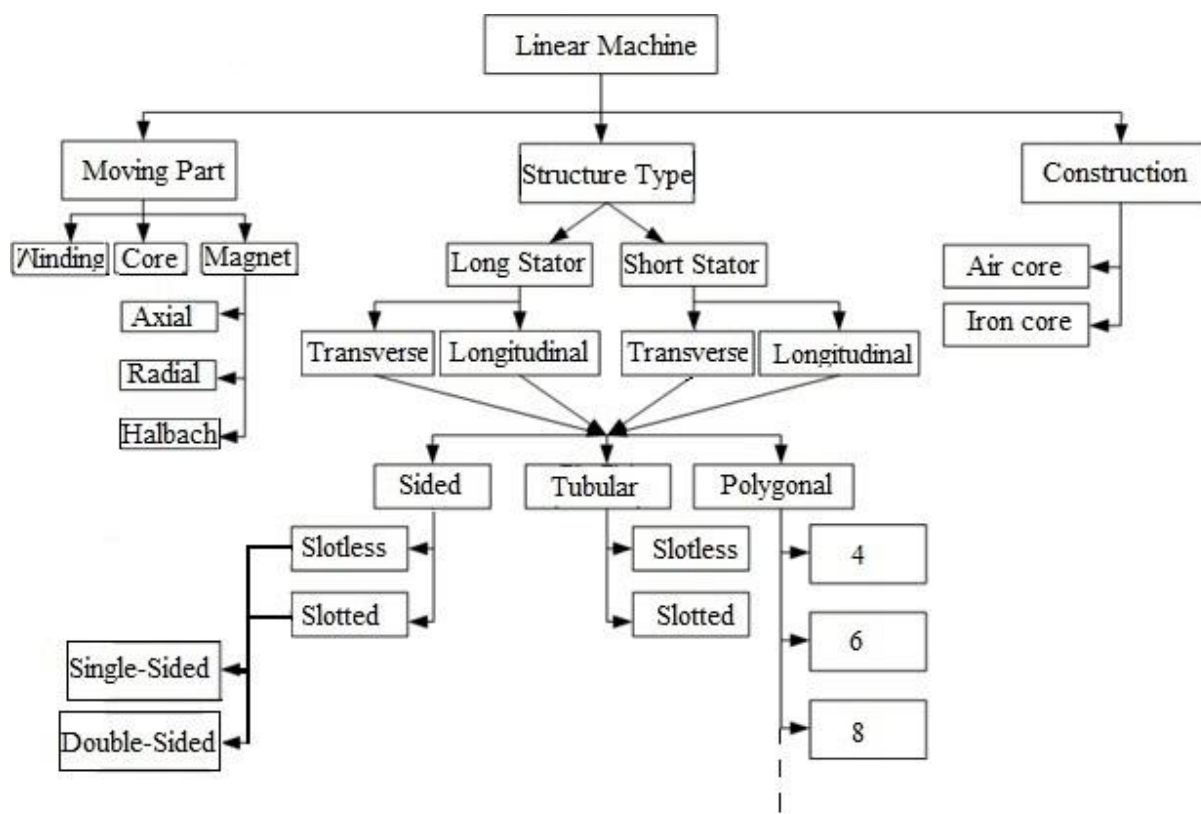


---

developed. One of these is the design of linear machines (LM). Linear generators are the machines producing electrical energy using any one of coil movement, core movement or the magnet movements. In short, it is based on the principle of inducing voltage on a conductor moving linearly in the magnetic field.

Linear generators are used in free-piston applications (hydrogen, gas, hybrid vehicles using gasoline) making reciprocation movement, wave energy (Archimedes wave oscillation system), the Stirling system, pen recorders, pumps, electromagnetic valve mechanism, active shock absorbers, vibrators, compressors, cell phone chargers, portable entertainment equipment, mobile lighting devices and in aerospace applications. High power machines are designed as high velocity ( $2 \text{ m/s} < \text{translator velocity}$ ) for free piston practices and as low velocity ( $\text{translator velocity} < 2 \text{ m/s}$ ) [1,2] for wave energy practices. Other practices can be assessed as very small or small powered practices. Due to the requirement for high force density of the generator for free-piston application or WEC application [1], a PM generator with slotted iron-cored tubular topologies has been proposed in this study.

Even though the first studies conducted on linear machines are on linear induction motor, continuous developments on permanent magnet technology have accelerated the works performed on new types of linear motors/generators. Comprehensive books are written on existing and new linear machine topologies (see Figure 1), application areas, advantage and disadvantages [3-6].



**Fig.1.** Linear Machine Topology

Arshad et al. investigated linear generator topologies that can be used in free-piston applications [7-9]. They have made a comparison in terms of efficiency, moving weight, magnet structure, power density and power factor. Wang and Howe tried to determine an appropriate geometry and analytical equations for a radial flux tubular linear machine [10]. Chen et al compared axial magnetic tubular linear generator with longitudinal flux for free-piston applications depending on the change of the number of poles in terms of iron, copper loss, efficiency and power density [11]. Wang and Howe compared the usage of soft magnetic materials (Samaloy 500-700) with silica plate (Transil 300) in tubular linear machines in terms of efficiency, power factor, power, iron and copper loss [12]. It is shown that silica plate materials give a better performance and the soft materials provide ease in manufacturing. Want et al. tried to estimate the knocking force with analytical equations in order to reduce knocking force in tubular linear machines and proposed two techniques to eliminate knocking force [10,13]. Pohl and Graf performed dynamic analysis of free-piston linear generator with Modelika program [14]. Ahmad et al. conducted a research about reducing knocking force by

---

using special magnet structure in tubular linear generators [15]. Although there are design interfaces for rotating machines in literature (virtual platforms), these interfaces are limited to linear systems. Philip et. al. developed an interface for a tubular linear motor with axial flux [16]. In this study, they performed the analysis of a new type primary tooth structure by changing the traditional tooth structure of the primary. Xiaoting et. al. created a design interface for tubular linear machines with axial flux [17]. They compared the data from the motor they designed analytically in the interface with the results they obtained by using the finite element method. Oswald and Herzog [18] described the design process of a tubular linear synchronous machine. Their paper does not include one or more special results, but offers an overview about all necessary steps of the automated design process and its interfaces with the optimization algorithm. Bianchi et. [19] developed an interface using Finite Element Codes. The object-oriented programming has been applied to the finite element analysis of the radial flux PM motors.

In this study; radial flux tube-type linear generator design was carried out. Package programs given in literature are investigated for the finite element solution. For the non-traditional generator topologies, there is no ready design tool available where analytical and finite element methods work together. Unlike the other studies, Matlab-GUI design interface is created in this study. In addition, steady-state and transient analysis is performed by creating the linear generator in Ansys-Maxwell 2D-rz plane automatically according to the obtained sizing data. General information about the obtained analysis results and design interface are provided.

### **1.1 Design of the Tubular Linear Generator**

One of the application areas of tubular linear generator is the free-piston application. Free-piston system is creating linear movement through a single carrier by removing crank slider of the internal combustion motor. In order for an internal combustion motor to run, the first movement is provided by running the linear generator as a motor. Conceptually linear generators comply with the working principles and conversions of rotating type machines. However, the basic characteristics vary (e.g. torque and angular velocity correspond to force and linear velocity, respectively). As known, the design is started with the power statement of traditional rotating motor or generators. But the linear machines are characterized with the force.



defined with Eq. 5:

$$L_p = T_{op} N_p \quad (5)$$

$D$  ve  $D_e$  value;

$$D = 2H_s + 2L_m + 2g \quad (6)$$

$$D_e = D + 2H_w + 2H_p \quad (7)$$

The ratio between the internal diameter and outer diameter is calculated as (Gama). The operating distance of linear generator is stated as the stroke height. Total stroke height is  $2L_{str}$  due to back and forth motion. Therefore, total secondary length is calculated with the Eq. 8:

$$L_s = L_p + 2L_{str} \quad (8)$$

Primary tooth width;

$$T_w = T_{os} - B_w \quad (9)$$

The ratio of hollow width ( $B_w$ ) to thread width ( $T_w$ ) affects the electrical and magnetic loading. As this ratio increases, the electrical loading also increases whereas the magnetic loading decreases due to the stray fluxes. It is crucial to select this ratio as 1.6 or at the range of 1.4-2 for maximum force [20]. Under these criteria, taking the hollow width and thread width equal is enough for the starting of design. Therefore, the ratio between the hollow width and hollow step (Beta) was taken 0.5. the hollow width;

$$B_w = T_w \quad (10)$$

While  $H_{p0}$  is given as the open hollow height,  $H_{p1}$  as the maximum hollow stud height,  $B_{w0}$  as the semi-open hollow width and  $B_{s1}$  as the top width in the radial smart machines, it is taken as equal to the bottom width ( $B_w$ ) in the linear machines.  $H_s$  is given as the coil height. This is given in equation 11.

$$H_w = H_{pw} - H_{p1} + H_{p0} \quad (11)$$

Stray flux for open hollow [21];

$$\lambda_{1s} \approx \frac{H_{pw}}{3B_w} + \frac{H_{p1} + H_{pw}}{B_w} \quad (12)$$

Effective air gap for open hollows [3];

$$\gamma \approx \frac{(B_w / g)^2}{5 + (B_w / g)} \quad (13)$$

$$K_c = \frac{1}{1 - \gamma g / T_{os}} \quad (14)$$

$$g_e = g K_c \quad (15)$$

The equation for the single layer collective coiling is as follows.

$$k_k = \frac{X_s}{6} \quad (16)$$

For bilayer collective coiling;

$$k_k = \frac{X_s}{3} \quad (17)$$

Coil temperature and coil resistance ( $\gamma_{20} = 0,0039$ )

$$R_{f20} = \rho_{20} \pi (D + Hp_w) \frac{k_k N_f j}{I} \quad (18)$$

$$\gamma = [1 + \gamma_2 (\Delta T)] \quad (19)$$

$$R_f = \gamma R_{f20} \quad (20)$$

Copper loss,

$$P_{cu} = 3 R_f I_f^2 \quad (21)$$

Iron loss,

$$P_{fe} = P_h + P_c + P_e \quad (22)$$

$P_h$  hysteresis loss,  $P_c$  eddy loss,  $P_e$  extra eddy loss

$$P_v = k_h fB_m^2 + k_c (fB_m)^2 + k_e (fB_m)^{1.5} \quad (23)$$

The efficiency of generator is determined in the form of the ratio of output power to the input power. The main factors which affect the efficiency are the losses. Here,  $P_o$  is given as output power,  $P_i$  as input power,  $P_{cu}$  as total copper loss which occur at rotor and stator,  $P_{fe}$  as total iron loss which occur at rotor and stator,  $P_m$  as mechanical losses (5% of output power [16]) and  $P_{wf}$  as the friction and wind losses three speed (fixed, trapezoidal and sinusoidal) profiles (approximately 2.7% of output power [22]):

$$P = 3E_f I_f \quad (24)$$

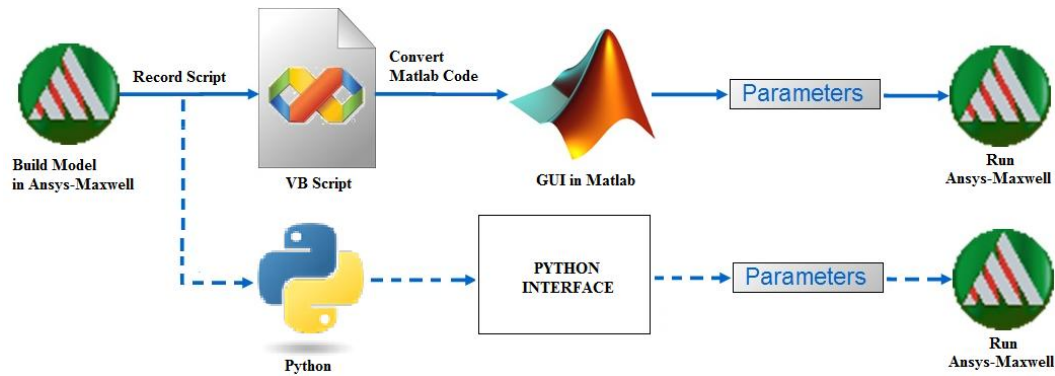
$$\eta = \frac{P_o}{P_i} = \frac{P_i - P_{cu} - P_{fe} - P_{wf}}{P_i} \quad (25)$$

## 1.2 Interface Designed in Matlab Gui

Matlab is one of the most commonly used program in solving engineering problems. Data obtained analytically from Matlab provide visual and ease of use in GUI platform. However, it is quite difficult to work with Matlab in finite element method applications. Instead, using package programs (Ansys, Comsol, Femlab, Infolytica etc.) using finite element method provides convenience to the users. Moreover, these package programs provide ease of drawing defined in their own interface (electric machinery modules etc.), and design of models. However, drawing unidentified or hard to define power geometry or systems, performing the analysis will take a long time. In addition, analyses like optimization become limited by the manufacturer. Therefore, some package program producers allow entering codes written in a way to communicate with the other software. For example, Ansys Maxwell Visual Basic Script supports Python language. The codes converted to these software languages must be converted into Matlab codes. The model converted to Matlab code is transformed into a dynamic model. Thus, in addition to applying different optimization methods, it allows to create your own model interface through GUI. In addition, by both addressing the users' conceptual imaginations and the development of new machine topologies (linear machines, cross or transverse, vernier-type or double-rotor or double-stator, etc. machines); package

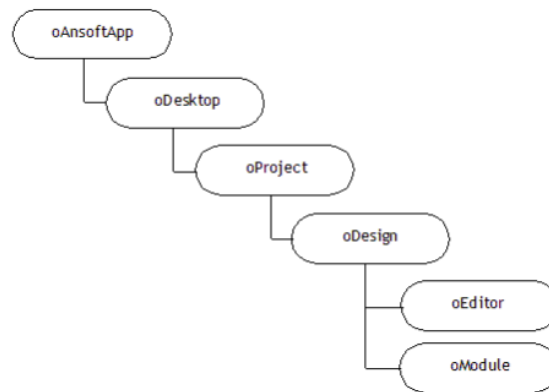


program developers make the package programs dynamic instead of creating finished interface for each machine topology. Briefly; by using the software languages of the package program, the solution of different designs can be performed. Ansys-Maxwell used widely in the literature supports Script or Python languages. Code conversions can be seen in Figure 3.



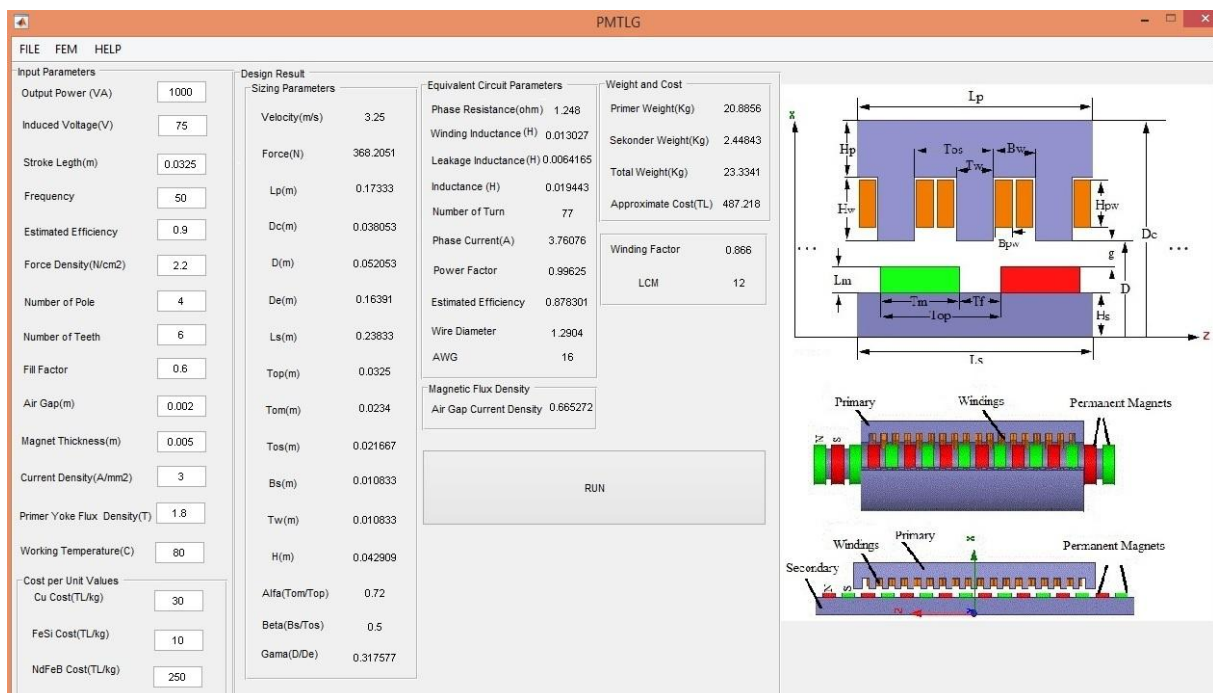
**Fig.3.** Code conversion processes

Firstly, from the settings of Ansys-Maxwell; .exe file of Matlab program installed in Miscellaneous Options tab is shown. Compatibility of Ansys-Maxwell used with the installed Matlab version should be taken into account. While creating the geometry of the model wanted to be designed in Ansys-Maxwell, it is recorded step by step into the Script file. This script can be recorded in two different extensions (.vbs and .py). In this study it is converted into Matlab language by recording as .vbs. At the same time, input parameters required for the design are defined in Matlab GUI. Since adding these parameters into the calculations and processing are made with pushbutton, converted code are entered underneath the pushbutton. In order for the prepared code to work in Ansys-Maxwell environment automatically, the command line given in Appendix 1 should be written at the beginning of the program. The definition of the parameters and the formula lines are then written and executed. However, other important point every main step in the design tool in Ansys-Maxwell is located as a module (Fig.4). While each module is defined separately, the name and the function of that module are defined. General information about writing the Script code can be found from the help files of Ansys-Maxwell [23].



**Fig.4.** Variable hierarchy of Ansys Maxwell Script Codes

Ansys Maxwell has a model under the name of Rmxprt for the analytical analysis of rotating machines. This module performs the analytical analysis of rotating machines. The analysis results are directly transferred to Maxwell 2D/3D module and analyzed with the finite elements method. But the design and analysis of linear machines due to that there is no linear generator design module in the current commercial software. Therefore, an interface which may be used in the design of linear machines was suggested (see Figure 5). This interface creates the sizing equations of generator within the direction of basic input parameters. The magnetic flux density at the air gap is calculated with the magnetic equivalent network model.



**Fig.5.** Matlab-GUI interface

FILE menu consists of the tabs of base open, save, print out and exit. Within the direction of sizing data obtained, the analyses may be conducted by directly selecting the analysis types under FEM menu. FEM menu is given in Figure 6 in detail. HELP menu includes the developer contact information.

FILE	FEM	HELP
Input F		Manyetostatic
Output		Transient Noload
Induc		Transient-Cogging Force
		Transient Loaded

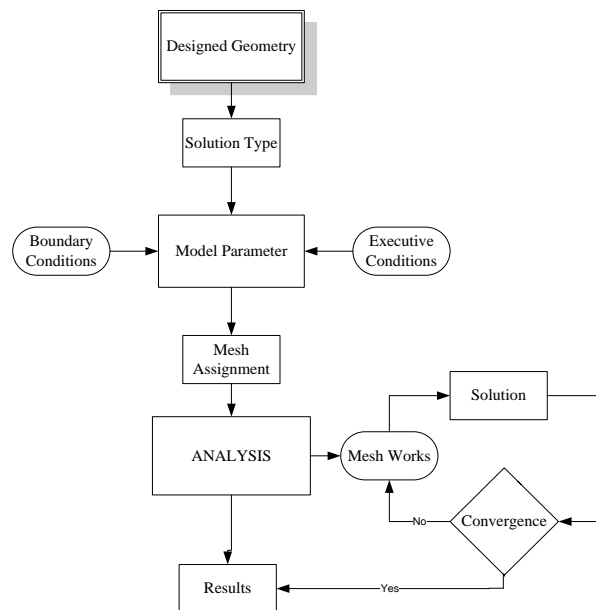
**Fig.6.** The solution methods under FEM menu are as follows

FEM menu are as follows:

- **Magnetostatics:** It gives the distribution of air gap magnetic flux density of geometry which is automatically created in Maxwell 2D and the graphic of flux density between the stator threads.
- **Transient Noload:** The change of voltage induced in the coils and core losses in case the geometry which is automatically created in Maxwell 2D operates at rated speed under unloaded working conditions are taken.
- **Transient Cogging Force:** The cogging force occurs due to the attraction power between the magnet and core threads. This force creates adverse effects on the net force. Therefore, it is crucial to know the form and amount of this force. With the analysis performed under the unloaded working conditions of geometry which is automatically created in Maxwell 2D, the cogging force may be taken along the slot pitch. Knocking force varies depending on the direction of the magnet flux. In the linear radial flux machine, the knocking force produces less knocking force than the linear axial flux machine [2].
- **Transient Loaded:** The flux changes belonging to the phases, load current, output power change, coil inductance changes may be taken at the rates speed of geometry created automatically in Maxwell 2D and under ohmic loaded working conditions.

A real structure of a system is taken into account by finite elements methods [24] for 2D/3D model analysis. Using finite elements methods save costs and time as well. The work

flow-chart of the software is given in Figure 7.



**Fig.7.** The work flow-chart of the finite elements methods

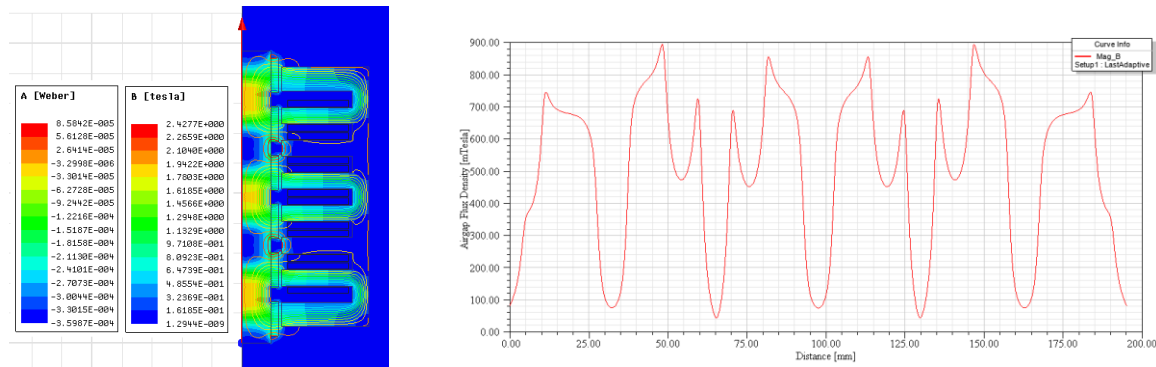
Desired analyses may be conducted in the solution methods under FEM menu. With the sizing data which were obtained as a result of analytical calculations, the flux diagram is applied in Figure 6 in accordance with the solution type selected in Figure 5. In Figure 6, the limit conditions were automatically assigned for every solution in the form of that the management parameters have been previously determined. The assignment of network elements for the finite elements was automatically made by taking the sizes of steel and magnet into consideration. Ansys Maxwell 2D/3D separates analysis region to a great number of mesh zones for determining electrical or magnetic fields in analyze of non-smooth geometrical models. Every element node and electrical potential in middle of edges are related as polynomial. Solving polynomials make analysers to reach to the result [25]. They were determined with 1% error in the software for the mangetostatic analyses. Also, total solution time (stroke length/velocity) was taken for transient time analysis.

### 1.3 Design Sample

FEM results obtained with the sizing data which were acquired within the direction of input parameters given in Figure 5 are given:

**Magnetostatic:** The magnetic flux lines, magnetic flux density and air gap flux density which

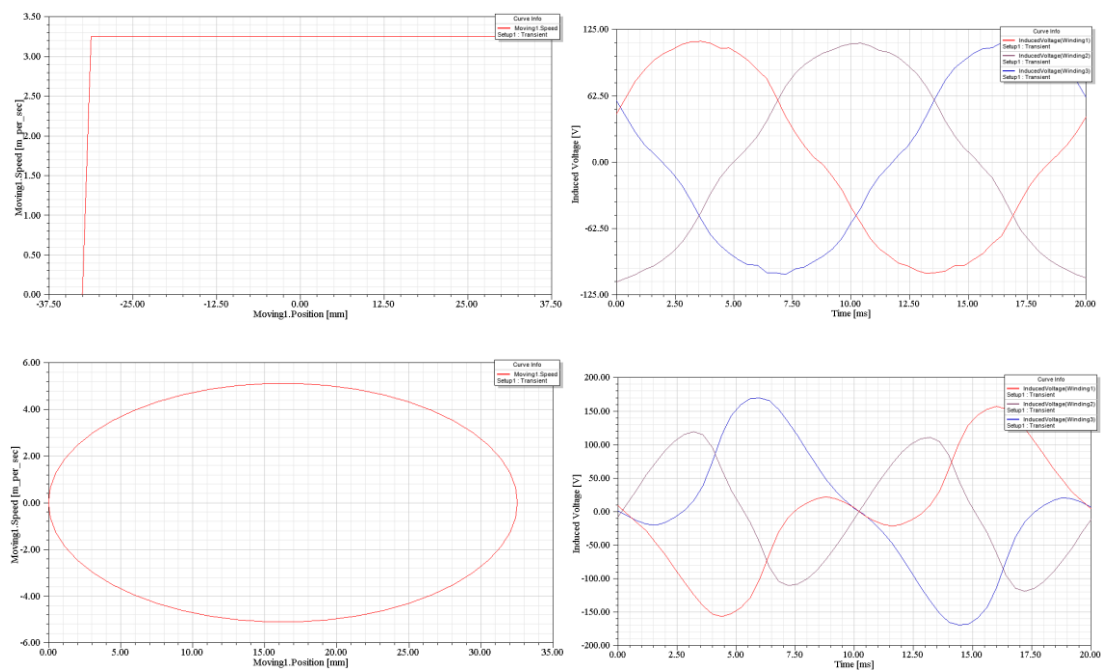
were obtained as a result of magnetostatic analysis are given in Figure 8:



**Fig.8.** (a) Magnetic flux lines, (b) Air gap magnetic flux density change

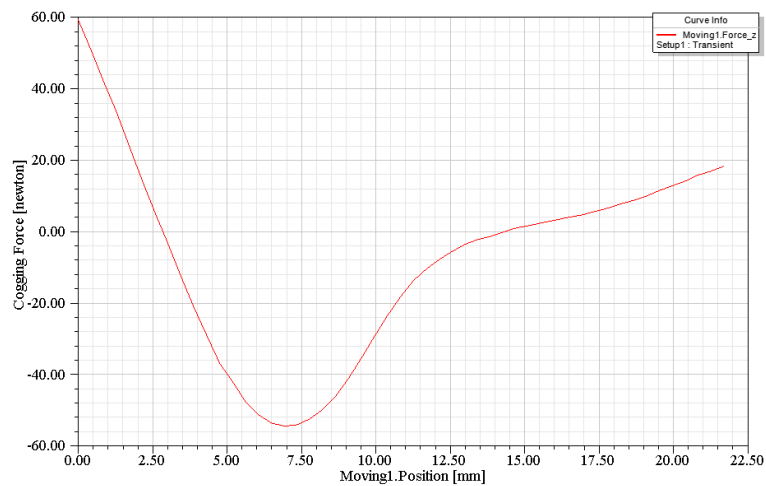
The air gap flux density given in Figure 8 is very close to the flux density calculated with the analytical calculation given in Figure 5.

**Transient Noload:** The unloaded working conditions occurred at the rated speed given in Figure 5. The solutions were automatically taken individually as two speed functions. When the speed is accepted as linear, the voltage induced in the coils are given in figure 9 (a), figure 9 (b) and it is given in figure 9 (c), figure 9 (d) in case the speed changes as sinusoidal:



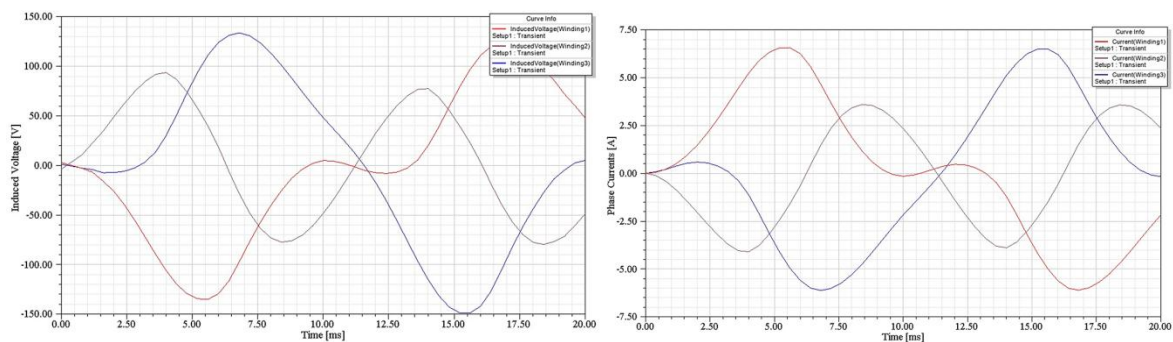
**Fig.9.** (a) Speed change (linear), (b) voltage induced in the coils, (c) Speed change (Sinusoidal), (d) Voltage

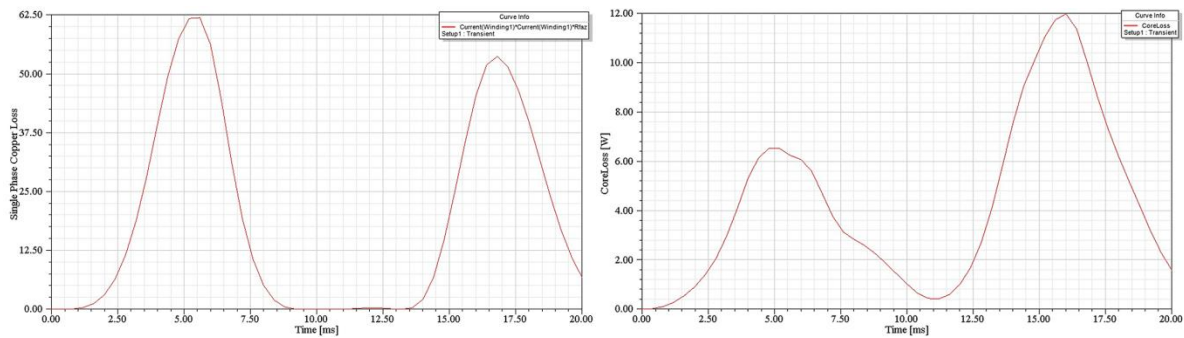
**Transient Cogging Force:** As the cogging force occurs as a result of the attraction power between the core thread and magnet, it is required the current the coils shall be dead. It is very hard to calculate the cogging force with the analytical calculation. In this respect, any analytical analysis was not made. Therefore, finite elements analysis was conducted. The sizing result which was obtained according to the sizing parameters given in Figure 5 is given in Figure 10:



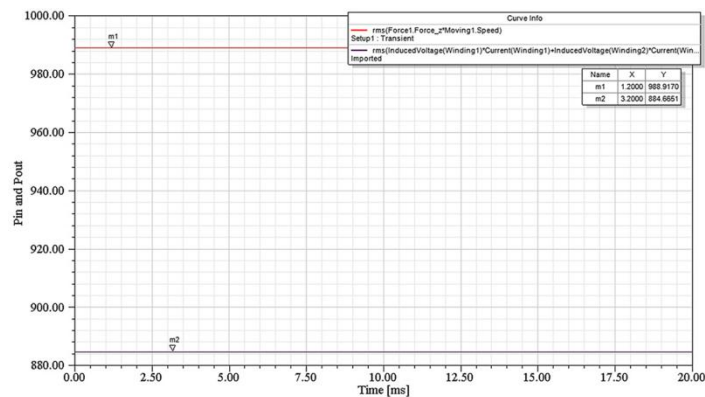
**Fig.10.** Cogging Force

**Transient Loaded:** In order to fully review the operating status of generator, the analyses shall be performed as loaded. In this respect, the results which were obtained along with the generator's loading ohmic load at the rated speed (phases voltages, phase currents, copper loss, iron loss, are given figure 11 and input and output powers are given figure 12. )





**Fig.11.** (a) Coil voltages, (b) current change, (c) Copper loss, (d) iron loss change



**Fig.12.** Input power and output power change

While the efficiency estimated as a result of the sizes given in figure 5 was 87%, the efficiency obtained under full load working conditions with finite elements was calculated as 89%. It was also seen that it was in concordance with the results of analytical calculation in terms of the phase voltage and phase current.

## 2. CONCLUSION

In this paper, a tubular linear generator design interface program was developed. This study is significant in terms of creating a source devoted to the studies of undergraduate, graduate and postgraduate students who study in the field of electrical machines concerning to the modeling of linear generators with analytical and finite elements. FEM based programs which have been currently used consist of modules which use analytical and finite elements method for rotating machines and axial machines. But the programs which have been currently used have no analytical sizing modules for linear generator solution. Therefore, this study fills the gap in this



field and offers alternative solutions. It is an interactive interface which teaches the students of electrical engineering and electrical machines how the design is conducted by using Ansys Maxwell with the schematic equations. The parametric analyses may be conducted by intervening to Ansys Maxwell as a result of the analyses conducted. Therefore, understanding the effect of variables will make contribution to the understanding of students. It is an example for automatic drawing and analysis for simple or complicated problems by preparing an interface in Matlab Gui with visual basic script based commercial software.

In the future studies, the optimization of relevant cost functions will be conducted by using the optimization methods which has been currently offered by Ansys-Maxwell. Also, the application of new optimization algorithms to the finite elements problems will be ensured along with that Matlab and Ansys Maxwell may communicate and perform the optimization. Therefore, efficient and effective analysis will have been conducted for size optimization with the optimization algorithms.

#### **Conflict of interest**

None declared.

#### **Funding**

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

### **3. APPENDIX**

Since the code included by the developed design tool is long, the conversion of Ansys Maxwell Script code into Matlab codes is shown below with a brief example. In order for the example to work, Matlab and Ansys Maxwell programs must be installed into the computer. The example is given in 2D rz plane. Therefore, from the setting section of the program, 2D Magnetostatics rz plane must be selected. It is adequate to write Matlab code in an M-file and run.

<b>Script Code</b>	<b>Matlab Code</b>
Dim oAnsoftApp Dim oDesktop Dim oProject Dim oDesign	iMaxwell=actxserver('AnsoftMaxwell.MaxwellScriptInterface'); Desktop = iMaxwell.GetAppDesktop();



<pre>Dim oEditor Dim oModule Set oAnsoftApp = CreateObject("AnsoftMaxwell.MaxwellScriptInterface") Set oDesktop = oAnsoftApp.GetAppDesktop() oDesktop.NewProject Set oProject = oDesktop.GetActiveProject() oProject.InsertDesign "Maxwell 2D", "Maxwell2DDesign1", "Magnetostatic", "" Set oDesign = oProject.SetActiveDesign("Maxwell2DDesign1") Set oEditor = oDesign.SetActiveEditor("3D Modeler") Set oDesign = oProject.GetActiveDesign()</pre>	<pre>Desktop=invoke(iMaxwell,'GetApp Desktop'); Desktop=iMaxwell.GetAppDesktop p(); Desktop.NewProject Project =Desktop.GetActiveProject(); invoke(Project,'InsertDesign','Max well 2D', 'Maxwell2DDesign2', 'Magnetostatic', '') Design = Project.GetActiveDesign(); DefinitionManager=Project.GetDef initionManager() Desktop.CloseAllWindows() Editor = Design.SetActiveEditor('3D Modeler');</pre>
<pre>oEditor.CreateRectangle Array("NAME:RectangleParameters", "IsCovered:=", true, "XStart:=", _ "10mm", "YStart:=", "0mm", "ZStart:=", "20mm", "Width:=", "-20mm", "Height:=", _ "10mm", "WhichAxis:=", "Y"), Array("NAME:Attributes", "Name:=", "Rectangle1", "Flags:=", _ "", "Color:=", "(132 132 193)", "Transparency:=", 0, "PartCoordinateSystem:=", _ "Global", "UDMId:=", "", "MaterialValue:=", "" &amp; Chr(34) &amp; "vacuum" &amp; Chr(34) &amp; "", "SolveInside:=", _ true)</pre>	<pre>invoke(Editor,'CreateRectangle', ... {'NAME:RectangleParameters', 'IsCovered:=' true, 'XStart:=' ... '10mm', 'YStart:=' '0mm', 'ZStart:=' '20mm', 'Width:=' '-20mm', 'Height:=' ... '10mm', 'WhichAxis:=' 'Y'}, {'NAME:Attributes', 'Name:=' 'Rectangle1', 'Flags:=' ... ', 'Color:=' '(132 132 193)', 'Transparency:=' 0, 'PartCoordinateSystem:=' ... 'Global', 'UDMId:=' , 'MaterialValue:=' 'vacuum', 'SolveInside:=' ... true})</pre>
<pre>oEditor.CreateRectangle Array("NAME:RectangleParameters", "IsCovered:=", true, "XStart:=", _ "25mm", "YStart:=", "0mm", "ZStart:=", "20mm", "Width:=", "-20mm", "Height:=", _ "-5mm", "WhichAxis:=", "Y"), Array("NAME:Attributes", "Name:=", "Rectangle2", "Flags:=", _ "", "Color:=", "(132 132 193)", "Transparency:=",</pre>	<pre>invoke(Editor,'CreateRectangle', {'NAME:RectangleParameters', 'IsCovered:=' true, 'XStart:=' ... '25mm', 'YStart:=' '0mm', 'ZStart:=' '20mm', 'Width:=' '-20mm', 'Height:=' ... '-5mm', 'WhichAxis:=' 'Y'}, {'NAME:Attributes', 'Name:=' 'Rectangle2', 'Flags:=' ...</pre>

<pre>0, "PartCoordinateSystem:=", _   "Global", "UDMId:=", "", "MaterialValue:=", "" &amp; Chr(34) &amp; "vacuum" &amp; Chr(34) &amp; "", "SolveInside:=", _ true)</pre>	<pre>", 'Color:=', '(132 132 193)', 'Transparency:=', 0, 'PartCoordinateSystem:=', ... 'Global', 'UDMId:=', '', 'MaterialValue:=', "vacuum", 'SolveInside:=', ... true})</pre>
<pre>oEditor.CreateRectangle Array("NAME:RectangleParameters", "IsCovered:=", true, "XStart:=", _   "31mm", "YStart:=", "0mm", "ZStart:=", "20mm", "Width:=", "-20mm", "Height:=", _   "-5mm", "WhichAxis:=", "Y"), Array("NAME:Attributes", "Name:=", "Rectangle5", "Flags:=", _   "", "Color:=", "(132 132 193)", "Transparency:=", 0, "PartCoordinateSystem:=", _   "Global", "UDMId:=", "", "MaterialValue:=", "" &amp; Chr(34) &amp; "vacuum" &amp; Chr(34) &amp; "", "SolveInside:=", _ true)</pre>	<pre>invoke(Editor,'CreateRectangle', {'NAME:RectangleParameters', 'IsCovered:=', true, 'XStart:=', ... '31mm', 'YStart:=', '0mm', 'ZStart:=', '20mm', 'Width:=', '-20mm', 'Height:=', ... '-5mm', 'WhichAxis:=', 'Y'}, {'NAME:Attributes', 'Name:=', 'Rectangle5', 'Flags:=', ... '', 'Color:=', '(132 132 193)', 'Transparency:=', 0, 'PartCoordinateSystem:=', ... 'Global', 'UDMId:=', '', 'MaterialValue:=', "vacuum", 'SolveInside:=', ... true})</pre>
<pre>oEditor.CreateRectangle Array("NAME:RectangleParameters", "IsCovered:=", true, "XStart:=", _   "31mm", "YStart:=", "0mm", "ZStart:=", "0mm", "Width:=", "20mm", "Height:=", _   "10mm", "WhichAxis:=", "Y"), Array("NAME:Attributes", "Name:=", "Rectangle6", "Flags:=", _   "", "Color:=", "(132 132 193)", "Transparency:=", 0, "PartCoordinateSystem:=", _   "Global", "UDMId:=", "", "MaterialValue:=", "" &amp; Chr(34) &amp; "vacuum" &amp; Chr(34) &amp; "", "SolveInside:=", _ true)</pre>	<pre>invoke(Editor,'CreateRectangle', {'NAME:RectangleParameters', 'IsCovered:=', true, 'XStart:=', ... '31mm', 'YStart:=', '0mm', 'ZStart:=', '0mm', 'Width:=', '20mm', 'Height:=', ... '10mm', 'WhichAxis:=', 'Y'}, {'NAME:Attributes', 'Name:=', 'Rectangle6', 'Flags:=', ... '', 'Color:=', '(132 132 193)', 'Transparency:=', 0, 'PartCoordinateSystem:=', ... 'Global', 'UDMId:=', '', 'MaterialValue:=', "vacuum", 'SolveInside:=', ... true})</pre>
<pre>oEditor.CreateRectangle Array("NAME:RectangleParameters", "IsCovered:=", true, "XStart:=", _   "0mm", "YStart:=", "0mm", "ZStart:=", "50mm",</pre>	<pre>invoke(Editor,'CreateRectangle', {'NAME:RectangleParameters', 'IsCovered:=', true, 'XStart:=', ...</pre>

<pre>"Width:=", "-100mm", "Height:=", _ "50mm", "WhichAxis:=", "Y"), Array("NAME:Attributes", "Name:=", "Rectangle4", "Flags:=", _ "", "Color:=", "(132 132 193)", "Transparency:=", 0, "PartCoordinateSystem:=", _ "Global", "UDMId:=", "", "MaterialValue:=", "" &amp; Chr(34) &amp; "vacuum" &amp; Chr(34) &amp; "", "SolveInside:=", _ true)</pre>	<pre>'0mm', 'YStart:=', '0mm', 'ZStart:=', '100mm', 'Width:=', '-200mm', 'Height:=', ... '100mm', 'WhichAxis:=', 'Y'}, {'NAME:Attributes', 'Name:=', 'Rectangle4', 'Flags:=', ... , 'Color:=', '(132 132 193)', 'Transparency:=', 0, 'PartCoordinateSystem:=', ... 'Global', 'UDMId:=', "", 'MaterialValue:=', "vacuum", 'SolveInside:=', ... true})</pre>
<pre>Set oDefinitionManager = oProject.GetDefinitionManager() oDefinitionManager.EditMaterial "NdFe35", Array("NAME:NdFe35", "CoordinateSystemType:=", _ "Cartesian", Array("NAME:AttachedData"), Array("NAME:ModifierData"), "permittivity:=", "1", "permeability:=", _ "1.0997785406", "conductivity:=", "625000", "dielectric_loss_tangent:=", "0", "magnetic_loss_tangent:=", _ "0", Array("NAME:magnetic_coercivity", "property_type:=", "VectorProperty", "Magnitude:=", - "-890000A_per_meter", "DirComp1:=", "1", "DirComp2:=", "0", "DirComp3:=", "0"), "thermal_conductivity:=", _ "0", "saturation_mag:=", "0gauss", "lande_g_factor:=", "2", "delta_H:=", "0Oe", "mass_density:=", _ "7400", "youngs_modulus:=", "147000000000", Array("NAME:thermal_expansion_coefficient", "property_type:=", _ "AnisoProperty", "unit:=", "", "component1:=", "3e-006", "component2:=", _ "-5e-006", "component3:=", "-5e-006"))</pre>	<pre>invoke(DefinitionManager,'AddMat erial', {NAME:NdFe35', 'CoordinateSystemType:=', ... 'Cartesian', {'NAME:AttachedData'}, {'NAME:ModifierData'}, 'permittivity:=', '1', 'permeability:=', ... '1.0997785406', 'conductivity:=', '625000', 'dielectric_loss_tangent:=', '0', 'magnetic_loss_tangent:=', ... '0', {'NAME:magnetic_coercivity', 'property_type:=', 'VectorProperty', 'Magnitude:=', ... '-890000A_per_meter', 'DirComp1:=', '1', 'DirComp2:=', '0', 'DirComp3:=', '0'}, 'thermal_conductivity:=', ... '0', 'saturation_mag:=', '0gauss', 'lande_g_factor:=', '2', 'delta_H:=', '0Oe', 'mass_density:=', ... '7400', 'youngs_modulus:=', '147000000000', {'NAME:thermal_expansion_coeff icient', 'property_type:=', ... 'AnisoProperty', 'unit:=', "", 'component1:=', '3e-006', 'component2:=', ...</pre>

	'-5e-006', 'component3:=' '-5e-006'})
oEditor.AssignMaterial Array("NAME:Selections", "Selections:=", "Rectangle5"), Array("NAME:Attributes", "MaterialValue:=", _ "" & Chr(34) & "copper" & Chr(34) & "", "SolveInside:=", true)	invoke(Editor,'AssignMaterial', {'NAME:Selections', 'Selections:=' 'Rectangle5'}, {'NAME:Attributes', 'MaterialValue:=' , ... "copper",'SolveInside:=' , true})
oEditor.AssignMaterial Array("NAME:Selections", "Selections:=", "Rectangle2"), Array("NAME:Attributes", "MaterialValue:=", _ "" & Chr(34) & "NdFe35" & Chr(34) & "", "SolveInside:=", true)	invoke(Editor,'AssignMaterial', {'NAME:Selections', 'Selections:=' 'Rectangle2'}, {'NAME:Attributes', 'MaterialValue:=' , ... "NdFe35",'SolveInside:=' , true})
oEditor.AssignMaterial Array("NAME:Selections", "Selections:=", "Rectangle1"), Array("NAME:Attributes", "MaterialValue:=", _ "" & Chr(34) & "steel_1010" & Chr(34) & "", "SolveInside:=", true)	invoke(Editor,'AssignMaterial', {'NAME:Selections', 'Selections:=' 'Rectangle1'}, {'NAME:Attributes', 'MaterialValue:=' , ... "steel_1010",'SolveInside:=' , true})
oEditor.AssignMaterial Array("NAME:Selections", "Selections:=", "Rectangle6"), Array("NAME:Attributes", "MaterialValue:=", _ "" & Chr(34) & "steel_1010" & Chr(34) & "", "SolveInside:=", true)	invoke(Editor,'AssignMaterial', {'NAME:Selections', 'Selections:=' 'Rectangle6'}, {'NAME:Attributes', 'MaterialValue:=' , ... "steel_1010",'SolveInside:=' , true})
Set oModule = oDesign.GetModule("BoundarySetup") oModule.AssignVectorPotential Array("NAME:VectorPotential1", "Edges:=", Array(58, _ 57, 56), "Value:=", "0", "CoordinateSystem:=", "")	Module = Design.GetModule('BoundarySetup' ) invoke(Module,'AssignVectorPoten tial',{ 'NAME:VectorPotential1', 'Edges:=' , {58, ... 57, 56}, 'Value:=' , '0', 'CoordinateSystem:=' , ''})
Set oModule = oDesign.GetModule("AnalysisSetup") oModule.InsertSetup "Magnetostatic", Array("NAME:Setup1", "Enabled:=", true, "MaximumPasses:=", _ 10, "MinimumPasses:=", 2, "MinimumConvergedPasses:=", 1, "PercentRefinement:=", _ 30, "SolveFieldOnly:=", false, "PercentError:=", 1,	Module = Design.GetModule('AnalysisSetup') invoke(Module,'InsertSetup', 'Magnetostatic', { 'NAME:Setup1', 'Enabled:=' , true, 'MaximumPasses:=' , ... 10, 'MinimumPasses:=' , 2, 'MinimumConvergedPasses:=' , 1,

<pre>"SolveMatrixAtLast:=", _   true, "PercentError:=", 1, "NonLinearResidual:=", 0.0001,      Array("NAME:MuHcOption", "MuNonLinearBH:=", _   true, "HcNonLinearBH:=", true, "ComputeHc:=", false))</pre>	<pre>'PercentRefinement:=', ...   30, 'SolveFieldOnly:=', false, 'PercentError:=',      1, 'SolveMatrixAtLast:=', ...   true, 'PercentError:=', 1, 'NonLinearResidual:=', 0.0001, {'NAME:MuHcOption', 'MuNonLinearBH:=', ...   true, 'HcNonLinearBH:=', true, 'ComputeHc:=', false}})</pre>
<pre>oDesign.AnalyzeAllNominal</pre>	<pre>invoke(Design,'Analyze', 'Setup1') Design.AnalyzeAllNominal</pre>
<pre>Set          oModule          = oDesign.GetModule("FieldsReporter")  oModule.CreateFieldPlot Array("NAME:Flux_Lines1", "SolutionName:=", _   "Setup1 : LastAdaptive", "QuantityName:=", "Flux_Lines",      "PlotFolder:=",      "A", "UserSpecifyName:=", _   0, "UserSpecifyFolder:=", 0, "StreamlinePlot:=", false, "IntrinsicVar:=", "", "PlotGeomInfo:=", Array( _   1, "Surface", "FacesList", 5, "16", "28", "40", "52", "64"), "FilterBoxes:=", Array( _   1, ""), Array("NAME:PlotOnSurfaceSettings", "Filled:=", false, "IsoValType:=", "Line", "SmoothShade:=", _   true, "AddGrid:=", false, "MapTransparency:=", true, "Refinement:=", 0, "Transparency:=", _   0,      Array("NAME:Arrow3DSpacingSettings", "ArrowUniform:=", true, "ArrowSpacing:=", 0, "MinArrowSpacing:=", _   0, "MaxArrowSpacing:=", 0), "GridColor:=", Array(255, 255, 255)))</pre>	<pre>Module          = Design.GetModule('FieldsReporter' )  invoke(Module,'CreateFieldPlot', {'NAME:Flux_Lines1', 'SolutionName:=', ...   'Setup1      :      LastAdaptive', 'QuantityName:=', 'Flux_Lines', 'PlotFolder:=',      'A', 'UserSpecifyName:=', ...   0, 'UserSpecifyFolder:=', 0, 'StreamlinePlot:=', false, 'IntrinsicVar:=', "", 'PlotGeomInfo:=', { ...   1, 'Surface', 'FacesList', 5, '16', '28', '40', '52', '64'}, 'FilterBoxes:=', { ...   1, ""}, {'NAME:PlotOnSurfaceSettings', 'Filled:=', false, 'IsoValType:=', 'Line', 'SmoothShade:=', ...   true, 'AddGrid:=', false, 'MapTransparency:=', true, 'Refinement:=',      0, 'Transparency:=', ...   0, {'NAME:Arrow3DSpacingSettings' , 'ArrowUniform:=', true, 'ArrowSpacing:=',      0, 'MinArrowSpacing:=', ...</pre>

	0, 'MaxArrowSpacing:=', 0}, 'GridColor:=', {255, 255, 255}}))
oModule.CreateFieldPlot Array("NAME:Mag_B1", "SolutionName:=", _ "Setup1 : LastAdaptive", "QuantityName:=", "Mag_B", "PlotFolder:=", "B", "UserSpecifyName:=", _ 0, "UserSpecifyFolder:=", 0, "StreamlinePlot:=", false, "IntrinsicVar:=", "", "PlotGeomInfo:=", Array( _ 1, "Surface", "FacesList", 5, "16", "28", "40", "52", "64"), "FilterBoxes:=", Array( _ 1, ""), Array("NAME:PlotOnSurfaceSettings", "Filled:=", false, "IsoValType:=", _ "Fringe", "SmoothShade:=", true, "AddGrid:=", false, "MapTransparency:=", true, "Refinement:=", _ 0, "Transparency:=", 0, Array("NAME:Arrow3DSpacingSettings", "ArrowUniform:=", true, "ArrowSpacing:=", _ 0, "MinArrowSpacing:=", 0, "MaxArrowSpacing:=", 0), "GridColor:=", Array(255, _ 255, 255)))	invoke(Module,'CreateFieldPlot', {'NAME:Mag_B1', 'SolutionName:=', ... 'Setup1 : LastAdaptive', 'QuantityName:=', 'Mag_B', 'PlotFolder:=', 'B', 'UserSpecifyName:=', ... 0, 'UserSpecifyFolder:=', 0, 'StreamlinePlot:=', false, 'IntrinsicVar:=', '', 'PlotGeomInfo:=', { ... 1, 'Surface', 'FacesList', 5, '16', '28', '40', '52', '64'}, 'FilterBoxes:=', {1, ""}, {'NAME:PlotOnSurfaceSettings', 'Filled:=', ... false, 'IsoValType:=', 'Fringe', 'SmoothShade:=', true, 'AddGrid:=', false, 'MapTransparency:=', ... true, 'Refinement:=', 0, 'Transparency:=', 0, {'NAME:Arrow3DSpacingSettings', 'ArrowUniform:=', ... true, 'ArrowSpacing:=', 0, 'MinArrowSpacing:=', 0, 'MaxArrowSpacing:=', 0}, 'GridColor:=', { ... 255, 255, 255}}))
	delete(iMaxwell);

## 6. REFERENCES

- [1] Zamri N. A. M., Ibrahim T. and Nor N. M. Analytical prediction of permanent magnet linear generator for wave energy conversion system in Malaysia Ocean. J. Fundam. Appl. Sci., 2018, 10(7S), 229-240.
- [2] Sinnadurai R, Ting C M, Hadi Q B. Modeling and design analysis of a three phase linear generator for sea waves energy conversion. J. Fundam. Appl. Sci., 2018, 10(6S), 366-386.

- 
- [3] Boldea I., Nasar S. A. Linear electric actuators and generators. Cambridge: Cambridge University Press., 1997.
- [4] Boldea I., Nasar S. A. Permanent-magnet linear alternators part 1: fundamental equations. IEEE Transactions on Aerospace and Electronic Systems, 1987, 3, 73-78.
- [5] Boldea I. Linear electric machines, drives, and MAGLEVs handbook. CRC Press., 2013.
- [6] Boldea I., Nasar S. A. Linear motion electromagnetic devices. New York: Taylor & Francis, 2001.
- [7] Arshad W. M., Thelin P., Bäckström T., Sadarangani C. Alternative electrical machine solutions for a free piston generator. In The Sixth Intl Power Engineering Conference (IPEC2003), Singapore, 2003.
- [8] Arshad W. M., Backstrom T., Thelin P., Sadarangani C. Integrated free-piston generators: an overview. IEEE NORPIE-02 Conference, Stockholm, 2002.
- [9] Arshad W. M., Sadarangani C., Backström T., Thelin P. Finding an appropriate electrical machine for a free piston generator. 19th Electrical Vehicle Symposium (EVS), Korea, 2002.
- [10] Wang J., Howe D. Design optimization of radially magnetized, iron-cored, tubular permanent magnet machines and drive systems. IEEE Transactions on Magnetics, 2004, 40, 3262–3277.
- [11] Chen A., Arshad W. M., Thelin P., Zheng P. Analysis and optimization of a longitudinal flux linear actuator for hybrid electric vehicle applications. IEEE Symp. Vehicle Power and Propulsion, 2004.
- [12] Wang J., Howe D. Influence of soft magnetic materials on the design and performance of tubular permanent magnet machines. IEEE Transactions on Magnetics, 2005, 41, 4057-4059.
- [13] Wang, J. Inoue M., Amara Y., Howe D. Cogging-force-reduction techniques for linear permanent-magnet machines. IEEE Proceedings-Electric Power Applications, 2005, 152, 731-738.
- [14] Pohl S. E., Graf M., Dynamic simulation of a free-piston linear alternator in modelica.



- Proceedings of the 4th International Modelica Conference, Hamburg, 2005, 393-399.
- [15] Ahmad, M. E., Lee H. W., Nakaoka M. Detent force reduction of a tubular linear generator using an axial stepped permanent magnet structure, *Journal of Power Electronics*, 2006, 6, 290–297.
- [16] Commins P., Moscrop J. W., Cook C. D. Novel tooth design for a tubular linear motor for machine tool axis. *IEEE International Conference on Mechatronics*, April 2011, 660-665.
- [17] Dong X., Papini F., Tamma B. Dynamic analysis tool for the design of tubular linear permanent magnet machines. *9th International Multi-Conference on Systems, Signals and Devices*, 2012, 1-5.
- [18] Oswald A., Herzog H G, Automated design of tubular linear generator for use within optimization tool. *XIX International Conference on Electrical Machines*, September 2010, 1-6.
- [19] Bianchi, N., Bolognani S., Grezzani G. Object-oriented algorithms for automatic finite element analysis of PM motors. *Second International Conference on Power Electronics, Machines and Drives*, University of Edinburgh, UK, 31 March-2 April 2004, paper no.498, 817-822.
- [20] Giacometti F., Lines C., Cruise R. Optimisation of the Slot Dimensions of a Large Air-gap Linear Synchronous Motor. Excerpt from the *Proceedings of the 2014 COMSOL Conference*, Cambridge, 2014.
- [21] Gieras F. J., Piech J. Z., Tomczuk B. *Linear synchronous motors: transportation and automation systems*, 2nd Edition, New York: CRC Press, 2011.
- [22] Liu Y., Arshad W. M., Leksell M., Thelin P. Influence of speed and current profiles upon converter dimensioning and electrical machine performance in a free-piston generator. In *Proceedings of the 4th international Symposium on Linear Drives for Industry Applications*, September 2003, 8-10.
- [23] Ansys Maxwell v16.0 Help File.
- [24] Osman G., *Electromagnetic Field Theory*. Ankara: Seckin Publishing House, 2007.
- [25] Ahmet F., İlhan T. Bir Manyetik Sistemin Maxwell 3D Alan Simülâtörü İle Statik Manyetik Analizinin Çözüm Süreçleri. *Journal of Technical Online*, 2007, 6, 221-240.



**How to cite this article:**

Arslan S, Gürdal O, Akkaya Oy S. Communication of Matlab Gui and Ansys Maxwell: An Education Tool For Tubular Linear Generator. *J. Fundam. Appl. Sci.*, 2019, *11(1)*, 117-141.