PROJECT RYU
# Contents

0. Abstract ......................................................................................................................... 2  
1. Introduction .................................................................................................................. 2  
2. Motor Structure ............................................................................................................. 3  
3. Input Parameters .......................................................................................................... 4  
4. Simulations ................................................................................................................... 5  
5. Advanced parameters .................................................................................................. 5  
6. Using the script with FEMM ......................................................................................... 6  
7. Conclusions .................................................................................................................. 7
0. Abstract

Information about loudspeaker design is very hard to find and there very little tools out there to help the aspiring DIY-er. The purpose of these scripts is to provide a starting point for a loudspeaker magnetic circuit design by simulating various geometries and parameter sets and choosing the best result.

These scripts are meant to be used with Finite Element Method Magnetics (FEMM) software. FEMM is a free application written by Dr. David Meeker that can do FEM analysis of magnetic circuits.

The scripts are written in Lua programming language and are available in the Project Ryu Loudspeaker Drivers repository on GitHub. Please visit the FEMM homepage for more information about the software and download link.

Depending on the time you will read this article there will probably be more than one script corresponding to different types of motors. In this article I will focus on the field coil motor script.

1. Introduction

Dynamic loudspeaker motors are basically actuators. A voice coil is immersed in a constant magnetic field and when current flows through the coil it will create a force (a consequence of Lorentz force) whose direction depends on the direction of the current through the coil.

In dynamic loudspeakers the constant magnetic field is generated with either a permanent magnet or a field coil. This magnetic field is then directed and concentrated in the area where the voice coil is. A magnetic circuit created from a material with much greater magnetic permeability than air (usually low carbon steel) is used to achieve this.

The force acting on the voice coil will be a product of magnetic flux density and length of the wire multiplied by current intensity. Ideally, the magnetic flux lines will always be perpendicular to the direction of the current flow in the wire and thus the force’s magnitude can be express in scalar form as follows:

\[ F = B L i \]

where \( F \) is the force, \( i \) is the current through the voice coil, \( L \) is the total length of the wire in the voice coil and \( B \) is the magnetic flux density. If we consider \( i \) to be the input of your system and \( F \) the output we can see how the electrical signal is transformed into a mechanical signal and how the product \( B L \) can be thought as an electro-mechanical coupling factor. Since \( L \) depends on voice coil, our interest in designing the magnetic circuit is focused on magnetic flux density \( B \).

Using FEM analysis we can predict this parameter for certain magnetic structures and one great free tool for this is FEMM application. This said, as with most applications, it requires some time, practice and knowledge to utilize it. For this reason I have created the helper script that should get any DIY-er started faster. It will generate a motor structure and field coil circuit based on your desired \( B \) value and a few other parameters.
2. Motor Structure

The geometry of the motor structure is calculated from the input parameters but it will result in classic motor structures. For field coil motors the structure consists of top steel plate, steel ring, steel bottom plate, steel central pole piece and the copper field coil.

Figure 1 depicts the classic structure of a field coil underhung motor and identifies it’s components. It is called an underhung motor because the voice coil’s winding height is smaller than the magnetic gap height.

![Field Coil Motor Structure](image)

**Figure 1. Field Coil Motor Structure.** (a) Voice Coil Former, (b) Voice Coil Winding, (c) Magnetic Gap, (d) Steel Top Plate, (e) Field Coil, (f) Steel Ring, (g) Steel Central Pole Piece, (h) Magnetic Flux Lines, (i) Steel Bottom Plate

Magnetic field is strongest at the center of the field coil and this is why the central pole piece is the easiest to saturate. We must ensure we have enough steel there to avoid this. If this section is saturated it will limit the flux density in the magnetic gap not matter how much the current through the field coil is increased or how big the field coil itself is. We should ensure that the saturation point is in the gap region only.

The next areas with strong magnetic field is on the field coil’s axis, above and below the field coil itself. The bottom plate’s thickness in that region must be considered again to avoid saturation. Above the field coil we have the magnetic gap and this is the area it is best to
saturate. As shown in Figure 1, the diameter of the central pole piece is reduced. This is because we need to fit the voice coil but also because it will concentrate the flux lines to the gap.

There will be a few flux lines crossing the air region between the field coil and the top plate because of this but with underhung motors it shouldn’t pose a problem. The reason saturation of pole pieces in the gap region is desired is because this way the flux density will be more stable and less prone to be influenced by the magnetic field generated by the voice coil itself. Another thing to note about the classic structure is that the magnetic gap height usually equals the thickness of the top plate.

3. Input Parameters

In loudspeaker motor design the most important area is the magnetic gap. One could argue the end result would be the same if the magnetic gap parameters and geometry are the same no matter what the rest of the motor looks like and under specific working conditions that would be true. For this reason, I chose the input parameters to those corresponding to the magnetic gap. Let’s enumerate them:

- **Voice coil diameter**: this represents the mean diameter

  \[ d_{vcm} = \left( d_{vco} + d_{vci} \right)/2 \]

  where \( d_{vcm} \) is the voice coil mean diameter, \( d_{vci} \) the voice coil inner diameter and \( d_{vco} \) the voice coil outer diameter. This parameter will determine the radius at which we will acquire the magnetic flux density values in the gap.

- **Voice coil height**: this represents the winding height and it is used together with desired \( X_{max} \) to calculate top plate thickness, magnetic gap height

  \[ h_g = 2 \times X_{max} + h_{vc} \]

  where \( h_g \) is the gap height and \( h_{vc} \) is voice coil height.

- **Voice coil width**: this represents the thickness of the voice coil including winding and former. One could define this parameter as:

  \[ w_{vc} = \left( d_{vco} - d_{vci} \right)/2 \]

  This parameter is used in computing the magnetic gap width by adding a clearance factor to the inside and outside of the voice coil.

  \[ w_g = w_{vc} + 2 \times w_{clear} \]

  where \( w_g \) is the magnetic gap width, \( w_{vc} \) the voice coil width and \( w_{clear} \) is the clearance between the voice coil and the steel pole pieces. The default clearance value is 0.3mm and in section 5 I will describe how to modify this value.

- **\( X_{max} \)**: this represents the desired distance the voice coil can travel in each direction with
negligible change in flux density.

- **Flux density**: represents the desired $B$ value in the magnetic gap. This value is used in the simulation process that will be described in the next section.

All these parameters are requested via a prompt when running the script. This prompt can be disabled if you can handle adjusting the parameters in the code.

Working directly on code will also enable you to adjust more advanced settings which will be discussed in section 5.

### 4. Simulations

Once the input parameters are set the script will begin the analysis. It will iterate through different motor sizes and for each motor size it will simulate different field coils by changing the wire diameter and thus the number of turns also. The physical size of the field coil will always be the same for a give motor size. For each field coil the script will use a range of current values and calculate the magnetic flux density and field coil properties.

Each simulation result that passes the desired flux density value will be stored. Once all the simulations are finished an optimization process begins. The program will take each stored result and calculate a weighted sum between flux density, motor size and field coil power. The record with the highest score will be chosen as optimum and results displayed. The weights of the sum represent the importance the user puts on specific parameter. For example if the user is willing to sacrifice a little flux density in order to get a more compact motor than the weight of the flux density parameter will be lower than that of the motor size parameter.

Under User Constants we can find the weight vector $w$ with $w[‘dia’]$ being the weight for motor size, $w[‘pow’]$ - weight for field coil power and $w[‘flx’]$ - weight for flux density. The weights take values between 0 and 1 with 0 meaning ignore and 1 meaning the most important.

**e.g.**

Objective: dont care about motor size, sacrifice a bit from flux density value for the best field coil power value

Weight vector:

$$w = \{[‘dia’] = 0, [‘pow’] = 1, [‘flx’] = 0.7\}$$

**Score:**

$$s = w[‘dia’] \times d_m + w[‘pow’] \times P_{fc} + w[‘flx’] \times B$$

where $s$ is the score value, $d_m$ is the motor diameter, $P_{fc}$ is the field coil power and $B$ is the magnetic flux density in the gap.

### 5. Advanced parameters

There are a number of parameters which can only be adjusted by modifying the script’s code. These parameters will influence the geometry, the total number of simulations and the end result.

- **Bottom plate thickness** [ default: 20 mm ]: This parameter influences the geometry and thus...
the magnetic circuit. If the bottom plate saturates you should increase the value. To adjust it modify the \texttt{plate} parameter under Program Constants.

- **Steel ring thickness** [ default: 10 mm ]: This parameter influences the geometry and thus the magnetic circuit. If this region saturates you should increase the value. To adjust it modify the \texttt{canplate} parameter under Program Constants.

- **Motor height** [ default: 120 mm ]: This represents the total motor height. To adjust it modify the \texttt{mheight} parameter under User Variables.

- **Minimum motor diameter** [ default: 120 mm ]: This is the start value that defines the motor size range to be simulated. This motor diameter will be the first to be simulated and then the diameter is incremented until it reaches the maximum value. To adjust it modify the \texttt{min\_dia} variable under User Constants.

- **Maximum motor diameter** [ default: 180 mm ]: This is the end value that defines the motor size range to be simulated. To adjust it modify the \texttt{max\_dia} variable under User Constants.

- **Minimum field coil wire diameter** [ default: 0.3 mm ]: For each motor geometry the scripts simulates a range of field coil wire diameters that will result in different number of turns and impedance. This parameter sets the start value of this range. Modify the \texttt{min\_cwdia} to adjust it.

- **Maximum field coil wire diameter** [ default: 1.5 mm ]: Maximum wire diameter to be used in field coil. Modify the \texttt{max\_cwdia} to adjust it.

- **Field coil start current** [ default: 0.1 A ]: As mentioned in previous section for each motor the script will test a range of current values through the field coil. This parameter holds the start value for the current and thus will influence the total number of simulations and the script’s running time. If you know the range of the current you will feed the field coil you can modify the variable \texttt{start\_} under Program Constants to reduce the simulation time.

- **Magnetic gap clearance** [ default: 0.3 mm ]: This parameter represents the distance between the voice coil and the pole pieces. In order to move freely there has to be a clearance between the voice coil and the steel. It influences the magnetic gap width value. To adjust it modify \texttt{gapclear\_} variable under Program Constants.

6. Using the script with FEMM

Usage is very simple, once FEMM application is started, press the “Open Lua script” button in the left tool bar indicated in figure 2 with a red rectangle. The script will then execute. By default the \texttt{prompter} variable is set to 1 and this enables a dialog with the user asking for the input parameters presented in section 3 after the user inputs a project name. The FEMM project will be saved under this name by default in \texttt{C:\textasciitilde\folder}. You can change this location by modifying the code and adjust the \texttt{save\_path} variable.
Once the script finished running, the final simulation will be loaded and the user can further improve it or export dxf with the geometry.

7. Conclusions

At the time this article was written, the script can be greatly improved but still provides a good start in designing a loudspeaker motor. Improvements that will follow will be increased mesh size for FEM analysis around the gap for better resolution, increased number of steel options, generate bottom plate and steel ring thickness dynamically, optimization of magnetic gap geometry just to name a few.

Since the project is open source, contributions are most welcomed, please follow the GitHub link at the top to get to the project's repository. Also reporting bugs is very appreciated so feel free to drop an email at contact[at]projectryu.com or report an issue on GitHub.